# KNOWLEDGE TRANSFER IN DYNAMIC MULTI-OBJECTIVE OPTIMIZATION

by

**GAN RUAN**

A thesis submitted to the University of Birmingham for the degree of DOCTOR OF PHILOSOPHY

School of Computer Science

College of Engineering and Physical Sciences

University of Birmingham

March 2024

UNIVERSITY OF
BIRMINGHAM

## University of Birmingham Research Archive

### e-theses repository

This unpublished thesis/dissertation is copyright of the author and/or third parties. The intellectual property rights of the author or third parties in respect of this work are as defined by The Copyright Designs and Patents Act 1988 or as modified by any successor legislation.

Any use made of information contained in this thesis/dissertation must be in accordance with that legislation and must be properly acknowledged. Further distribution or reproduction in any format is prohibited without the permission of the copyright holder.

# Abstract

Dynamic multi-objective optimization problems (DMOPs) are optimization problems possessing multiple conflicting objectives varying with time. This type of problems widely exists in the real world, such as planning, resource allocation and scheduling problems. Knowledge transfer, a methodology inspired by the machine learning community, has been recently used for solving DMOPs, since it can transfer useful information from solving one problem instance to solve another related problem instance, potentially speeding up the optimization process for the new instance. This thesis contributes to the study of knowledge transfer in dynamic multi-objective optimization as follows:

1. Intuitively, it is doubtful that knowledge transfer approaches would always be beneficial for improving optimization when solving DMOPs with changing position and/or shape of Pareto sets (PSs) and/or Pareto fronts (PFs), since transferring knowledge from an unrelated problem would be harmful to the optimization of the target problem. Therefore, an investigation is made to determine in which cases transfer works and fails. We find that knowledge transfer indeed does not always work and show that it fails on problems with fixed PS and when environmental changes are small. Moreover, we show that using a linear kernel function to transfer results in better quality of transferred solutions than using a Gaussian one.

2. Given that knowledge transfer contributes to improving optimization quality but introduces an overhead in the optimization process, a series of computational studies are carried out to comprehensively understand the efficiency and effectiveness of knowledge transfer when solving DMOPs with changing position and/or shape of PFs/PSs. Specifically, we computationally find that the 'inner' optimization method introduced by transfer learning is very time-consuming. We also find that spending the time optimizing instead of transferring knowledge would lead to better results, which defeats the purpose of knowledge transfer. Then, we propose to use two alternatives to replace the existing 'inner' optimization method and find that the transfer algorithm with one alternative achieves better transfer efficiency and solutions quality.

3. Different from most other DMOPs where dynamic changes lead to changing position and/or shape of PSs and/or PFs, DMOPs with a changing number of objectives usually result in expansion or contraction of the PF or PS manifold when the number of objectives increases or reduces, respectively. Therefore, to improve the optimization process when there is a change in the number of objectives, it is intuitive to determine the expansion and contraction directions. These directions could then be used to generate transferred solutions for the new environment and further expand or contract the PSs. Therefore, we firstly propose a heuristic way to determine PS expansion and contraction directions. Then, a novel knowledge transfer algorithm is proposed via PS expansion and contraction based on the found directions for solving DMOPs with a changing number of objectives. The algorithm is able to achieve transferred solutions

with good diversity after changes, improving optimization especially in fast changing environments.

4. Moreover, it may also be possible to learn the expansion and contraction directions, so that determining them does not rely entirely on heuristics, potentially leading to more effective transfer of knowledge. Therefore, a novel learning method for determining the expansion and contraction directions is proposed based on principal component analysis, further expanding and contracting the PSs. Specifically, principal component analysis is firstly conducted on the PS of the previous environment to get the candidates of the expansion and contraction directions. Then, the most promising directions for the expansion and contraction are selected according to dominance relationship between existing solutions and those generated along the candidate directions. This approach is able to improve solution quality, not only right after changes but also after optimization of different generations.

# Acknowledgements

First of all, I am immensely grateful to my supervisors Prof. Xin Yao and Dr. Leandro L. Minku. for their guidance, expertise, kindness and unwavering supports. Without their incredible patience, continuous encouragement, professional supervision, and selfless dedication, I could not even start to write this thesis. Not only do they assist me in building a scientific and rigorous research system, but also inspire me to cultivate an attitude of logic, comprehensive and critical thinking in almost all aspects of my life, for which I think it is the most significant purpose of education. Prof. Xin Yao has been a tremendous mentor for me and always pushes me to think deeper and ask myself why, which is extremely beneficial for the cultivation of my critical thinking. Dr. Leandro L. Minku is very supportive, nice, hard-working and always gives helpful when I faced difficulties in research, writing and life. The experience of working with Xin and Leandro is invaluable.

I would also express my thanks to the ECOLE project who gives the Funding for my studies and researches at the University of Birmingham. I would also extend my greatest thanks to my industrial supervisors Prof. Bernhard Sendhoff, Dr. Stefan Menzel from Honda Research Institute Europe, Dr. David Hudson, Dr. Robin de Regt from TATA Steel Europe and Dr. Zhao Xu from NEC Laboratories Europe for their supervisions, suggestions and supports as well as their helps when I was doing my secondments at these institutes.

I would also like to thank my thesis group member (Peter Tino) for his helpful suggestions and discussions on my research at every thesis group meeting, all of which greatly help improving the quality of my PhD research and the thesis.

I would also thank to Prof. Thomas Bäck from University of Leiden as well as the supervisors from Honda Research Institute Europe, NEC Laboratories Europe and University of Birmingham as ECOLE supervisors within the project for their useful suggestions and comments. I also thank to these four ECOLE beneficiaries for their arrangement and supports on holding summer schools, workshops, conferences, training courses and scientific talks to us as early stage researchers. I would also like to thank project mangers Miss Manisha Bhesania and Miss Hilary Anderson in the ECOLE projects for their arrangement on our secondments and corresponding meetings within the project.

I would also thank to all early stage researchers Dr. Stephen Friess from University of Birmingham, Dr Jiawen (Fay) Kong, Dr. Sibghat Ullah and Dr. Duc Anh Nguyen from University of Leiden, Dr. Thiago Rios, Dr. Sneha Saha from Honda Research Institute Europe, Dr. Giuseppe Serra from NEC Laboratories Europe within the ECOLE project for their companies, discussions and help.

I would also like to thank Prof. Jinhua Zheng, Prof. Juan Zou, Dr. Zhongwei Ma, Dr. Guo Yu, Dr. Zhanglu Hou, Dr. Xunzhao Yu, Dr. Ke Li, Dr. Renzhi Chen, for their help and support before and during my PhD studies.

Finally, I would give my special thanks to Prof. Tan Kay Chen and Dr. Miqing Li for agreeing to be my thesis examiners. Many thanks to them for giving me valuable and insightful suggestions and comments on improving the quality of my thesis. Also thank Dr. Shuo Wang for being the chair of my viva.

# Contents

**List of Figures** vii

**List of Tables** xiii

**1 Introduction** 1
- 1.1 Motivations . . . 2
- 1.2 Research Questions . . . 5
- 1.3 Contributions . . . 7
- 1.4 List of Publications Generated from the Thesis . . . 8

**2 Background and Literature Review** 10
- 2.1 Dynamic Multi-objective Optimization (DMO) . . . 10
  - 2.1.1 Basic Concepts of DMO . . . 10
  - 2.1.2 DMO with Changing Pareto Sets (PSs) and/or Pareto Fronts (PFs) . . . 11
    - 2.1.2.1 Diversity Strengthening . . . 11
    - 2.1.2.2 Memory Strategies . . . 12
    - 2.1.2.3 Prediction techniques . . . 13
    - 2.1.2.4 Transfer Optimization Approaches . . . 13
  - 2.1.3 DMO with a Changing Number of Objectives . . . 14
    - 2.1.3.1 Inheritance Strategy for Objective Increment or Replacement of Objectives . . . 14
    - 2.1.3.2 Comprehensive Analysis of DMOPs with a Changing NObj and Its State-of-the-art Solver . . . 15
- 2.2 Knowledge Transfer in Evolutionary Optimization . . . 19
  - 2.2.1 Basic Concepts of Transfer Optimization . . . 19
  - 2.2.2 Knowledge Transfer in Static Evolutionary Optimization . . . 21
    - 2.2.2.1 Explicit Knowledge Transfer . . . 21
    - 2.2.2.2 Implicit Knowledge Transfer . . . 23
  - 2.2.3 Knowledge Transfer in Dynamic Evolutionary Optimization . . . 25
    - 2.2.3.1 Transfer Learning-based Dynamic Multi-objective Evolutionary Algorithms (Tr-DMOEAs) . . . 26
    - 2.2.3.2 Other Transfer-based Algorithms . . . 28

**3 When and How to Transfer in Tr-DMOEAs [1] 30**
3.1 When Does Transfer Optimization Work/Fail in Dynamic Multi-objective Optimization? . . . 31
3.1.1 Computational Investigation into Transfer in DMO . . . 31
3.1.2 Avoiding Transfer on Problems with Fixed PS And Small Environmental Changes . . . 34
3.1.3 Experimental Studies of the Proposed Strategy . . . 35
3.2 How Does Transfer Work in Dynamic Multi-objective Optimization? . 36
3.2.1 Mathematical Proof that Gaussian Kernels Are Not Ideal . . . 36
3.2.2 Replacing Gaussian Kernel with Linear Kernel . . . 38
3.2.3 Experimental Evaluation of Different Kernels . . . 38
3.3 Improved Transfer in DMO . . . 40
3.3.1 Proposed Method to Generate An Initial Population . . . 40
3.3.2 Evaluation through Computational Studies . . . 41
3.3.2.1 Quality Comparison in the First Generation . . . 41
3.3.2.2 Solutions Quality Comparison after Optimization . . 42
3.3.2.3 Impact of EAs on Different DMOEAs . . . 46
3.3.2.4 Performance of DMOEAs on Different Benchmarks . 47
3.3.2.5 Influence of Different Settings of $\tau_t$ and $n_t$ on Tr-DMOEAs . . . 47
3.4 Summary . . . 48

**4 Improving the Efficiency of Transfer in Tr-DMOEA for Changing Shape/Position of PSs/PFs [2] 49**
4.1 Efficiency of Transfer Optimization in DMO . . . 50
4.2 Solutions Quality after Optimization for Which Transfer Cost Is Used 51
4.2.1 Computation Time of Transfer Used for Optimization . . . 51
4.2.2 Quality Comparison of Transferred And Random Solutions with Same Cost Budget . . . 52
4.2.2.1 Solution Quality Comparison in the First Generation 54
4.2.2.2 Solution Quality Comparison after Optimization . . 56
4.3 Computational Analysis of Three Tr-DMOEA Variants . . . 57
4.3.1 Computation Time Comparison of Three Tr-DMOEA Variants 57
4.3.2 Solutions Quality Comparison of Three Tr-DMOEA Variants . 60
4.3.2.1 Solutions Quality Comparison in the First Generation 60
4.3.2.2 Solutions Quality Comparison after Optimization . . 62
4.4 Summary . . . 64

**5 Diversity Enhancing Knowledge Transfer via PS Expansion/Contraction in DMO with a Changing Number of Objectives (NObj) [3] 65**
5.1 A Preliminary Investigation of State-of-the-art . . . 67
5.2 Proposed Knowledge Transfer DMOEA (KTDMOEA) . . . 69
5.2.1 Diversity Enhancing Knowledge Transfer . . . 69

[1] This chapter is written based on content from the published paper [1] as [*1] in Section 1.4.
[2] This chapter is written based on content from the published paper [2] as [*2] in Section 1.4.
[3] This chapter is written based on content from the submitted paper as [*3] in Section 1.4.

5.2.1.1 Expand the PS when Increasing the NObj . . . . . 70
5.2.1.2 Contract the PS when Decreasing the NObj . . . . . 74
5.2.2 KTDMOEA . . . . . 75
5.2.3 Time Complexity Analysis . . . . . 76
5.3 Experimental Setup . . . . . 77
5.3.1 Benchmark Problems . . . . . 77
5.3.2 Compared Algorithms . . . . . 78
5.3.3 Parameter Settings . . . . . 80
5.3.4 Performance Metrics . . . . . 80
5.4 Experimental Results and Analyses . . . . . 81
5.4.1 Initial Effectiveness of Knowledge Transfer . . . . . 82
5.4.1.1 NObj Increasing from 2 to 7 and then Decreasing from 7 to 2 . . . . . 82
5.4.1.2 NObj Decreasing from 7 to 2 and then Increasing from 2 to 7 . . . . . 86
5.4.1.3 Why Does Knowledge Transfer Usually Get Better Solution Quality Right after Changes? . . . . . 90
5.4.1.4 Why Doesn't Knowledge Transfer Sometimes Get Better Solution Quality after Changes? . . . . . 91
5.4.2 How Does Knowledge Transfer Help Optimization? . . . . . 92
5.4.2.1 NObj Increasing from 2 to 7 And then Decreasing from 7 to 2 . . . . . 92
5.4.2.2 NObj Decreasing from 7 to 2 And then Increasing from 2 to 7 . . . . . 99
5.4.2.3 Why Does Knowledge Transfer Help Optimization? . 103
5.4.2.4 Why Doesn't Knowledge Transfer sometimes Help Optimization? . . . . . 103
5.4.3 Further Analysis of Our Knowledge Transfer Methods . . . . . 106
5.4.4 Performance Comparison on Other Changes in the NObj . . . 109
5.4.5 Impact of Algorithm Parameters . . . . . 112
5.5 Summary . . . . . 113

**6 Learning to Expand and Contract PSs in DMO with a Changing NObj [4] 117**
6.1 A Further Computational Investigation of KTDMOEA . . . . . 119
6.2 Learning to Expand and Contract PSs (LEC) . . . . . 123
6.2.1 Learning Candidate Expansion and Contraction Directions via Principal Component Analysis . . . . . 124
6.2.1.1 Learning Candidate Expansion Directions . . . . . 124
6.2.1.2 Learning Candidate Contraction Directions . . . . . 126
6.2.2 Most Promising Expansion and Contraction Directions Selection 127
6.2.2.1 Selection of Expansion Directions to Improve Diversity 127
6.2.2.2 Selection of Contraction Directions to Improve Convergence . . . . . 128
6.2.3 PS Expansion and Contraction . . . . . 129
6.2.3.1 Diversity Enhancing PS Expansion . . . . . 129

[4]This chapter is written based on content from the submitted paper as [*4] in Section 1.4.

6.2.3.2 Convergence Enhancing PS Contraction . . . . . 129
6.2.4 Time Complexity Analysis . . . . . 130
6.3 Experimental Setup . . . . . 130
6.3.1 Benchmark Problems . . . . . 131
6.3.2 Compared Algorithms . . . . . 131
6.3.3 Parameter Settings . . . . . 132
6.3.4 Performance Metrics . . . . . 133
6.4 Experimental Results and Analyses . . . . . 133
6.4.1 Initial Effectiveness of Learnt PS Expansion and Contraction . 134
6.4.1.1 Overall Statistic and Specific Metric Results . . . . . 134
6.4.1.2 How and Why Does LEC Get Better Solution Quality Directly after Changes? . . . . . 143
6.4.1.3 Why Doesn't LEC sometimes Get Good Solution Quality Directly after Changes? . . . . . 143
6.4.2 How Does Learnt PS Expansion and Contraction Help Optimization Process after the Changes? . . . . . 145
6.4.2.1 Overall Statistic and Specific Metric Results . . . . . 145
6.4.2.2 Why Does Learnt PS Expansion and Contraction Help Optimization Process after the Changes? . . . . 152
6.4.2.3 Why Doesn't Learnt PS Expansion and Contraction sometimes Help Optimization Process after the Changes? . . . . . 152
6.4.3 Performance Comparison on Other Changes in the NObj . . . 171
6.4.4 Impact of Algorithm Parameters . . . . . 173
6.4.5 Running Time Analysis . . . . . 177
6.5 Summary . . . . . 177

**7 Conclusions and Future Work** **179**
7.1 Conclusions . . . . . 179
7.2 Future Work . . . . . 182

**Bibliography** **184**

# List of Figures

2.1 Comparison of population distributions when increasing the number of objectives on DTLZ2. Both figures are from Fig. 1 of [3]. . . . . 16

2.2 Comparison of population distributions when decreasing the number of objectives on DTLZ2. Both figures are from Fig. 2 of [3]. . . . . 16

3.1 Friedman ranking among MIGD of obtained solutions by all DMOEAs from left to right, 4 responding strategies with respect to 3 EAs including RMMEDA, NSGA-II and MOPSO (referred as RM, NS and MO for simplicity). Any pair of approaches whose distance between them is larger than CD are considered to be different. . . . . 46

4.1 Friedman ranking for three optimization methods (active set, SQP and interior point) from left to right regarding (a) computation time, (b) MIGD of solutions obtained at the first generation and (c) MIGD of optimized solutions at the last generation. Any pair of approaches whose distance between them is larger than CD are considered to be significantly different based on the Nemeyi post-hoc test. . . . . 60

5.1 Distribution of the reconstructed CA and DA obtained by DTAEA in the first generation right after changes when increasing NObj from 2 to 3 on F2 and WFG4. . . . . 68

5.2 Distribution of the reconstructed CA and DA obtained by DTAEA in the first generation right after changes when decreasing NObj from 3 to 2 on F2 and WFG4. . . . . 68

5.3 Flow chart of KTDMOEA. . . . . 70

5.4 Brief illustration of how to expand and contract the PS for increasing and decreasing NObj, respectively. . . . . 71

5.5 Friedman ranking among MHV, MGD and MMS of obtained solutions at the first generation by 6 algorithms in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 82

5.6 Friedman ranking among MHV, MGD and MMS of obtained solutions at the first generation by 6 algorithms in the changing sequence of firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 86

5.7 The distribution of the old population ('Old pop'), the extreme point ('ExPoint'), the detective population ('DetePop') and transferred population ('TrPop') via PS expansion on F2 and WFG1 at the first generation when increasing the NObj from 2 to 3. . . . . 90

5.8 The distribution of the old population ('Old pop') and transferred population ('TrPop') via PS contraction on F2 and WFG1 at the first generation when decreasing the NObj from 3 to 2. . . . . 91

5.9 Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by 6 algorithms in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 92

5.10 HV trajectories across the whole evolution process for the NObj changing sequence of '2-7-2 one by one'. . . . . 96

5.11 HV trajectories across the whole evolution process for the NObj changing sequence of '2-7-2 one by one'. . . . . 97

5.12 HV trajectories across the whole evolution process for the NObj changing sequence of '2-7-2 one by one'. . . . . 98

5.13 Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by 6 algorithms in the changing sequence of firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 99

5.14 Fitness landscape of bi-objective WFG1 problem with two variables ($x1$ and $x2$) where the sample step sizes for x1 and x2 are 0.005 and 0.1, respectively. . . . . 104

5.15 Fitness landscape of bi-objective WFG5 problem with two variables ($x1$ and $x2$) where the sample step sizes for x1 and x2 are 0.005 and 0.01, respectively. . . . . 105

5.16 Fitness landscape of bi-objective WFG9 problem with two variables ($x1$ and $x2$) where the sample step sizes for x1 and x2 are 0.002 and 0.004, respectively. . . . . 105

5.17 Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by DTAEA and DTAEAv1 in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 107

5.18 Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by KTDMOEA and KTDMOEAv1 in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 107

5.19 Friedman ranking among MHV of optimized solutions at the last generation by all algorithms in the changing sequences '2-3-5-7-5-3-2' and '7-5-3-2-3-5-7', respectively. . . . . 110

5.20 Friedman ranking among HV of optimized solutions at the last generation by 6 state-of-the-arts and 3 KTDMOEAs with different values of parameters theta (1, 2 and 4) in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 112

5.21 Distribution of populations obtained by all compared algorithm on WFG4 *at the first generation right after changes* when increasing the NObj from 2 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1). . . . . 114

5.22 Distribution of populations obtained by all compared algorithm on WFG4 *at the last generation after optimization* when increasing the NObj from 2 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1). . . . . 114

5.23 Distribution of populations obtained by all compared algorithm on WFG4 *at the first generation right after changes* when decreasing the NObj from 4 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1). . . . . 115

5.24 Distribution of populations obtained by all compared algorithm on WFG4 *at the last generation after optimization* when decreasing the NObj from 4 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1). . . . . 115

6.1 (a) Distribution of approximated population given 1000 generations on bi-F4; (b) Distribution of the old population ('Old pop'), i.e. reevaluating 'pop' obtained in (a) on the tri-F4, the extreme point ('ExPoint'), the detective population ('DetePop'), all used to find the expansion directions. Transferred solutions by KTDMOEA on F4 are still on the curve of 'Old pop', thus not being plotted to avoid solutions duplication, and failing to spread through the PF. . . . . 120

6.2 (a) Distribution of approximated population given 1000 generations on bi-F2; (b) Distribution of the old population ('Old pop'), i.e. reevaluating 'pop' obtained in (a) on the tri-F2, the extreme point ('ExPoint'), the detective population ('DetePop'), all used to find the expansion directions in the objective space, and transferred population('TrPop') via PS expansion in KTDMOEA. . . . . 121

6.3 Distribution of approximated population given 1000 generations on tri-F3 and solutions transferred by PS contraction of KTDMOEA at the first generation right after decreasing NObj from 3 to 2 on F3. . . 121

6.4 Distribution of approximated population given 1000 generations on tri-F2 and solutions transferred by PS contraction of KTDMOEA at the first generation right after decreasing NObj from 3 to 2 on F2. . . 122

6.5 Method of learning the candidate expansion and contraction directions for increasing and decreasing NObj, respectively: (a), **ev** is the eigenvector of the found $PS_{m=2}$ and **d**1 to **d**3 are three directions that are perpendicular to $PS_{m=2}$, (i.e. **ev** * $\mathbf{d}_i = 0$, where $i = 1, 2, 3$); **d**1 to **d**3 are the three candidate expansion directions (b), **ev1** and **ev2** are two eigenvectors of $PS_{m=3}$, which are also the candidate contraction directions. . . . . 124

6.6 Friedman ranking among MHV of obtained solutions at the first generation by 5 algorithms when $\tau_t = 50$, 100 and 200, respectively. . . . 134

6.7 Friedman ranking among GD of obtained solutions at the first generation by 5 algorithms when $\tau_t = 50$, 100 and 200, respectively. . . . . 135

6.8 Transferred solutions by KTDMOEA and LEC on two problem F2 and F4 directly after increasing the NObj from 2 to 3. The MGD and MHV values of both approaches are also put in the title of each subfigure. . . . . 138

6.9 Transferred solutions by KTDMOEA and LEC on two problem F2 and F3 directly after decreasing the NObj from 3 to 2. The MGD and MHV values of both approaches are also put in the title of each subfigure. . . . . 138

6.10 Friedman ranking among MHV of solutions after *gap* generations' optimization by 5 algorithms in the case of $\tau_t$=50 and when *gap* = 5, 25 and 50, respectively. . . . . 145

6.11 Friedman ranking among GD of solutions after *gap* generations' optimization by 5 algorithms in the case of $\tau_t$=50 and when *gap* = 5, 25 and 50, respectively. . . . . 147

6.12 Friedman ranking among MHV of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=100. . . . . 148

6.13 Friedman ranking among GD of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=100. . . . . 148

6.14 Friedman ranking among MHV of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=200. . . . . 148

6.15 Friedman ranking among GD of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=200. . . . . 149

6.16 HV trajectory of sampled generations (1, 5, 25 and 50) after each change over evolution on problems F4, WFG3, WFG6 and WFG9, when $\tau_t = 50$. . . . . 169

6.17 HV trajectory of sampled generations (1, 5, 25, 50 and 100) after each change over evolution on problems F4, WFG3, WFG6 and WFG9, when $\tau_t = 100$. . . . . 170

6.18 HV trajectory of sampled generations (1, 5, 25, 50, 100 and 200) after each change over evolution on problems F4, WFG3, WFG6 and WFG9, when $\tau_t = 200$. . . . . 171

6.19 Parallel coordinates of solutions obtained at the first generation after increasing the NObj from 2 to 7 on 7-objective WFG4. . . . . 172

6.20 HV trajectory of sampled generations (1, 5, 25, 100 and 200) after each change over evolution on one problem F3. . . . . 173

6.21 Friedman ranking among MHV of optimized solutions at the last generation by 6 state-of-the-arts and 3 LECs with different values of parameters theta (1, 2 and 4) in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 175

6.22 Friedman ranking among running time obtained by all compared algorithm in changing sequence of Equation (6.1). Here, larger ranks mean larger running time. . . . . 177

# List of Tables


3.1 Different configurations (C1 to C8) with different settings of $n_t$, $\tau_t$ and $\tau_T$ in all benchmark functions. . . . . 33

3.2 MIGD values of obtained solutions in the first generation after changes for Tr-RMMEDA and RMMEDA. Note that only C4, C6 and C8 are selected here because at each parameter setting, the population is given enough generations (i.e., 50) to ensure the population converges such that the failing of transfer is not due to transferring bad solutions. 34

3.3 MIGD of combined solutions and transferred solutions. . . . . 35

3.4 MIGD values for Gaussian kernel-based and linear kernel-based Tr-RMMEDA. . . . . 39

3.5 MIGD values for the original and improved Tr-RMMEDA . . . . . 41

3.6 MIGD values of four DMOEAs with the optimization algorithm RMMEDA on all benchmark problems. . . . . 43

3.7 MIGD values of four DMOEAs with the optimization algorithm NSGA-II on all benchmark problems. . . . . 44

3.8 MIGD values of four DMOEAs with the optimization algorithm MOPSO on all benchmark problems. . . . . 45

4.1 Computation time of each process of Tr-RMMEDA with one run, one change and one parameter on problem DMOP2. . . . . 51

4.2 Mean and standard deviations of MIGD values of transferred solutions and those solutions optimized using the computation time of transfer after being randomly generated. . . . . 53

4.3 Mean and standard deviations of MIGD values of optimized solutions with transfer and random solutions that first iterate for some generations which consume the same computation time as transfer optimization, and then are optimized the same generations as those with transfer. . . . . 55

4.4 Mean and standard deviations of computation time (in seconds) of three optimization methods for minimizing the distance of source and target problems in the latent space with 20 environment changes for all problems with all parameter settings. . . . . 59

4.5 Mean and standard deviations of MIGD values of transferred solutions in the first generation after change obtained by three Tr-RMMEDA variants with different inner optimization algorithms. . . . . 61

4.6 Mean and standard deviations of MIGD values of optimized solutions in the last generation after change obtained by three Tr-RMMEDA variants with different inner optimization algorithms. . . . . 63

5.1 Problem features of F1-F4 and WFG1-WFG9. . . . . 78

5.2 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 83

5.3 Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 84

5.4 Mean and standard deviation values of MMS metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 85

5.5 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 87

5.6 Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 88

5.7 Mean and standard deviation values of MMS metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 89

5.8 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms at the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 93

5.9 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 94

5.10 Mean and standard deviation values of MMS metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 95

5.11 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 100

5.12 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 101

5.13 Mean and standard deviation values of MMS metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one. . . . . 102

5.14 Mean and standard deviation values of MHV metric for solutions obtained by two pairs of comparison under different optimization algorithms at the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 107

5.15 Mean and standard deviation values of MGD metric for solutions obtained by two pairs of comparison under different optimization algorithms in the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 108

5.16 Mean and standard deviation values of MMS metric for solutions obtained by two pairs of comparison under different optimization algorithms in the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 108

5.17 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the last generation after optimization on all DMOPs with a changing NObj, where changes of the NObj following the pattern in equation 5.9. . . . . 110

5.18 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the last generation right after optimization on all DMOPs with a changing NObj, where changes of the NObj follow the pattern in equation 5.10. . . . . 111

5.19 Mean and standard deviation values of MHV metric for optimized solutions obtained by 5 state-of-the-arts and three KTDMOEAs with different values of parameters ($\theta = 1$, 2 and 4) at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one . . . . . 113

6.1 The number of cases when LEC obtains significant better quality of solutions than (+), worse than (-) and equal to (=) corresponding algorithms at the first generation measured by metrics MHV and MGD for all changes ('overall'), the changes of increasing NObj from 2 to 7 ('InNObj') and decreasing NObj from 7 to 2 ('DeNObj'), respectively. 135

6.2 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj . . . . . 136

6.3 Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj. . . . . 139

6.4 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . 140

6.5 Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . 141

6.6 Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing number of objectives, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 142

6.7 Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . 144

6.8 The number of cases when LEC obtains significant better quality of solutions than (+), worse than (-) and equal to (=) corresponding algorithms after optimization of different generations under different frequency of changes (i.e. $\tau_t$=50, 100 and 200) measured by metrics MHV and MGD averaged across all changes ('Overall'), the changes of increasing NObj from 2 to 7 ('InNObj') and the changes of increasing NObj from 2 to 7 'DeNObj', respectively. . . . . 146

6.9 The p-values for significant differences in average value detected by Friedman test for MHV and MGD of different *gap* and $\tau_t$. . . . . 147

6.10 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 150

6.11 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 151

6.12 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 153

6.13 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 154

6.14 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 155

6.15 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 156

6.16 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 157

6.17 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 158

6.18 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 159

6.19 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 160

6.20 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 161

6.21 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 162

6.22 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 163

6.23 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 164

6.24 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 165

6.25 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. 166

6.26 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 167

6.27 Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. . . . . 168

6.28 Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 6 and then decreasing it from 6 to 2, both two by two. . . . . 174

6.29 Mean and standard deviation values of MHV metric for optimized solutions obtained by 4 state-of-the-arts and three LECs with different values of parameters ($\lambda = popsize/10$, $popsize/20$ and $popsize/30$) at the last generation of all changes on all DMOPs with a changing number of objectives, when firstly decreasing the number of objectives from 7 to 2 and then increasing it from 2 to 7, both one by one . . . 176

6.30 Mean value of running time (in seconds) obtained by all compared algorithms when $\tau_t = 50$ in the changing sequence of Equation (6.1) under 31 runs. . . . . 178

# Chapter 1

# Introduction

This chapter gives an overview of this thesis regarding the research topic of knowledge transfer in dynamic multi-objective optimization. Specifically, section 1.1 details the motivations of why we investigate knowledge transfer to solve dynamic multi-objective optimization problems (DMOPs). Based on these motivations, what follows are proposed research questions of handling DMOPs via knowledge transfer, which are presented in section 1.2. Thereafter, a number of contributions of thesis are presented in section 1.3 after answering these research questions. Finally, section 1.4 presents the list of publications generated from this thesis.

## 1.1 Motivations

Dynamic multi-objective optimization problems (DMOPs) [4] are characterized as optimization problems with multiple objectives conflicting with each other and changing over time. This type of problem widely exists in real world problems, such as controlling [5], resource allocation [6] and scheduling [7, 8]. For example, in the control of crop-producing greenhouse [5], there are three time-varying objectives which should be optimized simultaneously. These objectives include maximal production of crops, minimal energy consumption to generate heat and minimal exhaust emission. All of these are affected by the time-changing temperature and sunlight.

The example given above illustrates a DMOP where the objectives change over time, but the number of objectives (NObj) remains the same. Another type of DMOPs where the NObj is time-dependent also widely exists in real-world scenarios [9–12]. For instance, in project scheduling, people usually urge to minimize the makespan and total salary cost simultaneously. However, given a tighter deadline to be met, the salary cost may not be important and would not be considered as an objective any longer under the new circumstance [9–11]. In some other examples, if an application is running on a system with a wired power supply, there is no need to consider the energy consumption [12]. However, once the system is detached from the wired power supply and only relies on batteries, minimizing the energy consumption becomes a new objective.

The research on solving DMOPs with fixed NObj has been a well established research area in past decades [13, 14]. The key challenge in dynamic mutli-objective optimization (DMO) with fixed NObj is how to constantly trace changing position and/or shape of Pareto fronts (PFs) and/or Pareto sets (PSs) before the next environment change [4, 15]. PS of a multi-objective optimization problem is the set of Pareto optimal solutions in the solution (decision) space. PF is the image of PS in the objective space. Formal definitions of PS and PF can refer to Section 2.1.1.

Aiming at this goal, researchers have proposed prediction-based methods [15–17]. This kind of methods predict what the good solutions in the next environment are after learning the regularity of the environment changes using different models, such as Autoregressive and Kalman filter. In most prediction-based approaches, it is implicitly assumed that the evolution of the solutions used to train and test the prediction model obeys a fixed independent and identical probability distribution (IID). However, this is not always true under dynamic environments in optimization, since the environmental changes may result in different evolution patterns over time. Consequently, the prediction model based on the incorrect assumption may cause inaccurate prediction of optimal solutions. Knowledge transfer, which does not make this assumption, is a good candidate for solving DMOPs if it can learn and exploit the relationship among different problems. In this thesis, we will propose knowledge transfer approaches that do not rely on predictions to solve, avoiding the issues

associated with non-IID of prediction-based methods.

In the field of machine learning, knowledge transfer is also called transfer learning [18], which is a kind of machine learning method that is able to transfer the knowledge from a source task to a target task. The inherent characteristic of knowledge transfer makes it intuitive to apply knowledge transfer to explore useful experience or information that have been obtained in one task/problem to solve another related task/problem. The reason is that some similar or related problems/tasks may share some common features, which help to transfer experience from one problem to help solving another problem. As a result, computational resources can be significantly saved when solving similar problems later.

The study of knowledge transfer in evolutionary computation originates from 90s of last century [19–21] in the form of case-based reasoning where solution information of the old problems, stored as cases in a case base and retrieved later to help solving a new problem. Recently, knowledge transfer starts to attract more and more attentions in evolutionary computation as transfer learning [18] becomes popular in machine learning field. Specifically, knowledge transfer has been leveraged to solve multi-tasking problems where multiple tasks (problems) are needed to be solved simultaneously, which induces the birth of evolutionary multi-tasking optimization (EMT). One of the earliest framework in EMT is by Gupta et. al [22] [23]. Subsequently, additional work around transfer optimization for EMT [24] has been proposed.

Additionally, knowledge transfer has also been adopted to solve dynamic optimization problems as it is able to predict what goods are in the next environment without assuming the IID as most prediction-based strategies. For example, transfer learning-based dynamic multi-objective optimization algorithms (Tr-DMOEAs) [25] might be one of the earliest attempt of transfer learning at DMO, where the main idea behind is to transfer solutions in the PF of the previous environment to generate an initial population for the next environment. Later on, other DMOEAs based on knowledge transfer [26, 27] using different transfer learning algorithms have been proposed for solving DMOPs. In most existing studies of using knowledge transfer into DMO, the specific knowledge to be transferred is the optimal solutions from historical problem-solving, i.e., they can be categorized as solution-based transfer optimization approaches. We will also focus on solution-based knowledge in this thesis, as the aforementioned RQs we are trying to answer are proposed based on existing solution-based transfer optimization approaches.

Although there are some transfer optimization studies focusing on solving DMOPs, few work has been conducted to leverage transfer optimization to solve dynamic single objective optimization problems. To the best of our knowledge, the only one work of using knowledge transfer to solve dynamic single problems is by Liu et.al [28]. Specifically, this approach trains a neural network to learn the transfer model of environmental changes from the previous environment to the current one. However,

such training relies on a mapping of fitness values between solutions of the previous and new environments that is not directly applicable to multiple objectives. More details about the approach can be found in Section 2.2.3.

Even though knowledge transfer has achieved initial successes in evolutionary computation, especially when solving DMOPs, it is intuitively impossible that any transfer methods can be used at any cases, since transferring knowledge from an unrelated problem would be harmful to the optimization of the target problem. Moreover, the results from [25] confirm the intuition and show that knowledge transfer does not always work well on all problems with different types of changing PSs/PFs and different severity of change. It is therefore important to understand when and how knowledge transfer does not work well, since it enables people have a better understanding in which case transfer should be used and avoided and on which way knowledge should be transferred, which is essential for successfully applying transfer in optimization, such that solutions with better quality can be achieved by using transfer. However, there is a lack of such understanding in the literature.

Moreover, knowledge transfer, as an extra component embedded into normal process of evolutionary algorithms, would inevitably induce extra computational budget. Experimental studies of [25] show that the embedded knowledge transfer in DMO consume extra computation budget. However, it is unclear how much computational budget the knowledge transfer costs, especially in dynamic multi-objective optimization. Moreover, there is also a lack of investigation on whether extra computational budget caused by transfer is acceptable. In other words, whether using the extra cost of knowledge transfer for optimization from scratch would result in better solution quality than optimization with transfer. Few work has been conducted to study all of these aspects of knowledge transfer.

Even though many studies have been conducted to address DMOPs with a fixed NObj, few studies have been done to solve another type of DMOPs where the NObj changes over time, which also exist in many real-world scenarios, as discussed above. Moreover, few study has been conducted to investigate using knowledge transfer to tackle DMOPs with a changing NObj. One of the pioneer for studying changing NObj is the discussion of objective increment and replacement [29] where an inheritance strategy re-evaluating the current non-dominated solutions was used to deal with such changes. The most recent work is the proposal of the Dynamic Two Archive Evolutionary Algorithm (DTAEA) [3], in which two co-evolving populations, i.e, a convergence archive (CA) and a diversity archive (DA) are simultaneously maintained during the evolution. Whenever environmental changes occur, CA and DA are reconstructed to preserve as much convergence and diversity as they can in the new environment.

However, the knowledge transfer strategies in these algorithms for changing NObj simply copy previously optimized solutions and re-evaluate them in the new environment, which is incapable of providing enough diversity in DMOPs with a chang-

ing NObj containing more complex problem features including PF shapes (convex, discontinuous and mixed shape of convex and concave) and fitness landscapes (non-separability and deceptiveness). Moreover, the diversity strategies in most existing DMOEAs for changing PSs/PFs are unable to provide proper diversity, as they are designed to cover the search space as widely as possible rather than expanding or contracting the PSs that is required when solving DMOPs with a changing NObj. Therefore, it is an important problem of how to enhance the diversity of knowledge transfer when solving DMOPs with changing NObj containing more complex problem features.

It can be derived from the Karush–Kuhn–Tucker condition that PS of a continuous multi-objective optimization problem is a piecewise continuous (NObj-1)-dimension manifold, under mild conditions [30]. Following this fact, in DMOPs with a changing NObj, the dynamics can lead to the expansion or contraction of PS/PF manifold. It is therefore essential to study how to expand and contract PS when increasing and decreasing the NObj, respectively. Intuitively, determining the correct expansion and contraction directions might be one of the methodology to expand and contract the PS, since it enable us to generate solutions along the expansion and contraction directions, so as to further expand and contract the PS. Therefore, how to determine the correct expansion and contraction directions could be one of the most important problems.

## 1.2 Research Questions

Even though knowledge transfer has been widely and successfully used in many aspects of evolutionary computation, like multi-tasking optimization, dynamic multi-objective optimization with fixed NObj, etc, few work has been carried out to investigate in which case transfer should be used and avoided and on which way knowledge should be transferred such that knowledge transfer is beneficial to achieve solutions with better quality. In order to use knowledge transfer to fill the gap that most prediction strategies in DMO assume the IID, we have the first research question:

- **Research question 1 (RQ1)**: When and how to transfer knowledge when solving DMOPs with fixed NObj while changing PSs and/or PFs position and/or shape?

Considering that knowledge transfer is an extra component embedded into normal process of evolutionary computation, it would inevitably induce extract computational budget. It is unclear what the computational cost of knowledge transfer is. The original aim of using transfer optimization in DMO is to find good solutions such that the population can reach the optimum as soon as possible in the

optimization process. However, if the transfer process itself is very time consuming, the original aim of using transfer is defeated by its computational cost. It is unclear if the computational cost of transfer is acceptable in view of this. Moreover, it is unclear whether the efficiency of knowledge transfer could be enhanced under the condition that transferred solution quality is not degenerated. Therefore, we have the second research question.

- **Research question 2 (RQ2)**: What is the computational cost of knowledge transfer and could it be so high as to defeat the purpose of using knowledge transfer in DMO to find good solutions as quickly as possible when changes occur? How to improve the efficiency of knowledge transfer in DMO while preserving the quality of solutions found when solving DMOPs with fixed NObj while changing PSs and/or PFs position and/or shape?

Even though knowledge transfer has been widely studied in solving DMOPs with a fixed NObj, few studies have been carried out to solve DMOPs with a changing NObj, no less than research involving knowledge transfer. It is therefore significant to study how to use knowledge transfer to solve DMOPs with a changing NObj. Furthermore, one representative of state-of-the-art for dealing with changing NObj is proved to be incapable of providing proper and enough diversity in Section 5.1 when solving DMOPs with a changing NObj containing more complex problem features. These complex problem features include PF shapes (convex, discontinuous and mixed shape of convex and concave) and fitness landscapes (nonseparability and deceptiveness). It is therefore important to improve the diversity right after changes when solving such problems. And for the optimization of DMOPs, solutions with good quality are also urged after optimization for any optimization problems. Moreover, determining PS expansion and contraction directions is an essential step to expand and contract PSs when solving DMOPs with a changing NObj. Generally, diversity is the appearance of a population in the objective space of a problem. As knowledge transfer influences the initial population after an environmental change, it has an impact on the diversity of such a population.

Note that different means of determining the PS expansion and contraction directions might be available, we split the research question of 'how to determine the PS expansion and contraction directions' into the third and fourth research questions such that the PS expansion and contraction directions are determined in different ways.

Therefore, we propose the third research question from the perspective of heuristic when determining the directions.

- **Research question 3 (RQ3)**: How to determine PS expansion and contraction directions in a heuristic way to expand and contract PSs when solving DMOPs

with a changing NObj, so as to enhance diversity via knowledge transfer right after changes and further help the optimization itself?

Considering that there might be different way from the heuristic in obtaining the expansion and contraction directions, it would be beneficial to learn promising directions. Therefore, we have the fourth research question.

- **Research question 4 (RQ4)**: How to learn the PS expansion and contraction directions when solving DMOPs with a changing NObj, so as to expand and contract PS right after changes and further help the optimization itself?

## 1.3 Contributions

This thesis is dedicated to answering the research questions in 1.2, leading to the following contributions.

1. We have advanced the state-of-the-art of knowledge transfer in DMO via computational studies on the performance of knowledge transfer in DMO after answering the **first research question (RQ1)** when and how to transfer in DMO. Specifically, we select transfer learning-based dynamic multi-objective evolutionary algorithm (Tr-DMOEA) as a representation algorithm of state-of-the-art transfer-based DMOEA. We test Tr-DMOEA on a series of benchmark problems with different problem features and dynamic characteristics, trying to figure out when transfer achieves better solution quality. A mathematical proof is presented to further explore why transfer works or fails, which derives the answer of how to transfer. Thereafter, this thesis proposes an approach to transfer when it is beneficial. The specific criterion for determining when transfer works is that transferred solutions have better quality than optimal solutions obtained in the previous environment (Chapter 3).

2. To address the **second research question (RQ2)**, we carry out a computational study on the effectiveness of knowledge transfer in the state-of-the-art transfer-based algorithm to reveal what the computational cost of knowledge transfer is in DMO. Similarly, we select Tr-DMOEA as a representation algorithm of state-of-the-art transfer-based DMOEA to figure out the efficiency of knowledge transfer and which component consumes mostly if it is costly. The computational results show that the knowledge transfer in Tr-DMOEA is indeed time costly. Then, to verify whether the computational cost of transfer is acceptable, a computational study is further conducted through comparing optimization with transfer and using computational cost of transfer to optimize randomly generated solutions. Results show that optimization from scratch

using transfer cost obtains better solution quality, which demonstrates that the computation time saved in optimization is defeated by computationally expensive transfer. What follows is two more efficient alternatives of the most costly component introduced to improve the efficiency without degenerating solution quality by knowledge transfer (Chapter 4).

3. To address the **third research question (RQ3)**, a novel knowledge transfer-based method, called knowledge transfer dynamic multi-objective evolutionary algorithm (KTDMOEA), is proposed after considering the limitations of the state-of-the-art algorithm on problems with complex problem features in the fitness landscape (nonseparability and deceptiveness) and complex PF shapes (convex, discontinuous and mixed shape of convex and concave). In this method, PS expansion and contraction mechanisms are proposed to enhance diversity for dealing with changing NObj in DMOPs. (Chapter 5).

4. A new knowledge transfer-based approach, called learning to expand and contract PS (denoted as LEC) is proposed to determine the expansion and contraction directions in a learning way, so as to address the **fourth research question (RQ4)**. LEC automatically learns candidate directions for expansion and contraction by mining the data feature of the PS before the change. Then, it selects the most promising expansion and contraction directions according to certain criteria from these candidate directions after exploration about the new problem beyond the evaluation of existing candidate solutions on the new set of objectives. (Chapter 6).

## 1.4 List of Publications Generated from the Thesis

Published conference papers:

[*1] Gan Ruan, Leandro L Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. When and how to transfer knowledge in dynamic multi-objective optimization. In 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pages 2034-2041. IEEE, 2019. (Corresponding to Chapter 3).

[*2] Gan Ruan, Leandro L Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. Computational study on effectiveness of knowledge transfer in dynamic multi-objective optimization. In 2020 IEEE Congress on Evolutionary Computation (CEC), pages 1-8. IEEE, 2020. (Corresponding to Chapter 4).

Accepted journal papers:

[*3] Gan Ruan, Leandro L Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. Knowledge Transfer for Dynamic Multi-objective Optimization with a

Changing Number of Objectives. IEEE Transactions on Emerging Topics in Computational Intelligence. (Accepted) (Corresponding to Chapter 5).

[*4] Gan Ruan, Leandro L Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. Learning to Expand/Contract Pareto Sets in Dynamic Multi-objective Optimization with a Changing Number of Objectives. IEEE Transactions on Evolutionary Computation. (Accepted) (Corresponding to Chapter 6).

# Chapter 2

# Background and Literature Review

This Chapter presents the background on dynamic multi-objective optimization and knowledge transfer, as well as the literature review on application of knowledge transfer to evolutionary optimization.

## 2.1 Dynamic Multi-objective Optimization (DMO)

### 2.1.1 Basic Concepts of DMO

Before introducing dynamic multi-objective optimization, some concepts in evolutionary multi-objective optimization are firstly defined.

A minimization dynamic multi-objective optimization problem (DMOP) [31] can be mathematically formulated as follows:

$$\begin{cases} \min \ \mathbf{F}(\mathbf{x}, t) = \big(f_1(\mathbf{x}, t), \ldots, f_{m(t)}(\mathbf{x}, t)\big)^T \\ s.t. \ \ \mathbf{x} \in \Omega_x, \ t \in \Omega_t, \end{cases} \tag{2.1}$$

where $\Omega_x \subseteq R^n$ is the decision (variable) space; $t$ is the discrete time instance; $\Omega_t \subseteq R$ is the time space. $\mathbf{F}(\mathbf{x}, t) : \Omega_x \times \Omega_t \to R^{m(t)}$ is the objective function vector that evaluates a candidate solution $\mathbf{x} = (x_1, ..., x_n)$ at time $t$. $m(t)$ is the number of objective at time $t$.

**Definition 1.** *Pareto dominance: Supposing that* $\boldsymbol{p}$ *and* $\boldsymbol{q}$ *are any two individuals in a population at time* $t$, $\boldsymbol{p}$ *is said to dominate* $\boldsymbol{q}$, *written as* $f(\boldsymbol{p}) \prec f(\boldsymbol{q})$, *if* $f_i(\boldsymbol{p}) \leq f_i(\boldsymbol{q}), \forall i \in 1, 2, ..., m(t)$ *and* $f_j(\boldsymbol{p}) < f_j(\boldsymbol{q}), \exists j \in 1, 2, ..., m(t)$.

**Definition 2.** *Pareto optimal: a solution* $\boldsymbol{p}$ *is said to be Pareto optimal if it is not dominated by any other solutions in* $\Omega$*.*

**Definition 3.** *Pareto set [31]: The Pareto set (PS) is a set of all Pareto optimal solutions in* $\Omega$ *and can be defined mathematically as follows:*

$$PS = \{x \in \Omega_x | x \text{ is Pareto optimal}\}. \tag{2.2}$$

**Definition 4.** *Pareto front [31]: The Pareto front (PF) is the image of PS in the objective space at time step* $t$ *and can be defined mathematically as follows:*

$$PF = \{\boldsymbol{F}(x) \in R^{m(t)} | x \text{ is Pareto optimal}\}. \tag{2.3}$$

Note that, when we mention PS or PF of problems from the perspective of problems without considering the algorithms to solve them, the PS and PF of mentioned here represents the true PS and PF of the problem. However, when designing an algorithm that attempts to approximate the PS or PF, we use the term "PF" to represent the non-dominated solution set in the objective space found by a specific evolutionary algorithm and "PS" to refer to the solution set in the decision (parameter) space constructed from the set of non-dominated solutions. The reason is that from the perspective of algorithms, the true PS or PF is unknown to the algorithm when solving a problem of interest.

### 2.1.2 DMO with Changing Pareto Sets (PSs) and/or Pareto Fronts (PFs)

The dynamics in dynamic multi-objective optimization problems (DMOPs) with fixed number of objective, i.e. $m(t)$ is a constant in equation (2.1), usually results in the change of the PS and/or PF shape and/or position [31]. Therefore, the key challenge of solving DMOPs is how to constantly track the changing shape and/or position of PSs and/or PFs. Many change response strategies have been proposed in the past decades [32, 33]. They can be classified into diversity strengthening that can be further divided into diversity introduction [7, 34–37] and diversity preservation [38], memory [39], prediction [15, 17, 40–45] and transfer optimization techniques [25–27].

#### 2.1.2.1 Diversity Strengthening

The reason for the poor performance of static multi-objective evolutionary algorithms on DMOPs is the loss of population diversity after environmental changes. In other words, the almost converged solutions in previous state are very likely to encounter degeneration under new environment since it is difficult for them to get

out of the sub-optimum. Therefore, the diversity strengthening technique [34, 38] becomes the most intuitive method. Further, this method can be classified into two categories:

- Diversity introduction: As its name suggests, this method is to explicitly introduce diversity to the population whenever there is an environmental change. After the developing of recent years, there are several methods. The earliest one is the random diversity introduction ([34]), randomly generating some or all solutions in the search space to introduce diversity. Later on, some techniques more advanced than random initialization, such as hyper-mutation approaches [7, 35, 36] and heuristics [37] were proposed to be carried on the current population to help them jump out of the current positions and adapt to the varied environment.
- Diversity preservation: The multiple population method is a representative of this sort of method, in which the diversity of population is maintained in the whole process of optimization through several sub-populations, rather than explicitly introducing diversity via extra solutions. One of the representative algorithms is by Goh et al. [38], where population diversity is preserved in a competitive and cooperative manner.

Even though diversity strengthening has achieved some successes in solving DMOPs, there are still some limitations. For diversity introduction, it is not trivial to decide how much diversity should be introduced whenever a change occurs. Insufficient diversity introduction would not help the current population jump out the current position and excessive diversity introduction would induce slow convergence, both of which would be detrimental to the fast adaptation to environmental changes. As for diversity maintenance techniques, it is challenging to determine how to divide the population into different sub-populations and how to design interactive strategies to transfer information among these sub-populations.

#### 2.1.2.2 Memory Strategies

Considering that there is some periodicity in present benchmark problems, researchers have been trying to record previously found good solutions in a memory and then copy them to the responding population whenever a new environment comes, which is classified into the memory strategies. For instance, a hybrid memory paradigm [39] is proposed in which a new population is re-initialized either by utilizing a local search on the elite solutions stored in the archive or by re-evaluating the previous non-dominated solutions. Memory strategies work well at the latter environmental changes of the evolution process since at the beginning of the evolution environments optimal solutions that would perform well in the current environment may not have been found yet.

#### 2.1.2.3 Prediction techniques

Different environments in DMOPs usually have correlations and relationships with each other, and the changes in DMOPs may also have the characteristic of regularity and periodicity. These features of DMOPs lay a foundation for the design of prediction-based responding techniques. Briefly speaking, based on already optimized solutions in previous environments, prediction [17, 40, 41] aims to learn a model, which can reflect the regularity of changing optimal solutions. After that, some expected optimal solutions in next environment are predicted using the learnt model, so as to quickly converge to the true optima. Two of the early representative algorithms of prediction strategies are the forward-looking prediction strategy (FPS) [40] and the population prediction strategy (PPS) [16], in which an autoregressive (AR) model [46] is used to take the role of prediction. The difference is that the optimal solutions after changes in FPS are directly predicted using the optimal solutions of previous environments as the input, while in PPS, the population is divided into two parts, i.e. centroid and manifold, to be predicted separately with the centroid and manifold of previous environments as the input. Besides the used of AR model as the prediction model, other models like Kalman filter [42] were also leveraged as the prediction model to develop prediction strategies in [17, 43].

FPS and PPS have achieved good performance regarding predicted solution quality, while both of them require much training time to train the AR model such that they do not have good performance at the beginning of the initial stage (i.e. environmental changes) of the whole evolution process. Bearing this in mind, researchers [15, 44, 45] subsequently proposed to apply the moving directions of centroids of previous two environments to predict the movement of centroid after the change, so as to predict the whole population.

Even though these prediction-based techniques [15, 17, 40–45] have gained apparent and competent successes in DMO, in most prediction-based approaches, it is implicitly assumed that the evolution of the solutions used to train and test the prediction model are independently and identically distributed according to a fixed probability distribution. However, this is not always true under dynamic environments in optimization, since the environmental changes may result in different evolution patterns over time. Consequently, the prediction model based on the incorrect assumption may cause inaccurate prediction of optimal solutions.

#### 2.1.2.4 Transfer Optimization Approaches

Transfer optimization [18, 47] or knowledge transfer, which does not require the assumption of independent and identical probability distribution in prediction, is a good candidate for solving DMOPs, since it can learn and exploit the relationship among different problems. In addition, knowledge transfer is able to transfer useful

information from one problem to help solving another related problem. Due to these properties of knowledge transfer, it has been successfully utilized to solve DMOPs [25–27]. The existing transfer optimization approaches will be further detailed in Section 2.2.

More comprehensive reviews about dynamic optimization in terms of benchmarking, methodological design, algorithmic evaluation and applications can be found in ([13, 14, 48]).

### 2.1.3 DMO with a Changing Number of Objectives

This section reviews the existing work on solving DMOPs with a changing number of objective. By looking at equation (2.1), we can see that the difference between (1) DMOPs with changing PSs shape and/or position of PSs and/or PFs and (2) DMOPs with a changing NObj is in $m(t)$ i.e. $m(t)$ is changing over time for DMOPs with a changing NObj rather than a constant for the other type of DMOPs. The study on DMO with a changing number of objective originates from the work of investigating the effects of the objective increment and replacement were discussed in [29]. Afterwards, although people have realized the importance of tackling DMOPs with a changing NObj and mention this concept in [49–51], few work existed studying this problem until recently [3].

#### 2.1.3.1 Inheritance Strategy for Objective Increment or Replacement of Objectives

The investigation [29] on the effects of the objective increment and replacement might be the earliest work of studying DMO with a changing number of objective. In this work, it has been found that some common members are shared by Pareto-optimal sets before and after the objective replacement. The authors proposed an inheritance strategy based on this observation, in which solutions found before the change are re-evaluated in the new objective functions and then non-dominated solutions are selected to adapt to the increment or replacement of objectives. Experimental studies in [29] show that the inheritance strategy is able to help multi-objective evolutionary algorithms (MOEAs) to achieve better optimized solution quality compared to optimization from scratch, through embedding the inheritance strategy into three MOEAs and comparing each of MOEA with the inheritance strategy to the same MOEA with a randomly initialized population.

#### 2.1.3.2 Comprehensive Analysis of DMOPs with a Changing NObj and Its State-of-the-art Solver

Recently, a comprehensive investigation was conducted on the problem properties of DMOPs with a changing NObjs and the challenges of solving it in [3]. Based on the challenges, the dynamic two archive evolutionary algorithm (DTAEA) was proposed.

**2.1.3.2.1 Problem properties for DMOPs with a changing NObj** Specifically, the problem properties of DMOPs with a changing NObj are summarized as follows:

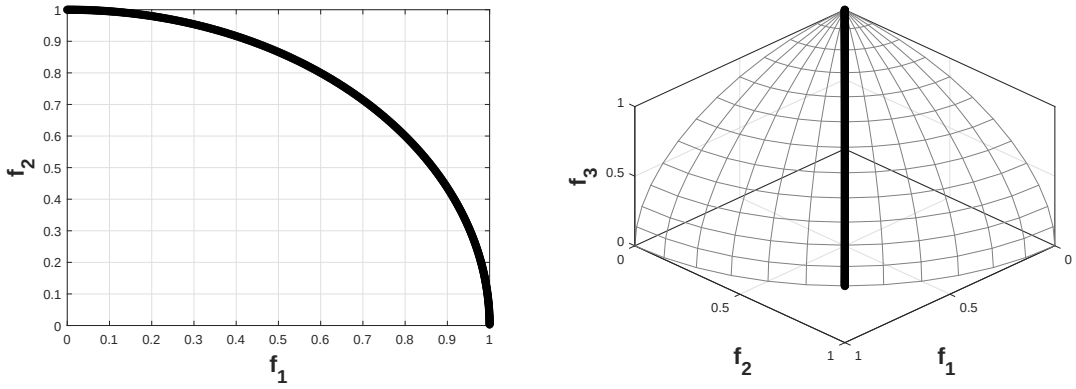
1. When increasing the NObjs, the PF and the PS at time step $t$ are a subset of those at time step $t+1$; on the contrary, when decreasing the NObjs, the PF and the PS at time step $t$ are a superset of those at time step $t+1$; This property was presented in the form of theorem and was proved in the Supplementary file of [3].

2. Instead of changing the position and/or the shape of the original PF and/or PS, increasing or decreasing the NObjs usually results in the expansion and contraction of the dimension of the PF or PS manifold, which is derived from a regularity property as follows. Under some mild conditions, the regularity property, induced from the Karush-Kuhn-Tucker condition [52], means that the PF and PS of a continuous $m$-objective MOP is an $(m-1)$-dimensional piecewise continuous manifold in both the objective space and the decision space.

The work in [53] might be the first attempt at showing that surface of PS is at most an (m - 1) dimensional surface in the decision space. Many efficient MOEAs were developed after considering this this regularity property explicitly (e.g., [30, 54–56]) or implicitly (e.g., [57–59]) in their implementation. Additionally, the above statement might be firstly proposed in an explicit manner in [30] and used to develop the regularity model-based multiobjective estimation of distribution algorithm. Continuous DMOPs with the regularity property was also considered in [3], according to which the $PF_t$ and $PS_t$ of the DMOP at time step $t$ is a (m(t) - 1)-dimensional piecewise continuous manifold in both objective space and decision space. Following [3], this thesis has the same consideration. According to the regularity property in the second point above, the second property of DMOPs with a changing NObj can be easily derived.

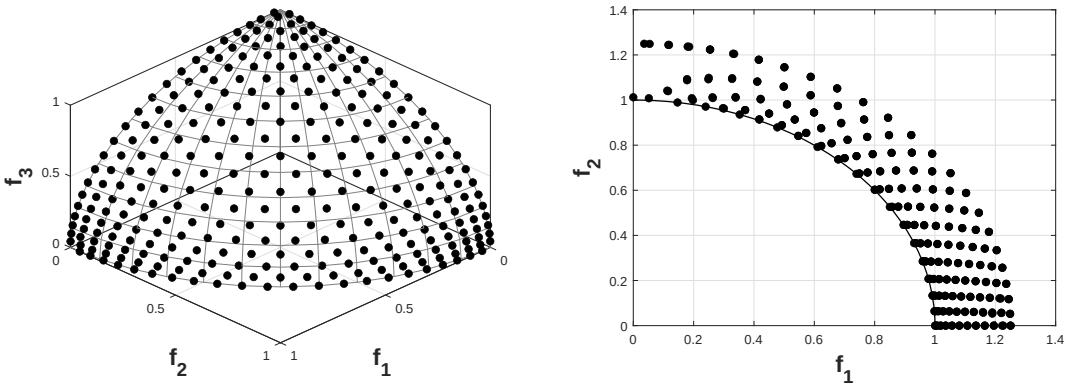
**2.1.3.2.2 Challenges of solving DMOPs with a changing NObj** In order to unfold the challenges of solving DMOPs with a changing NObj, a preliminary study was

conducted in [3] by using DTLZ2 [60] as the benchmark problem and multi-objective EA based on decomposition (MOEA/D) [57] as the algorithm for investigations. In this thesis, we only briefly introduce the experimental results to indicate the challenge of tackling changing NObj and specific details can be found in the original paper [3]. A good approximation to the PF by running MOEA/D on a bi-objective DTLZ2 instance for 200 generations and the corresponding population distribution re-evaluating the objective values when increasing the NObj from 2 to 3 are shown in Figures 2.1(a) and 2.1(b), respectively. Similarly, when decreasing the NObj, a good approximation to the PF by running MOEA/D on a tri-objective DTLZ2 instance for 200 generations and the corresponding population distribution re-evaluating the objective values when decreasing the NObj from 3 to 2 are shown in Figures 2.2(a) and 2.2(b), respectively.




(a) Approximation in the 2-D case (b) After increasing one objective

Figure 2.1: Comparison of population distributions when increasing the number of objectives on DTLZ2. Both figures are from Fig. 1 of [3].




(a) Approximation in the 3-D case (b) After decreasing one objective

Figure 2.2: Comparison of population distributions when decreasing the number of objectives on DTLZ2. Both figures are from Fig. 2 of [3].

It is clear from Figure 2.1 that when increasing the NObj, the population after the change is still on the true PF, meaning that population convergence is not affected. Therefore, the challenge of solving DMOPs when increasing NObj is how to propel the population jump out of the crowded areas, so as to improve population

diversity to cover the whole true PF. Similarly, when decreasing the NObj, it can be seen from Figure 2.2 that some solutions still stay on the PF with not so good diversity after the objective reduction; while the others are moved away from the PF. Therefore, the difficult in solving DMOPs when decreasing the NObj is how to pull unconverged solutions back to the PF with good diversity, so as to improve both population convergence and diversity. Besides, it can be also seen from Figures 2.1(a) and 2.2(a) that increasing and decreasing the NObj on DTLZ2 indeed results in the expansion and contraction of PF.

#### 2.1.3.2.3 Dynamic Two Archive Evolutionary Algorithm (DTAEA)

**Overall Frameworks** Bearing these challenges in mind, the authors in [3] proposed dynamic two archive evolutionary algorithm (DTAEA) to tackle DMOPs with a changing NObj, in which two complementary populations, covergence archive (CA) and diversity archive (DA), are simultaneously maintained in the evolution process. The CA is utilized to provide a constantly competitive selection pressure towards the optima while the DA is used to maintain diversified solutions as much as possible. At each iteration of the evolutionary algorithm, if a environmental change occurrs, mating parents are separately selected from the CA and the DA for offspring reproduction according to certain mechanism. Then, the offspring is used to update the CA and DA to make them preserve solutions with convergence and diversity separately.

**NObj Change Response** On the other hand, whenever environmental changes occur, CA and DA are reconstructed to preserve as much convergence and diversity as they can in the new environment. More specifically, when increasing the NObj, considering that population convergence is not degenerated at all, all solutions in the old CA are copied to the new CA. Then, in order to obtain as much diversity as possible, totally randomly generated solutions are put into the new DA via employing the canonical Latin hypercube sampling (LHS) method [61] to uniformly sample random solutions from the decision space. When decreasing the NObj, nondominated solutions of the old CA are all copied to the new CA to preserve population convergence. If the new CA is not full, the polynomial mutation is carried out on some selected good nondominated solutions to generate some mutants to fill the gap in the new CA. Dominated solutions of the old CA are all copied to the new DA. If the DA is not full, solutions are also randomly and uniformly sampled using LHS method in the decision space. Considering that DTAEA involves copying solutions from the previous environment, DTAEA can be seen as a kind of knowledge transfer-based algorithm, as it makes use of knowledge acquired from previous problem. The specific analysis on the limitations of DTAEA will be presented in Section 5.1.

**Density Estimation Method** To estimate the density of each solution in the population, $N(t)$ uniformly distributed weight vectors are generated following the method in [62] which are specified as $\mathbf{W}(t) = \{w^1(t), ..., w^{N(t)}(t)\}$. Therefore, the objective space is divided by these weight vectors into $N(t)$ subspaces, i.e., $\delta^1(t), ..., \delta^{N(t)}(t)$. A subspace $\delta^i(t)$ ($i \in \{1, ..., N(t)\}$) is defined as:

$$\delta^i(t) = \{\mathbf{F}(\mathbf{x}, t) | \langle \mathbf{F}(\mathbf{x}, t), w^i(t) \rangle \leq \langle \mathbf{F}(\mathbf{x}, t), w^j(t) \rangle\}, \tag{2.4}$$

where $j \in 1, ..., N(t)$ and $\langle \mathbf{F}(\mathbf{x}, t), w^i(t) \rangle$ is the perpendicular distance between F(x, t) and the reference line formed by the origin and $w^i(t)$.

After determining subspaces, each solution in the current population is associated with a unique subspace according to its position in the objective space. Specifically, for a solution $\mathbf{x}$, the index of its associated subspace is calculated as:

$$l = \underset{i \in \{1,...,N(t)\}}{\arg\min} \langle \mathbf{F}'(\mathbf{x}, t), w^i(t) \rangle, \tag{2.5}$$

where $\mathbf{F}'(\mathbf{x}, t)$ is the normalized objective vector of $\mathbf{x}$, whose calculation can be found in [3]. Based on the relationship between solutions and subspaces, the density of a subspace is evaluated as the number of solutions in it.

**Update Mechanism of the CA** Subsequently, the offspring population $A$ is used to update the $Pop$ via the process of CA update in DTAEA. Firstly, $Pop$ and $A$ are combined and the nondominated sorting [4] is conducted on them to divide them into several non-domination levels, i.e. $F_1$, $F_2$ and so on. Later on, from the first level $F1$, each non-domination level is chosen one at a time to construct the new $Pop$. This procedure continues until the size of the new $Pop$ equals to or for the first time exceeds the population size $N$. Denote the last included non-domination level as $F_s$, solutions in $F_{s+1}$ are not added to the $Pop$. If the number of solutions in non-domination level from 1 to $s$ is equal to $N$, this update process stops. Otherwise, the density estimation method introduced above is used to estimate the density information of the current population. Then, one worst solution from the most crowded subspace is removed at a time until the population is trimmed to its predefined threshold. For a subspace $\delta^i(t)$, the the worst solution $\mathbf{x}^*$ is defined as:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \delta^i(t)}{\arg\max} \{g^{tch}(\mathbf{x} | w^i(t)), \mathbf{z}^*(t)\}, \tag{2.6}$$

where

$$g^{tch}(\mathbf{x} | w^i(t), \mathbf{z}^*(t)) = \max_{1 \leq j \leq m(t)} \{|f_j(x, t) - z_j^*(t)| / w_j^i(t)\}, \tag{2.7}$$

where $\mathbf{z}^*(t)$ is the $t$-th ideal objective vector $\mathbf{z}^*(t) = (z_1^*(t), ..., z_{m(t)}^*(t))^T$, and $z_i^*(t) = \min f_i(\mathbf{x}, t), i \in \{1, ..., m(t)\}$.

## 2.2 Knowledge Transfer in Evolutionary Optimization

Inspired from cognitive capabilities of human-beings who are able to reuse a pool of knowledge from past experience when faced a never before encountered problem of interest, transfer optimization [47] is innovated and introduced to evolutionary computation as a novel optimization paradigm, so as to enhance its optimization efficiency and effectiveness, compared to optimization from scratch. One of the motivations behind this innovation is that real-world problems seldom exist in isolation. The experience or knowledge obtained from one problem-solving might be beneficial for the optimization of another related or similar problem, which is corresponding to human-beings gathering experience from past problems and leveraging the experience to solve new problems. Another one is that rapidly growing volume, variety, and complexity of the real-world problems of today poses enormous challenges on advanced and efficient problem-solving methodologies. The reason is that transfer optimization enables people to abstract useful information when solving problems with few volume, static or less complexity and adopt such experience to help solving similar but large, dynamic or complex problems in a more effective and more efficient way.

### 2.2.1 Basic Concepts of Transfer Optimization

As a newly proposed systematic evolutionary optimization paradigm, evolutionary transfer optimization has attracted enormous attentions in recent years, due to the reason that knowledge transfer is able to transfer useful information from one problem to help solving another related problem.

Conceptual realizations of the transfer optimization paradigm can be classified into three different categories according to what transfer optimization can be applied in practical scenarios [23].

- Multitasking transfer optimization : Given $K$ tasks $\mathcal{T}_1, ..., \mathcal{T}_K$, multitasking optimization aims to optimize these $K$ tasks simultaneously. Knowledge such as solution(s) obtained at any stage **while** solving the source tasks can be transferred to the target while solving the target task. Comprehensive reviews on evolutionary multitask transfer optimization can be found in [63, 64].

- Sequential transfer optimization: a strict assumption is made which is that when tackling task $\mathcal{T}_K$, previous $K-1$ tasks (i.e. $\mathcal{T}_1, ..., \mathcal{T}_{K-1}$) have been already solved. Different from multitasking transfer optimization, this type of transfer optimization paradigm aims at transferring the knowledge from source tasks **after** solving them to the target task. In other words, while solving the

target task, the sources tasks are assumed to have been already solved with optimal solutions.

- Multiform transfer optimization [65]: Different from multitasking transfer optimization and sequential transfer optimization where multiple different problems are associated, multiform transfer optimization tries to exploit alternate formulations of a single target problem of interest. Practical problems sometime involve different ways of formulating the same problem, such as selecting the fidelity of an approximate objective function in multi-fidelity optimization [66]. Therefore, the challenge is that it is difficult to decide which formulation is most appropriate for a particular problem, since different formulations may have pose different search behavior for algorithms. Mostly, multiple formulations in multiform transfer optimization could be regarded as multiple tasks to be solved simultaneously.

Considering that this thesis is about knowledge transfer in dynamic multi-objective optimization, we conduct the literature review of transfer optimization in a way that the transfer optimization is classified into stationary (i.e. static) and nonstationary (i.e. dynamic) scenarios. Given that in multiform transfer optimization, there are multiple formulations of the same problems to be solved, transfer algorithms for solving multi-tasking optimization problems can be also used to solve multiform optimization problems. The reason is that these formulations of the same problems can be just regarded as the multiple tasks to be solved simultaneously. Therefore, in the next context, transfer optimization in static scenarios will only review sequential transfer optimization and multi-tasking transfer optimization methods.

Even though knowledge transfer in DMO was not included into the above classification of transfer optimization, there exists connections and distinctions between transfer in DMO and sequential transfer optimization. Specifically, both of them assume when solving the problem of interest, there are a number of problems already been solved. Then, for transfer in DMO and sequential transfer optimization, the experience or knowledge obtained by previously solved problems/tasks can be used to assist the optimization of the target problem/task. The main difference between them is that there is a series of problems of interest to be solved at different time steps in transfer for DMO, while there is only one single target task in sequential transfer optimization. Then, transfer in DMO could be a series of sequential transfer optimization problems, where the problem at each single time step can be regarded as the target problem of sequential transfer optimization and the problems at previous time steps are the source problems. Therefore, the transfer optimization algorithms designed for solving sequential transfer optimization could be applied to address DMOPs.

In this thesis, knowledge transfer is regarded as a general concept of containing two meanings, i.e. transfer learning methods from machine learning field and the idea of transferring knowledge without explicitly using any transfer learning

methods. More specifically, for the former case, the usage of transfer learning in optimization is directly and explicitly adopting a transfer learning method, such as directly applying transfer component analysis [67] as the mean of transfer. For the latter case, the knowledge to be transferred could be solutions or some other information or knowledge obtained from solving one problem instance in helping solving another problem instance. If solutions are used to be transferred, the meaning of transfer is like conducting some process or mechanisms on the solutions from one problem instance to generate some new solutions for another related problem instance after considering the characteristic of the problem instance of interest. In other words, the latter cases is more like using transfer to make solutions obtained from solving one problem instance to adapt such that the adapted solutions are good for another related problem instance.

### 2.2.2 Knowledge Transfer in Static Evolutionary Optimization

The study of knowledge transfer in evolutionary computation can trace back to the last century [19–21] which is called case-based reasoning injection. Until recently, the term of knowledge transfer [23, 47] is emerging to induce the new optimization paradigm of evolutionary transfer optimization [47] in evolutionary computation community. According to how the knowledge is transferred among different tasks (problems), evolutionary transfer optimization is classified into two categories, i.e. explicit knowledge transfer and implicit knowledge transfer.

#### 2.2.2.1 Explicit Knowledge Transfer

In explicit knowledge transfer, knowledge from the source task(s) is explicitly transferred to help solving the target transfer via direct injection of knowledge or after some transformation on the knowledge before the injection. Explicit knowledge transfer can be applied to either multi-tasking transfer optimization problems or sequential transfer optimization problem. Therefore, the literature in explicit knowledge transfer is further classified into the categories of explicit multi-tasking transfer and explicit sequential transfer to be reviewed. Note that even though the term 'sequential transfer' is not explicitly mentioned in most explicit knowledge transfer for sequential optimization problems, they still belong to the category of explicit sequential transfer, since these transfer algorithms assume the source problems have been already solved with optimal solutions when conducting the knowledge transfer to solve the target problem.

**2.2.2.1.1 Explicit Sequential Knowledge Transfer** The work in [19] might be one of earliest applications of the knowledge transfer to evolutionary computation. In

this work, knowledge is transferred among the open shop scheduling problems via directly injecting solutions from the optimized scheduling problem with the number of 5% to 10% of population size into the population of solving a related scheduling problem. Later on, more depth studies [21] on selecting most similar solutions via the problem-dependent solution similarity measure were carried out to transfer knowledge among related problems. In [68], the idea of explicit transfer via autoencoding evolutionary search was used to solve heterogeneous problems where problem dimension or number of objectives may be different. More specifically, solutions at certain former generation of the source and target problem are regarded as as the input and output to train a simplified autoencoder to establish the bridge between the source and target problems. Then, optimized solutions from certain latter generation of the source task are regarded as the input of the trained autoencoder to output transferred solutions to be directly injected to the current population for the target task.

**2.2.2.1.2 Explicit Multi-tasking Knowledge Transfer** The same idea of using the autoencoder as the model to build the connection among source and target tasks was also adopted to solve multi-tasking problems [69]. To alleviate potential negative influence caused by knowledge transfer, Da et. al [24] designed an online learning method and problem similarity estimation. Specifically, probabilistic models of multiple solved source tasks are firstly constructed and then they are stacked together with the learned similarity as the coefficients of each model in the mixture model. Finally, solutions are sampled from the stacked mixture model to injected to the target task. A novel knowledge transfer algorithm was proposed aiming at selecting nondominated solutions and their several closest (based on the Euclidean distance) solutions as transferred solutions of the source tasks to achieve positive and effective knowledge transfer in multi-objective EMT [70]. In [71], the transfer was achieved by learning a representation latent space in EMT via considering decision space of each task as a manifold and then mapping manifolds of all tasks into a latent space. In such latent space, task relationships, represented as the joint mapping matrix, are learnt by increasing their similarities while keeping the necessary features for all tasks and the topology of each manifold. Then, the joint mapping matrix is used to generate transferred solutions for the target task for information transfer across different decision spaces during the evolutionary process. Explicit knowledge transfer was achieved [72] under the framework of EMT for solving capacitated vehicle routing problems after learning the mapping across different problems using the same idea to [68, 69] of autoencoder as the model to build the connection among source and target tasks.

**2.2.2.1.3 Limitations of Explicit Knowledge Transfer in DMO** Considering that there are lots of studies on explicit knowledge transfer and this thesis only concerns about knowledge transfer in dynamic multi-objective optimization, this paragraph presents the limitations of explicit knowledge transfer from the perspective of DMO.

Although the explicit knowledge transfer induce good results regarding solution quality in sequential transfer and multi-tasking transfer optimization, the essence of most explicit sequential transfer methods regarding directly injecting solutions from the source tasks to the target task is similar to the memory strategy for DMOPs with changing shape and/or positions mentioned in Section 2.1.2 and the inheritance strategy for objective increment or replacement of objectives in 2.1.3.1. In these explicit transfer method, solutions of the previously solved problems are solved in a memory and recovered to be injected to the problem-solving of the target. Accordingly, explicit transfer methods have the same limitations of memory strategy for DMOPs with changing shape and/or positions, which is that they work poorly when the previous problems of which stored solutions is not so similar to the current problem. As for DMOPs with a changing NObj, these explicit knowledge transfer also has the similar drawbacks to the inheritance strategy, which has been highlighted in Section 2.1.3.2, i.e. poor diversity when increasing NObj and both poor diversity and convergence when decreasing NObj. This can be derived from Figures 2.1(b) and 2.2(b), even though there is only one NObj change in these figures, since the drawback would be much severer in terms of problems with more than 3 objectives and/or changes more than one objectives.

As for the knowledge transfer methods via some transformation models to generate transferred solutions either for sequential transfer and multi-tasking transfer optimization, some of them have been utilized to solve DMOPs with changing shape and/or position of PSs and/or PFs, such as using the autoencoder as the transformation model in [73]. More details of using the transformation models to generate transferred solutions to tackle DMOPs with changing shape and/or position of PSs and/or PFs and the limitations of these methods will be elaborated in Sections 2.2.3.1 and 2.2.3.2. For DMOPs with a changing NObj, these transfer methods based on transformation models are unable to tackle the challenges of solving DMOPs with a changing NObj, since most of these transfer methods in multi-tasking and sequential transfer optimization do not consider solving problems with different NObj, thus without the capabilities of increasing proper diversity when the NObj increases and increasing both convergence and diversity when the NObj decreases.

#### 2.2.2.2 Implicit Knowledge Transfer

In implicit transfer, knowledge transfer is achieved through the interactions of genetic and cultural factors. Note that the implicit knowledge transfer is achieved by selecting mating parents that represent good quality of the solutions for the source and target problems respectively to conduct the crossover in the evolution process. Therefore, implicit knowledge transfer can only applied in the sense of multi-tasking optimization, since there are multiple tasks needed to be optimized concurrently and no target task is specified.

**2.2.2.2.1 Implicit Multi-tasking Knowledge Transfer** Inspired from the models of multifactorial inheritance where interactions of genetic and cultural factors influences complex developmental traits among offspring, multifactorial evolutionary algorithm (MFEA) [22] was proposed as one of the earliest framework of implicit EMT using one single population to solve multi-tasking optimization problems, where the crossover is conducted on mating parents selected from solutions that possess information of different tasks to implement the explicit knowledge transfer. Then, in [74], the concept of EMT was leveraged to tackle bi-level optimization problems concurrently where one optimization task is nested within another via implicit genetic transfer. Later on, in [75], the multi-objective version of MFEA was innovated to solve multi-objective EMT, i.e. solving multiple multi-objective optimization problems simultaneously via the implicit transfer. The second version of MFEA (MFEA-II) [76] was then proposed to conduct online learning and exploit similarities among multiple tasks via principled theoretical arguments through minimizing the tendency of harmful interactions between tasks.

In [77], MFEA was generalized to solve expensive optimization problems through transferring knowledge from computationally cheap optimization problems after incorporating decision variable translation and shuffling into original MFEA, which are designed for problems with different optima locations and different dimension of search space, respectively. An adaptive knowledge transfer strategy [78] via adaptive configuration of crossovers on different individuals in MFEA was put forward to solve both single and multiple optimization problems under the settings of EMT. The idea of MFEA was also generalized to tackle multiple vehicle routing problems with heterogeneous capacity, time window, and occasional driver in the framework of EMT [79], after adapting definitions of MFEA from continuous space to discrete space, such as integrating multiple vehicle routing problems with different properties into a unified search space. In [80], a coevolutionary multitasking framework embedded in particle swarm optimization was proposed to handle the problem in composites manufacturing industry, in which the knowledge transfer is achieved by updating the position of particles for the target task via adding the current global best position of the source target.

**2.2.2.2.2 Limitations of Implicit Knowledge Transfer in DMO** Given that there are lots of studies on implicit knowledge transfer and this thesis only concerns about knowledge transfer in dynamic multi-objective optimization, this paragraph presents the limitations of implicit knowledge transfer from the point view of DMO.

Even though implicit knowledge transfer has gained tremendous successes in multi-tasking optimization, they are inapplicable to DMOPs. The reason is that the knowledge transfer is achieved in the process of optimizing multiple tasks at the same time via conducting the crossover on mating parents from source and target tasks. However, in DMO, problems at previous environments have been optimized and the optimal solutions have been obtained, in which the intermediate solutions

are unable to join the evolution for the current problem.

### 2.2.3 Knowledge Transfer in Dynamic Evolutionary Optimization

This section reviews knowledge transfer-based approaches applied to dynamic evolutionary optimization. These approaches all belong to the category of evolutionary sequential optimization introduced in Section 2.2.1. In other words, they treat each pair of consecutive problem instances in an DMOP as a separate sequential optimization problem. Note that knowledge transfer in dynamic evolutionary optimization locates at the category of explicit transfer. The reason is that, in implicit transfer, transfer is achieved when simultaneously solving multiple problems/tasks by crossover on solutions from different problems/tasks, while problems of previous environments have been already solved in dynamic evolutionary optimization.

Although transfer optimization has recently achieved some successes in the field of EMT, it has rarely been studied in dynamic optimization and dynamic multi-objective optimization (DMO). To the best of our knowledge, the paper [28] is the only work of leveraging knowledge transfer to solve dynamic single objective optimization problems. In this work, a neural network is proposed to learn the transfer model of how the environment changes from the previous environment to the current one. Specifically, solutions from these two environments are clustered (in the solution space). In the previous environment, each local optimum forms a cluster center and then each solution from the previous environment is allocated to the nearest cluster. It is assumed that the local optima in the new environment is near to the local optima of the previous environment. Therefore, randomly generated solutions in the new environment are also clustered to the local optima of the previous environment. Then, clustered solutions from the previous and the current environment are paired based on their scaled fitness values. These paired solutions are regarded as the training samples with solutions from the previous and the current environments as the input and output, respectively. Lastly, the optimal solutions from the previous environment are regarded as the input of the trained neural network to output the solutions as the transferred solutions. Notably, this kind of transfer method is unable to be directly applied to solving DMOPs. In particular, the clustering method cannot be directly used in DMO, since there are no local optima in multi-objective optimization. In addition, solutions for multi-objective optimization cannot directly use the pairing method in [28], since solutions' 'fitness' is an objective vector rather than a single value. Moreover, the assumption on the proximity of the previous and new local optima is rather strong, which might not hold in DMO.

It can be seen from the mathematical formulation of DMOPs in equation (2.1) that problems at different time steps only have different parameter $t$, showing that other parts of the problems are similar or the same. This foundation naturally enables us assume that the changing problems are related, there are good opportu-

nities for the application of knowledge transfer in DMO. Since the first introduction of transfer optimization in DMO and the proposal of transfer learning-based dynamic multi-objective optimization algorithms (Tr-DMOEAs) [25], more and more papers [26, 27, 73, 81] recently apply knowledge transfer to solve DMOPs. This section will present these knowledge transfer-based algorithms in details.

#### 2.2.3.1 Transfer Learning-based Dynamic Multi-objective Evolutionary Algorithms (Tr-DMOEAs)

Almost all knowledge transfer methods applied in dynamic optimization explicitly leverage transfer learning [18] techniques from machine learning community to conduct the knowledge transfer, some of which combine normal evolutionary techniques like convergence or diversity preservation. One of the earliest work of using knowledge transfer in DMO is the introduction of transfer learning-based dynamic multi-objective optimization algorithms (Tr-DMOEAs) [25]. The main idea behind Tr-DMOEA is to utilize transfer component analysis [67] to transfer solutions in the PF of the previous environment to generate an initial population for the next environment via mapping the source and target optimal solutions into the latent space. In this section, the transfer component analysis will be first introduced.

**2.2.3.1.1 Transfer component analysis (TCA)** Domain adaptation methods, as a branch of transfer learning [18], aim at finding a good feature representation to decrease the difference between the distributions of source and target domains when transferring the knowledge acquired from the source domain(s) to help addressing the target task in the target domain. There are lots of similarity measures in the literature of transfer learning, one of which is the maximum mean discrepancy (MMD) [82]. MMD has been widely used in many transfer learning algorithms due to the simplicity and accuracy as a nonparametric distance estimation method in the reproducing kernel Hilbert space (RKHS).

Let $\mathcal{P}$ and $\mathcal{Q}$ are distributions defined on a domain $\mathcal{X}$. Let $X = (x_1, ..., x_m)$ and $Y = (y_1, ..., y_n)$ be random variable sets, drawn independently and identically distributed (i.i.d.) from $\mathcal{P}$ and $\mathcal{Q}$ respectively. Let $\mathcal{F}$ be a class of functions $f : \mathcal{X} \to \mathbb{R}$, then the empirical estimate of distance between $\mathcal{P}$ and $\mathcal{Q}$ defined by MMD is as follows:

$$MMD(\mathcal{F}, \mathcal{P}, \mathcal{Q}) := \sup_{f \in \mathcal{F}} \left( \frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{j=1}^{n} f(y_j) \right). \tag{2.8}$$

By the fact that in a RKHS, function evaluation can be written as $f(x) =$

$\langle \phi(x), f \rangle$, where $\phi(x) : \mathcal{X} \to \mathcal{H}$, MMD can be rewritten as follows:

$$MMD(\mathcal{F}, \mathcal{P}, \mathcal{Q}) := \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(x_i) - \frac{1}{n} \sum_{j=1}^{n} \phi(y_j) \right\|_{\mathcal{H}}^2. \tag{2.9}$$

i.e. the second normal form of $X$ and $Y$ in the RKHS.

Let

$$k_{x_i, y_j} = \kappa(x_i, y_j) = \langle \phi(x_i), \phi(y_j) \rangle = \phi(x_i)^T \phi(y_j), \tag{2.10}$$

where $\kappa(.,.)$ is a kernel function and $\phi(.)$ is a feature mapping function. Here, the Gaussian kernel function is used. MMD can be further rewritten after expanding equation (2.9) using so-called "kernel trick" [83] as follows:

$$MMD(\mathcal{F}, \mathcal{P}, \mathcal{Q}) = \mathbf{tr}(KL), \tag{2.11}$$

where

$$\mathbf{K} = \begin{pmatrix} K_{X,X} & K_{X,Y} \\ K_{Y,X} & K_{Y,Y} \end{pmatrix}, \mathbf{K}, \mathbf{L} \in \mathbb{R}^{(m+n)*(m+n)}, \mathbf{L}_{i,j} = \begin{cases} \frac{1}{m^2} & x_i, x_j \in X \\ \frac{1}{n^2} & y_i, y_j \in Y \\ -\frac{1}{mn} & \text{otherwise.} \end{cases}$$

A dimension reduction method denoted as MMD embedding (MMDE) was proposed by Pan et al. [84] to obtain a low-dimensional representation to reduce the difference between source and targets distributions, and preserve the main statistical properties of the source and targets data. However, In MMDE, the kernel function $\kappa$ is learned (or optimized) from the data, which is computationally expensive. Therefore, transfer component analysis was proposed by the authors in [67] and its semisupervised version of TCA transforms the problem of learning an entire kernel matrix $K$ to a low-rank matrix $\hat{W}$ instead. Suppose that $\hat{W}$ is a $(m+n) \times d$ matrix. For any vector $x$, let $\phi(x) = W^T \kappa_x \in \mathbb{R}^d$, where $\kappa_.$ is a feature mapping function. Let $\kappa_x = [\kappa(x_1, x), ..., \kappa(x_m, x), \kappa(y_1, x), ..., \kappa_(y_n, x)]^T$, and equation (2.11) can be transformed as follows:

$$MMD(\mathcal{F}, \mathcal{P}, \mathcal{Q}) := tr(\hat{W}^T KLK\hat{W}). \tag{2.12}$$

More details about the specific process of TCA can be found in [67] [25].

**2.2.3.1.2 Tr-DMOEAs** Once $\hat{W}$ is found, Tr-DMOEAs will initialize the population in the target environment with solutions whose objective vectors are close to that of any good solution from the source environment in the latent space. For that, it needs to find solutions $y_k$ whose objective vector is close to that of a solution $x_l$ from $PF$ of the problem in the source environment ($PF_s$), i.e., making the following formula minimal:

$$||\phi(x_l) - \phi(y_k)||. \tag{2.13}$$

In this thesis, we call the problem of searching for the solution $y_k$ of this formula 'inner problem' in transfer learning.

The specific process of Tr-DMOEAs is presented as follows. Whenever a change occurs, a population is randomly generated in the search space and evaluate it in the previous and current environments to get two sets of objective vectors. Then, these two objective vectors are input to TCA to learn the matrix $W$ such that they are close in the latent space. Afterwards, map the PF of the problem before the change $PF_t$ to the latent space using the feature mapping function $\phi(x)=W^T\kappa_x$. Lastly, for each optimal solution in $PF_t$, find a solution $x$ in the objective space of the problem after the change such that they are close in the latent space. Details of Tr-DMOEAs can be found in [25].

#### 2.2.3.2 Other Transfer-based Algorithms

In [73], the authors utilized a simplified autoencoder (AE) to acquire knowledge from previous two environments ($t-1$ and $t$) via regarding nondominated solution sets (**NDS**$s$) obtained in previous two environments ($t-1$ and $t$) as the input ($\mathbf{NDS}_{t-1}$) and output ($\mathbf{NDS}_t$) of the AE, and then transfer the knowledge through inputting NDS of the previous one environment ($\mathbf{NDS}_t$) to the AE to regard the output as the forecasted solutions for the new environment ($t+1$). A manifold transfer learning method was developed in [26] to decrease the difference between optimal solutions found in the previous environment and solutions to be found in the new environment after mapping them into the geodesic flow to achieve the knowledge transfer.

Another popular transfer learning technique TrAdaboost [85] was also used in DMO to transfer previously optimal solutions to generate an initial population under the assistance of a presearch strategy in [27] or knee point [81]. More specifically, in [27], a presearch strategy was used to select some solutions with good diversity, and then these solutions and optimal solutions from past environments are regarded as the target and source solutions, respectively, to train the TrAdaboost with a set of weak classifiers by regarding nondominated solutions and other solutions as the training samples with positive and negative labels (i.e. +1 and -1). After getting the parameters and weights of a set of weak classifiers in the training process, the TrAdaboost is adopted to classify randomly generated solutions in the current environment and regarding solutions with positive label as transferred solutions. In [81], the authors predicted the knee point at certain region of the problem after the change based on the knee point at the same region of the problem before the change after estimating the knee point moving direction by adding a deflection angle to the angle of moving direction of two knee points in previous two environments. Then, estimated knee points and randomly generated dominated solutions of the problems after the change are regarded as training samples with positive and negative labels for the target task. Similarly, the knee points and randomly generated dominated solutions of the problems before the change are regarded as training samples with

positive and negative labels for the source task. Afterwards, all these training samples from the source and target tasks are leveraged to train the TrAdaboost. Lastly, the method of producing the transferred solutions is the same to that in [27].

Even though these transfer learning-based algorithms obtained somehow good transferred solutions, it is not true that transfer always helps the optimization from their experimental results. There are still some problems where the transfer does not get better results. In addition, training the transfer learning models usually takes extra computational time, which has not been considered in the optimization process.

# Chapter 3

# When and How to Transfer in Tr-DMOEAs [1]

Even though transfer optimization algorithms have achieved initial success in obtaining good quality of solutions, experimental results in [25] showed that transfer optimization does not always work well. There are still some problems and/or cases where the transfer does not get better results. It is therefore important to understand why and when transfer optimization does not work well. Only after understanding it can we make some improvements regarding transfer optimization in DMO.

Considering the general process of transfer learning [18], there are three main components: (1) what to transfer; (2) when to transfer; (3) how to transfer. In DMO, an algorithm always obtains a set of good solutions at the end of each environment. Considering that problems at different environments are correlated and similar, there might be common knowledge shared by good solutions obtained in the previous environments and solutions to be found in the next environment. Therefore, in terms of the first question, good solutions found for the previous environment are typically aimed to be transferred to the next environment in DMO and have used as the knowledge to be transferred in [25, 86]. However, there are no straightforward answers to the other two questions.

This chapter thus performs an investigation to answer **Research question 1 (RQ1)** presented in Section 1.2: "When and how to transfer knowledge when solving DMOPs with changing PSs and/or PFs position and/or shape?" which includes the (2) and (3) sub-research questions. Then, a new method for transfer optimization in DMO is proposed after answering these two sub-research questions.

We utilize the state-of-the-art method transfer learning-based dynamic multi-objective evolutionary algorithms (Tr-DMOEAs) [25] as an example of transfer op-

[1] This chapter is written based on content from the published paper [1] as [*1] in Section 1.4.

timization algorithm to investigate the answers to these questions. Specifically, Section 3.1 investigates when transfer optimization works/fails through a computational study under the optimization framework of Tr-DMOEAs and finds that transfer fails (i.e. performs worse than just copying the solutions from the previous environment) on problems with fixed PS and small environmental changes. In Section 3.2, a mathematical proof that Gaussian kernels in equation (2.10) of Section 2.2.3.1 are not ideal is given to indicating that there is room for improvement if a more suitable kernel is chosen. Here, the Gaussian kernels are used to calculate the inner product of two solutions after mapping them from the objective space to the latent space. Then, we replace the Gaussian kernel with a linear one and verify whether this replacement improves quality of transferred solutions using an experimental study. Section 3.3 combines the answers of when and how to transfer to propose an improved version of Tr-DMOEAs, which considers both avoiding transfer on problems with fixed PS and small environmental changes and replacing the Gaussian kernel with a linear one. Lastly, the effectiveness of the improved version of Tr-DMOEAs over Tr-DMOEAs is verified through a computational study on a set of benchmark problems.

## 3.1 When Does Transfer Optimization Work/Fail in Dynamic Multi-objective Optimization?

This section analyzes when transfer works or fails in DMO, such that some potential improvements can be put forward.

### 3.1.1 Computational Investigation into Transfer in DMO

Generally, whenever a new environment change happens, there is a population that has already been optimized for the previous environment. To check whether transfer optimization works, we compare the quality of the following solutions on the new environment: (1) transferred solutions and (2) solutions copied from the previously optimized population. More specifically, we define that transfer works/fails as follows:

- Transfer works: (1) are of at least similar (and ideally better) quality than (2) in the new environment;
- Transfer fails: (1) have worse quality than (2) in the new environment.

In order to figure out when transfer works/fails, we conduct an experimental study to compare transferred solutions and solutions copied from the previously

optimized population. Since we utilize Tr-DMOEA as a specific transfer algorithm to investigate when transfer works/fails, most experimental setup follows Tr-DMOEA [25]. The specific used benchmark problems, algorithms, performance metric and parameter settings is shown as follows:

- Benchmark problems: Since the existing transfer optimization algorithm Tr-DMOEA [25] is selected as the transfer algorithm to answer the questions of when and how to transfer, we adopt the IEEE CEC 2015 benchmark problems [87] used in the work of Tr-DMOEA. In addition, these problems have different types of PSs/PFs changes, which are beneficial for us to computationally analyze and figure out when transfer optimization works or fails.

- Compared algorithms: We use one transfer algorithm and one static multi-objective evolutionary algorithm to generate transferred solutions and copied solutions from the previous environment, respectively. The state-of-the-art transfer algorithm Tr-DMOEA [25] is adopted. RMMEDA [30] is selected as the optimization algorithm. RMMEDA is a regularity model-based multi-objective estimation of distribution algorithm. It is able to make the population converge quickly before the next change, avoiding the cases that unconverged solutions affect the results.

- Performance metric: Inverted Generational Distance (IGD) [88] is used to compare the performance of two solutions sets. IGD can measure the diversity and convergence of a solution set found by an algorithm, so it can give us a comprehensive understanding about the performance of compared algorithms. MIGD [89] is a modified version of IGD, which is the average IGD values in all changes. The smaller the MIGD the better the algorithm.

- Parameter settings: There are 20 changes for each of the used benchmark problems. In Tr-DMOEA [25], there are eight different configurations (C1-C8) with different frequency of changes ($\tau_t$) and severity of changes ($n_t$), which are shown in Table 3.1. In order to study the effectiveness of Tr-DMOEAs in different severity of changes, three different severity of changes (i.e., $n_t = 10$, 1 and 20) are used. It is worth noting that Section 3.1 and Section 3.2 where the answers of when and how to transfer are computationally investigated, only C4, C6 and C8 are selected. The reason is that in these three parameter configurations (C4, C6 and C8), $\tau_t$ is set as 50, meaning that the population runs 50 generations within each change, which enables the population to converge before the next change. This allows us to ensure that any failing of the transfer is not due to transferring bad solutions. The population size is set as 200, as in Tr-DMOEA [25].

Here, MIGD values for Tr-RMMEDA and RMMEDA are compared. After each environmental change, RMMEDA gets an initial population found in last generation of the previous environment, while Tr-RMMEDA obtains an initial population

Table 3.1: Different configurations (C1 to C8) with different settings of $n_t$, $\tau_t$ and $\tau_T$ in all benchmark functions.

| | $n_t$ | $\tau_t$. | $\tau_T$ |
|---|---|---|---|
| C1 | 10 | 5 | 150 |
| C2 | 10 | 10 | 250 |
| C3 | 10 | 25 | 550 |
| C4 | 10 | 50 | 1050 |
| C5 | 1 | 10 | 250 |
| C6 | 1 | 50 | 1050 |
| C7 | 20 | 10 | 250 |
| C8 | 20 | 50 | 1050 |

$n_t$, $\tau_t$ and $\tau_T$ are the severity of change, frequency of change and maximum number of iterations, respectively.

through transfer optimization. These two initial populations are evaluated in the new environment and then used to calculate the IGD values. MIGD values for Tr-RMMEDA and RMMEDA are the average of IGDs under 20 environments. The comparison results are shown in Table 3.2, in which $Tr.$ and $Copy$ mean MIGD values for Tr-RMMEDA and RMMEDA respectively. The better value obtained by the algorithm is shown with dark background.

It can be observed from the table that transferred solutions are all worse than those from the previous environment on problems HE2, HE7 and HE9, no matter what kinds of changes are present in these benchmark problems. All HE problems have fixed POS. Therefore, it can be concluded that transfer optimization fails on problems with fixed POS. For other problems, transferred solutions are better than those copied from the previous environment when $n_t = 1$ (C6). Regarding other two kinds of changes (C4 and C8), cases that transferred solutions are better for C4 are more than those for C8. As a result, transfer optimization works better when the environment change is larger. To sum up, transfer optimization fails on problems with small changes and with fixed POS.

One reason for this conclusion is that optimal solutions copied from the previous environment will not become so bad in the new environment, when the changes are small. Furthermore, for problems with fixed POS, optimal solutions will always keep optimal. Besides, the existing transfer optimization method is not able to transfer good solutions from the source problem to make them still good in the target problem.

Table 3.2: MIGD values of obtained solutions in the first generation after changes for Tr-RMMEDA and RMMEDA. Note that only C4, C6 and C8 are selected here because at each parameter setting, the population is given enough generations (i.e., 50) to ensure the population converges such that the failing of transfer is not due to transferring bad solutions.

| Prob. | C4 | | C6 | | C8 | |
|---|---|---|---|---|---|---|
| Methods | Tr. | Copy | Tr. | Copy | Tr. | Copy |
| FDA4 | 0.082 | 0.129 | 0.098 | 3.987 | 0.081 | 0.081 |
| FDA5 | 0.389 | 0.179 | 1.294 | 5.211 | 0.325 | 0.115 |
| $FDA5_{iso}$ | 0.280 | 0.079 | 0.708 | 0.597 | 0.294 | 0.077 |
| $FDA5_{dec}$ | 0.645 | 0.242 | 3.006 | 4.901 | 0.583 | 0.119 |
| DIMP2 | 11.630 | 19.361 | 11.379 | 19.621 | 12.466 | 22.124 |
| DMOP2 | 1.531 | 0.798 | 35.126 | 40.256 | 1.165 | 0.216 |
| $DMOP2_{iso}$ | 0.002 | 0.002 | 0.082 | 0.111 | 0.002 | 0.002 |
| $DMOP2_{dec}$ | 1.044 | 2.533 | 5.182 | 61.288 | 1.028 | 0.324 |
| DMOP3 | 0.751 | 0.472 | 33.648 | 37.936 | 0.656 | 0.121 |
| HE2 | 0.365 | 0.125 | 0.187 | 0.125 | 0.266 | 0.123 |
| HE7 | 0.309 | 0.051 | 0.285 | 0.042 | 0.311 | 0.049 |
| HE9 | 0.288 | 0.238 | 0.273 | 0.215 | 0.263 | 0.245 |

For each problem with each parameter, Tr-RMMEDA and RMMEDA get initial populations after each change. MIGD is the average of IGDs of initial populations with one run under 20 changes. The better values obtained by the algorithm are shown with dark background.

### 3.1.2 Avoiding Transfer on Problems with Fixed PS And Small Environmental Changes

Generally, it is unlikely that the optimal solutions transferred from a problem to anther problem would still be optimal. Even if the existing Tr-DMOEA is improved, it is unlikely that transferred solutions by the improved Tr-DMOEA would be optimal. As the copied solutions from the previous environment are very good solutions for situations with small changes or problems with fixed POS, it would be very difficult for transferred solutions to beat the copied ones. Therefore, the most intuitive idea to prevent negative results obtained by transfer optimization in such cases would be to prevent using transferred solutions.

It has been computationally shown that transfer optimization fails on problems with small changes and fixed POS in DMO. In order to improve the performance of transfer optimization in DMO, the most intuitive idea is to avoid transfer optimization when it fails. In addition, when transfer optimization works well, it should be definitely used. However, it is impossible for algorithms to foresee which problem has fixed POS and when the environmental changes are small. To overcome this problem, after each change, transferred solutions and copied ones from the previous environment are firstly combined together. After that, nondominated sort and crowding distance in NSGA-II [4] are used to rank the combined solutions on the

new environment. Lastly, solutions with the population size are selected from the combined population as the initial population.

### 3.1.3 Experimental Studies of the Proposed Strategy

This section analyzes the effectiveness of the strategy proposed in section 3.1.2 through experiments. Firstly, all experiment setups are the same as those in Section 3.1.1. The only difference is that we compare the proposed strategy against the original transfer approach, to check whether it can improve the MIGD values of the initial populations after the changes.

Table 3.3: MIGD of combined solutions and transferred solutions.

| Prob. | C4 | | C6 | | C8 | |
|---|---|---|---|---|---|---|
| Methods | Tr. | Comb. | Tr. | Comb. | Tr. | Comb. |
| FDA4 | 0.082 | 0.070 | 0.098 | 0.057 | 0.081 | 0.070 |
| FDA5 | 0.389 | 0.185 | 1.294 | 0.226 | 0.325 | 0.181 |
| $FDA5_{iso}$ | 0.280 | 0.265 | 0.708 | 0.226 | 0.294 | 0.260 |
| $FDA5_{dec}$ | 0.645 | 0.270 | 3.006 | 0.295 | 0.583 | 0.353 |
| DIMP2 | 11.630 | 3.514 | 11.379 | 3.793 | 12.466 | 3.807 |
| DMOP2 | 1.531 | 0.004 | 35.126 | 18.035 | 1.165 | 0.004 |
| $DMOP2_{iso}$ | 0.002 | 0.004 | 0.082 | 0.112 | 0.002 | 0.004 |
| $DMOP2_{dec}$ | 1.044 | 0.019 | 5.182 | 0.191 | 1.028 | 0.017 |
| DMOP3 | 0.751 | 0.003 | 33.648 | 18.022 | 0.656 | 0.003 |
| HE2 | 0.365 | 0.069 | 0.187 | 0.056 | 0.266 | 0.061 |
| HE7 | 0.309 | 0.040 | 0.285 | 0.036 | 0.311 | 0.040 |
| HE9 | 0.288 | 0.226 | 0.273 | 0.199 | 0.263 | 0.237 |

For each problem with each parameter, combined solutions are the initial population introduced in Section 3.1.2, while transferred solutions are the initial population obtained by Tr-RMMEDA in each change. MIGD is the average of IGDs of these initial populations with one run under 20 changes. The better values that the solution set have are shown with dark background.

The specific comparison results of transferred to combined solutions are shown in Table 3.3. The better value that the solution set has is shown with dark background. Combined solutions are termed as *Comb*. It is clear that combined solutions are better than transferred ones (*Tr.*) on almost all test problems under different environments. In addition, the Wilcoxon Rank Sum test with the significance level 0.05 is carried out to indicate significance between results obtained by transferred and combined solutions. MIGD value of *Tr.* and *Comb.* for each problem with one parameter is regarded as one observation for the test [90]. The test obtained a p-value of 1.2532e-4. This shows that the combined solutions are significantly better than transferred solutions in the first generation after change with a level of significance of 0.05.

## 3.2 How Does Transfer Work in Dynamic Multi-objective Optimization?

In this section, we give the answer of how transfer works in DMO and highlight the weakness of using the Gaussian kernel in Tr-DMOEA.

### 3.2.1 Mathematical Proof that Gaussian Kernels Are Not Ideal

In this section, we briefly introduce the foundations of Tr-DMOEA first. Then the weakness of using the Gaussian kernel will be highlighted. The detailed process of Tr-DMOEA can be found in [25].

In Domain Adaption Learning (DAL) [91], a transfer learning method, it is assumed that a transformation should be found to a latent space where the difference between the distributions of source and target domain is minimized. Once this transformation is found, it can act as a bridge to connect the source domain and the target domain. Then solutions that have been optimized in one domain can be transferred to be good solutions in another domain through this bridge.

The distance between the distributions of the source and target domain can be calculated through the Maximum Mean Discrepancy (MMD) [92], which evaluates the distance between two distributions in the Reproducing Kernel Hilbert Space.

The estimated MMD in RKHS can be calculated as:

$$MMD(\mathcal{F}, p, q) := \left\| \frac{1}{m} \sum_{i=1}^{m} \phi(Fs_i) - \frac{1}{n} \sum_{i=1}^{n} \phi(Ft_i) \right\|_{\mathcal{H}}^2. \tag{3.1}$$

In Tr-DMOEA, the distribution of the source and target domains in consideration is the distribution of the objective vectors of source and target solutions. Therefore, in Tr-DMOEA, $FS$ and $FT$ are the objective vectors of randomly generated solutions in the source problem before a change and target problem after a change, respectively. The function $\phi$ is defined as $\phi(F) = W^T \kappa(F)$, where $W$ is a transformation matrix which maps the objective vector into the latent space, and $\kappa(F)$ is defined as follows, where $\kappa(\cdot, \cdot)$ is a kernel function [83]:

$$\kappa(F) = [\kappa(Fs_1, F), ..., \kappa(Fs_m, F), \kappa(Ft_1, F), ..., \kappa(Ft_n, F)]^T.$$

Once $W$ is found, Tr-DMOEA will initialize the population in the target envi-

ronment with solutions whose objective vectors are close to that of good solutions from the source environment in the latent space. For that, it needs to find solutions $t_k$ whose objective vector is close to that of a solution $s_l$ from $POF$ of the source problem, i.e., whose $||\phi(Fs_l) - \phi(Ft_k)||$ is minimal. This distance can be expanded as follows:

$$\begin{aligned} &||\phi(Fs_l) - \phi(Ft_k)|| = ||W^T\kappa(Fs_l) - W^T\kappa(Ft_k)|| \\ =&||W^T[\kappa(Fs_l) - \kappa(Ft_k)]|| = \sum_{j=1}^{d}\Big(\phi_j(Fs_l) - \phi_j(Ft_k)\Big)^2 \\ =&\sum_{j=1}^{d}\left\{\sum_{i=1}^{m+n} W_{ji} \times \Big(\kappa(F_i, Fs_l) - \kappa(F_i, Ft_k)\Big)\right\}^2. \end{aligned} \tag{3.2}$$

in which $d$ is the dimension of the latent space; $F = Fs$ when $i \in [1, m]$ and $F = Ft$ when $i \in [m+1, m+n]$.

From the above we can see that the more similar $\kappa(F_i, Fs_l)$ and $\kappa(F_i, Ft_k)$ are, the smaller the distance in the latent space. In existing Tr-DMOEA, the used kernel function is the Gaussian kernel:

$$\kappa(F, Ft_k) = e^{-(F-Ft_k)^T(F-Ft_k)}. \tag{3.3}$$

Here, we consider bi-objective problems: $Fs_l = (Fs_l^1, Fs_l^2)$; $Ft_k = (Ft_k^1, Ft_k^2)$; $F = (f^1, f^2)$; therefore:

$$\begin{aligned} &\kappa(F_i, Fs_l) - \kappa(F_i, Ft_k) \\ =&e^{-(Fs_l^1 - F_i^1)^2 - (Fs_l^2 - F_i^2)^2} - e^{-(Ft_k^1 - F_i^1)^2 - (Ft_k^2 - F_i^2)^2}. \end{aligned} \tag{3.4}$$

Then, the difference between $\kappa(F_i, Fs_l)$ and $\kappa(F_i, Ft_k)$ is the difference between their exponent:

$$\Big|\Big((Fs_l^1 - F_i^1)^2 + (Fs_l^2 - F_i^2)^2\Big) - \Big((Ft_k^1 - F_i^1)^2 + (Ft_k^2 - F_i^2)^2\Big)\Big| \tag{3.5}$$

$$= \Big|\Big((Fs_l^1 - F_i^1)^2 - (Ft_k^1 - F_i^1)^2\Big) + \Big((Fs_l^2 - F_i^2)^2 - (Ft_k^2 - F_i^2)^2\Big)\Big|. \tag{3.6}$$

To make eq. (5) minimal, both terms of eq (6) should be minimal. To simplify the analysis, only the first term is considered here. When the first term is equal to 0, it can be re-written as follows:

$$\begin{aligned} &\Big|Fs_l^1 - F_i^1\Big| = \Big|Ft_k^1 - F_i^1\Big| \\ \Rightarrow Fs_l^1 - F_i^1 =&Ft_k^1 - F_i^1 \quad \textbf{OR} \quad Fs_l^1 - F_i^1 = F_i^1 - Ft_k^1. \end{aligned} \tag{3.7}$$

Therefore, solutions for $Ft_k^1$ are ($Ft_k^2$ is the similar as $Ft_k^1$):

$$Ft_k^1 = Fs_l^1 \qquad \textbf{OR} \qquad Ft_k^1 = 2F_i^1 - Fs_l^1. \tag{3.8}$$

Therefore, objective values of found solutions are close to either those of the solution $Fs_l$ in $POF_s$ or twice over those of randomly generated solutions ($F_i$). The original intention of Tr-DMOEA was to find a solution in the target environment whose fitness $Ft_k^1$ is similar to that of a $POF_s$ solution in the latent space. However, there is no reason to believe that a solution in the target domain that is similar to a random solution from the source domain in the latent space is going to be a good solution.

In this case, the Gaussian kernel function is not the ideal in present version of Tr-DMOEA.

### 3.2.2 Replacing Gaussian Kernel with Linear Kernel

According to section 3.2.1, the selection of Gaussian kernel is not ideal. After reviewing present common kernel functions, we find that the linear kernel functions [93] [94] [95] overcome the problem presented by the Gaussian kernel function explained in section 3.2.1. In the following, the details why the linear kernel overcomes this problem are explained.

Here, the simplest linear kernel is used:

$$\kappa(F, Ft_k) = F^T Ft_k. \tag{3.9}$$

Therefore,

$$\kappa(F_i, Fs_l) - \kappa(F_i, Ft_k) = F_i^1(Fs_l^1 - Ft_k^1) + F_i^2(Fs_l^2 - Ft_k^2). \tag{3.10}$$

It can be seen from eq (5) when the distance is minimal, $\kappa(F_i, Fs_l) - \kappa(F_i, Ft_k)$ would be close to 0. In this case, $Fs_l^1 = Ft_k^1$ and $Fs_l^2 = Ft_k^2$. Therefore, searched solution $Ft_k = (Ft_k^1, Ft_k^2)$ will be close to the solution $Fs_l = (Fs_l^1, Fs_l^2)$ in last POF, being a potentially good solution to initialize the population in the target domain.

### 3.2.3 Experimental Evaluation of Different Kernels

In section 3.2.2, it has been mathematically proved that the linear kernel function overcomes the problem of the Gaussian kernel. In order to verify the effectiveness of it from the perspective of experiment, specific experiment will be conducted.

In this experiment, two algorithms will be compared. One is Tr-DMOEA with the Gaussian kernel function. Another is Tr-DMOEA with the linear kernel function. For each algorithm, all the experiment setups are the same as those in Section 3.1.1, including used benchmark problems, algorithm parameter setting, frequency and severity of changes, performance metric and so on. The comparison results of the Gaussian and linear kernel based Tr-DMOEA are shown in Table 3.4. In the Table, *Gauss.* and *Lin.* represent the Gaussian and linear kernel based Tr-DMOEA, respectively.

Table 3.4: MIGD values for Gaussian kernel-based and linear kernel-based Tr-RMMEDA.

| Prob. | C4 | | C6 | | C8 | |
|---|---|---|---|---|---|---|
| Methods | Lin. | Gauss. | Lin. | Gauss. | Lin. | Gauss. |
| FDA4 | 0.052 | 0.082 | 0.048 | 0.098 | 0.052 | 0.081 · |
| FDA5 | 0.238 | 0.390 | 0.825 | 1.295 | 0.163 | 0.325 |
| $FDA5_{iso}$ | 0.204 | 0.280 | 0.628 | 0.708 | 0.185 | 0.294 |
| $FDA5_{dec}$ | 0.464 | 0.645 | 0.641 | 3.006 | 0.265 | 0.583 |
| DIMP2 | 10.909 | 11.630 | 10.527 | 11.379 | 12.076 | 12.466 |
| DMOP2 | 0.006 | 1.5313 | 33.085 | 35.126 | 0.003 | 1.166 |
| $DMOP2_{iso}$ | 0.002 | 0.002 | 0.087 | 0.082 | 0.002 | 0.002 |
| $DMOP2_{dec}$ | 0.053 | 1.044 | 9.107 | 5.182 | 0.070 | 1.028 |
| DMOP3 | 0.003 | 0.751 | 35.076 | 33.648 | 0.003 | 0.656 |
| HE2 | 0.115 | 0.365 | 0.096 | 0.187 | 0.071 | 0.266 |
| HE7 | 0.301 | 0.308 | 0.265 | 0.285 | 0.300 | 0.311 |
| HE9 | 0.303 | 0.288 | 0.297 | 0.273 | 0.294 | 0.263 |

For each problem with each parameter, Gaussian kernel-based and linear kernel-based Tr-RMMEDA get initial populations after each change. MIGD is the average of IGDs of these initial populations with one run under 20 changes. The better values that the method gets are shown with dark background.

It can be seen from the table that when change is small (C8) and medium (C4), Tr-DMOEA with linear kernel function is better than that with Gaussian kernel function except HE9 problem. Plus, when changes are large (C6), there are only 4 out of 12 cases that linear kernel-based Tr-DMOEA is worse than Gaussian kernel-based Tr-DMOEA. As a whole, the linear kernel greatly improves the performance of solution quality after change in the first generation, compared with the Gaussian kernel one. In addition, the Wilcoxon Rank Sum test with the significance level 0.05 is carried out to indicate significance between results obtained by transferred and combined solutions. MIGD value of *Lin.* and *Gauss.* for each problem with one parameter is regarded as one observation data for the test. The result with $h = 1$ and $p = 0.0132$ shows that transferred solutions with the linear kernel are significantly better than those with the Gaussian one in the first generation after change.

## 3.3 Improved Transfer in DMO

It has been mathematically and experimentally shown that when and how to transfer is vitally important in DOM. Transfer optimization fails on problems with stationary POS and when changes are small. The kernel function also matters in Tr-DMOEA, which contributes the idea of replacing the Gaussian kernel with a linear one. Therefore, in order to make improvements on present Tr-DMOEA, this section proposes novel method for Tr-DMOEA.

### 3.3.1 Proposed Method to Generate An Initial Population

**Algorithm 3.1: Responding strategy based on improved transfer**

**Input:** Two consecutive DMOPs $F_{t-1}(\cdot)$ and $F_t(\cdot)$; $POS$ and $POF$ at the environment $t-1$; linear kernel function $\kappa$

**Output:** The responding initial population $\mathbf{P}_{ini}$.

1 Initialization;
2 Randomly sample two solution sets in the search space of problems $F_{t-1}(\cdot)$ and $F_t(\cdot)$, as $X_{t-1}$ and $Y_t$;
3 Evaluate $X_{t-1}$ and $Y_t$ on their own objective functions to get their objective vectors $\mathbf{F}_{t-1}$ and $\mathbf{F}_t$;
4 Use $\mathbf{F}_{t-1}$ and $\mathbf{F}_t$ as well as *the linear kernel function* to get the transformation matrix W [25] ;
5 Determine target domain solutions $X_{Tr}$ whose fitness is similar to that of the $\mathrm{POF}_t$ solutions in the latent space. *the linear kernel function* is used here to map solutions to the latent space;
6 Calculate the objective values of $X_{Tr}$ and $POS_{t-1}$ on problem $F_t(\cdot)$;
7 Sort on the combined populations $X_{Tr}$ and $POS_{t-1}$ through nondominated sort and crowding distance;
8 Select the top $N$ solutions from the combined population as $P_{ini}$, where $N$ is the population size;
9 **Return** $\mathbf{P}_{ini}$.

The main idea in a DMOEA is to generate an initial population such that the population can quickly reach the new optimum after a change. Therefore, in Tr-DMOEA, an initial population is produced through transferring good solutions from the previous environment.

In this section, we present a method that makes use of the strategies proposed in Sections 3.1.2 and 3.2.2 to improve Tr-DMOEA's solutions in the initial population after the changes. Firstly, in the process of transfer, the Gaussian kernel will be replaced with a linear one, as described in Section 3.2.2. Secondly, considering that it is difficult to judge when changes are small and which problem has fixed POS for

Table 3.5: MIGD values for the original and improved Tr-RMMEDA

| Prob. | C4 | | C6 | | C8 | |
|---|---|---|---|---|---|---|
| Methods | Tr. | ImTr. | Tr. | ImTr. | Tr. | ImTr. |
| FDA4 | 0.082 | 0.070 | 0.098 | 0.058 | 0.081 | 0.071 |
| FDA5 | 0.389 | 0.285 | 1.294 | 0.822 | 0.325 | 0.249 |
| $FDA5_{iso}$ | 0.280 | 0.344 | 0.708 | 0.633 | 0.294 | 0.336 |
| $FDA5_{dec}$ | 0.645 | 0.327 | 3.006 | 0.507 | 0.583 | 0.254 |
| DIMP2 | 11.630 | 12.093 | 11.379 | 12.163 | 12.466 | 11.860 |
| DMOP2 | 1.531 | 0.006 | 35.126 | 32.984 | 1.165 | 0.003 |
| $DMOP2_{iso}$ | 0.002 | 0.004 | 0.082 | 0.102 | 0.002 | 0.004 |
| $DMOP2_{dec}$ | 1.044 | 0.052 | 5.182 | 10.026 | 1.028 | 0.024 |
| DMOP3 | 0.751 | 0.004 | 33.648 | 33.219 | 0.656 | 0.004 |
| HE2 | 0.365 | 0.115 | 0.187 | 0.136 | 0.266 | 0.112 |
| HE7 | 0.309 | 0.309 | 0.285 | 0.292 | 0.311 | 0.319 |
| HE9 | 0.288 | 0.255 | 0.273 | 0.275 | 0.263 | 0.254 |

For each problem with each parameter, the original and improved Tr-RMMEDA get initial populations after each change. MIGD is the average of IGDs of these initial populations with one run under 20 changes. The better values that the method gets are shown with dark background.

a DMOEA, transferred solutions and solutions copied from the previous environment are combined together, as described in Section 3.1.2. The initial population is selected from the combined populations and then regarded as the initial population for the problem in next environment. The detailed procedures of improved Tr-DMOEA are shown in Algorithm 3.1. The differences with regard to the original Tr-DMOEA are highlighted in the pseudocode. This algorithm can be embedded into any population based evolutionary algorithms.

### 3.3.2 Evaluation through Computational Studies

In order to evaluate the effectiveness of the proposed improved Tr-DMOEA, computational studies are conducted in this section. Firstly, we compare the performance of initial population produced by the original transfer and improved one in DMO following a similar procedure to that used in Sections 3.1.3 and 3.2.3. Then, the performance of populations obtained by several compared methods after optimization is stated. The reason why we are now also comparing the results after optimization is to verify whether the improved Tr-DMOEA helps to solve DMOPs, compared with other state-of-the-arts.

#### 3.3.2.1 Quality Comparison in the First Generation

Similar to previous experiments, comparison results of solutions after the change produced by the original and improved Tr-RMMEDA are shown in Table 3.5. Ex-

perimental setup is the same as those in Section 3.1.1. It can be observed from the table that results of the improved Tr-RMMEDA are better than the original one on most of investigated problems. In addition, the Wilcoxon Rank Sum test with the significance level 0.05 is carried out to indicate significance between results obtained by transferred and combined solutions. MIGD value of $Tr.$ and $ImTr.$ for each problem with one parameter is regarded as one observation data for the test. The result with $h = 1$ and $p = 0.0269$ shows that the transferred solutions through the original Tr-DMOEA are significantly better than those through the improved Tr-DMOEA in the first generation after change.

#### 3.3.2.2 Solutions Quality Comparison after Optimization

Here, the experiment is conducted to verify the effect of the proposed method at the final generation after the changes. The experiment setup is the similar to those in Section 3.1.1 except for the following.

1. Three investigated population-based algorithms, the fast and elitist multi-objective genetic algorithm (NSGA-II) [4], multi-objective particle swarm optimization algorithm (MOPSO) [96] and regularity model-based multi-objective estimation of distribution algorithm (RMMEDA) [30], are regarded as the optimization algorithms. The DMOEAs with the original transfer are termed as Tr-DMOEAs. Similarly, DMOEAs with proposed method are written as ImTr-DMOEAs.

2. Compared algorithms: For each ImTr-DMOEA, there are three compared DMOEAs: the original Tr-DMOEA, the algorithm with copied solutions from last change (DMOEA) and that with totally random population (Rnd-DMOEA). The reason for using three different optimization algorithms is to test the performance sensitivity of transfer optimization method in three different type of algorithms.

3. Parameters of different algorithms are set as the same in their original papers [25] [16] [4] [96]. For the parameters regarding the frequency of changes and severity of changes of these problems, **all parameters** shown in Table 3.1 are used. It aims to show the performance of all compared methods under different frequency and severity of changes.

MIGD results of four compared DMOEAs with three different optimization algorithms on all test problems are shown in Tables 3.6-3.8.

Table 3.6: MIGD values of four DMOEAs with the optimization algorithm RMMEDA on all benchmark problems.

| Prob. | C1 | | | | C2 | | | | C3 | | | | C4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | RND | COPY | Trans. | ImTr. | RND | COPY | Tran. | ImTr. | RND | COPY | Trans. | ImTr. | RND | COPY | Trans. | ImTr. |
| FDA4 | 0.434 | 0.105 | 0.052 | 0.059 | 0.300 | 0.080 | 0.052 | 0.062 | 0.103 | 0.065 | 0.056 | 0.065 | 0.068 | 0.060 | 0.057 | 0.070 |
| FDA5 | 0.905 | 0.263 | 0.705 | 0.191 | 0.725 | 0.164 | 0.567 | 0.188 | 0.369 | 0.099 | 0.273 | 0.175 | 0.164 | 0.087 | 0.137 | 0.165 |
| $FDA5_{iso}$ | 0.315 | 0.066 | 0.267 | 0.227 | 0.177 | 0.066 | 0.170 | 0.192 | 0.093 | 0.068 | 0.080 | 0.276 | 0.066 | 0.066 | 0.066 | 0.263 |
| $FDA5_{dec}$ | 2.010 | 0.968 | 1.186 | 0.313 | 1.656 | 0.314 | 0.865 | 0.267 | 0.963 | 0.201 | 0.780 | 0.234 | 0.649 | 0.122 | 0.494 | 0.232 |
| DIMP2 | 9.063 | 16.351 | 8.471 | 8.746 | 7.824 | 12.310 | 6.968 | 7.269 | 5.153 | 7.916 | 5.097 | 5.087 | 4.115 | 5.381 | 3.980 | 4.129 |
| DMOP2 | 1.961 | 20.295 | 0.465 | 0.003 | 0.956 | 5.994 | 0.220 | 0.003 | 0.091 | 0.143 | 0.040 | 0.003 | 0.004 | 0.009 | 0.003 | 0.003 |
| $DMOP2_{iso}$ | 0.002 | 0.002 | 0.002 | 0.004 | 0.002 | 0.002 | 0.002 | 0.004 | 0.002 | 0.002 | 0.002 | 0.004 | 0.002 | 0.002 | 0.002 | 0.004 |
| $DMOP2_{dec}$ | 2.266 | 2.148 | 0.538 | 0.079 | 1.084 | 2.390 | 0.420 | 0.045 | 0.374 | 0.154 | 0.147 | 0.051 | 0.105 | 0.069 | 0.061 | 0.038 |
| DMOP3 | 1.453 | 13.041 | 0.243 | 0.003 | 0.603 | 3.677 | 0.095 | 0.003 | 0.056 | 0.141 | 0.015 | 0.003 | 0.003 | 0.008 | 0.004 | 0.003 |
| HE2 | 2.779 | 0.451 | 0.527 | 0.140 | 2.397 | 0.268 | 0.476 | 0.110 | 1.581 | 0.116 | 0.293 | 0.068 | 0.885 | 0.085 | 0.146 | 0.063 |
| HE7 | 0.172 | 0.049 | 0.166 | 0.045 | 0.129 | 0.048 | 0.125 | 0.041 | 0.076 | 0.050 | 0.077 | 0.038 | 0.053 | 0.051 | 0.052 | 0.039 |
| HE9 | 0.342 | 0.244 | 0.294 | 0.241 | 0.313 | 0.243 | 0.273 | 0.237 | 0.283 | 0.250 | 0.254 | 0.233 | 0.264 | 0.239 | 0.244 | 0.229 |
| Prob. | C5 | | | | C6 | | | | C7 | | | | C8 | | | |
| Methods | RND | COPY | Trans. | ImTr. | RND | COPY | Tran. | ImTr. | RND | COPY | Trans. | ImTr. | RND | COPY | Trans. | ImTr. |
| FDA4 | 0.547 | 0.547 | 0.051 | 0.053 | 0.069 | 0.068 | 0.051 | 0.057 | 0.277 | 0.069 | 0.052 | 0.063 | 0.066 | 0.058 | 0.057 | 0.071 |
| FDA5 | 1.061 | 2.166 | 0.558 | 0.345 | 0.271 | 0.457 | 0.195 | 0.179 | 0.675 | 0.123 | 0.386 | 0.183 | 0.183 | 0.084 | 0.114 | 0.164 |
| $FDA5_{iso}$ | 0.243 | 0.212 | 0.176 | 0.186 | 0.061 | 0.061 | 0.062 | 0.180 | 0.171 | 0.066 | 0.164 | 0.192 | 0.065 | 0.066 | 0.065 | 0.258 |
| $FDA5_{dec}$ | 1.127 | 2.049 | 0.948 | 0.426 | 0.328 | 0.472 | 0.270 | 0.206 | 1.566 | 0.219 | 0.945 | 0.127 | 0.609 | 0.098 | 0.537 | 0.144 |
| DIMP2 | 7.845 | 13.596 | 7.338 | 7.230 | 4.341 | 7.575 | 4.202 | 4.289 | 8.176 | 15.904 | 7.218 | 7.281 | 4.126 | 7.204 | 4.097 | 3.929 |
| DMOP2 | 26.944 | 37.821 | 23.744 | 18.039 | 18.308 | 28.273 | 18.899 | 18.008 | 0.874 | 1.528 | 0.295 | 0.003 | 0.004 | 0.005 | 0.003 | 0.003 |
| $DMOP2_{iso}$ | 0.110 | 0.110 | 0.110 | 0.090 | 0.110 | 0.110 | 0.110 | 0.112 | 0.002 | 0.002 | 0.002 | 0.004 | 0.002 | 0.002 | 0.002 | 0.004 |
| $DMOP2_{dec}$ | 2.742 | 59.767 | 1.116 | 0.797 | 0.232 | 36.583 | 0.238 | 0.224 | 1.325 | 0.300 | 0.402 | 0.015 | 0.143 | 0.020 | 0.050 | 0.012 |
| DMOP3 | 25.957 | 35.728 | 22.625 | 18.016 | 18.280 | 27.185 | 18.654 | 18.011 | 0.459 | 0.269 | 0.120 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 |
| HE2 | 1.807 | 0.258 | 0.281 | 0.071 | 0.726 | 0.084 | 0.085 | 0.061 | 2.342 | 0.279 | 0.432 | 0.094 | 0.907 | 0.085 | 0.123 | 0.061 |
| HE7 | 0.116 | 0.042 | 0.114 | 0.035 | 0.047 | 0.041 | 0.048 | 0.035 | 0.131 | 0.052 | 0.130 | 0.042 | 0.053 | 0.046 | 0.051 | 0.040 |
| HE9 | 0.287 | 0.214 | 0.249 | 0.203 | 0.234 | 0.221 | 0.219 | 0.201 | 0.316 | 0.247 | 0.275 | 0.234 | 0.264 | 0.248 | 0.243 | 0.228 |

For each problem with each parameter, all algorithms get optimized populations by RMMEDA in the last generation of each change. MIGD is the average of IGDs of these populations with one run under 20 changes. The best values that the algorithm obtains are shown with dark background.

Table 3.7: MIGD values of four DMOEAs with the optimization algorithm NSGA-II on all benchmark problems.

| Prob. | C1 | | | | C2 | | | | C3 | | | | C4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | RND | COPY | Trans. | ImTr. | RND | COPY | Tran. | ImTr. | RND | COPY | Trans. | ImTr. | RND | COPY | Trans. | ImTr. |
| FDA4 | 0.676 | 2.329 | 0.314 | 0.365 | 0.657 | 2.376 | 0.237 | 0.391 | 0.429 | 2.134 | 0.152 | 0.293 | 0.280 | 0.777 | 0.118 | 0.263 |
| FDA5 | 1.039 | 1.888 | 0.362 | 0.812 | 0.947 | 2.147 | 0.318 | 0.733 | 0.693 | 1.812 | 0.321 | 0.519 | 0.419 | 1.106 | 0.291 | 0.327 |
| FDA5$_{iso}$ | 0.279 | 0.197 | 0.790 | 0.290 | 0.176 | 0.133 | 0.416 | 0.141 | 0.099 | 0.113 | 0.229 | 0.095 | 0.098 | 0.100 | 0.135 | 0.098 |
| FDA5$_{dec}$ | 1.671 | 2.536 | 0.501 | 1.075 | 1.428 | 2.415 | 0.409 | 0.966 | 0.877 | 1.250 | 0.380 | 0.696 | 0.722 | 0.636 | 0.300 | 0.409 |
| DIMP2 | 9.683 | 12.735 | 5.990 | 9.502 | 7.685 | 10.792 | 4.233 | 7.111 | 4.780 | 8.172 | 3.312 | 4.394 | 3.976 | 5.292 | 2.404 | 3.724 |
| DMOP2 | 2.492 | 29.246 | 0.744 | 1.618 | 2.214 | 24.647 | 0.560 | 1.730 | 0.476 | 5.311 | 0.147 | 0.572 | 0.164 | 0.099 | 0.047 | 0.130 |
| DMOP2$_{iso}$ | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 |
| DMOP2$_{dec}$ | 2.806 | 5.644 | 0.825 | 1.043 | 2.071 | 5.083 | 0.640 | 1.263 | 0.858 | 1.053 | 0.367 | 0.664 | 0.486 | 0.615 | 0.244 | 0.384 |
| DMOP3 | 2.267 | 24.577 | 0.619 | 1.102 | 1.769 | 26.646 | 0.338 | 1.030 | 0.463 | 3.987 | 0.128 | 0.510 | 0.083 | 0.153 | 0.056 | 0.100 |
| HE2 | 2.821 | 1.834 | 0.312 | 2.409 | 2.557 | 1.296 | 0.264 | 1.982 | 1.155 | 0.784 | 0.211 | 1.247 | 0.464 | 0.367 | 0.172 | 0.401 |
| HE7 | 0.228 | 0.086 | 0.073 | 0.163 | 0.194 | 0.075 | 0.074 | 0.104 | 0.138 | 0.088 | 0.096 | 0.093 | 0.090 | 0.072 | 0.077 | 0.070 |
| HE9 | 0.363 | 0.296 | 0.398 | 0.344 | 0.351 | 0.298 | 0.371 | 0.333 | 0.328 | 0.295 | 0.324 | 0.316 | 0.303 | 0.287 | 0.302 | 0.311 |

| Prob. | C5 | | | | C6 | | | | C7 | | | | C8 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | RND | COPY | Trans. | ImTr. | RND | COPY | Tran. | ImTr. | RND | COPY | Trans. | ImTr. | RND | COPY | Trans. | ImTr. |
| FDA4 | 1.027 | 0.559 | 0.243 | 0.894 | 0.617 | 0.593 | 0.101 | 0.478 | 0.644 | 1.966 | 0.258 | 0.362 | 0.283 | 0.592 | 0.102 | 0.227 |
| FDA5 | 1.185 | 0.542 | 0.419 | 0.711 | 0.709 | 0.490 | 0.337 | 0.368 | 0.896 | 1.176 | 0.227 | 0.806 | 0.325 | 0.798 | 0.243 | 0.421 |
| FDA5$_{iso}$ | 0.258 | 0.173 | 0.251 | 0.284 | 0.125 | 0.140 | 0.173 | 0.166 | 0.205 | 0.173 | 0.217 | 0.138 | 0.096 | 0.100 | 0.248 | 0.096 |
| FDA5$_{dec}$ | 1.399 | 1.730 | 1.021 | 0.485 | 1.181 | 0.921 | 0.774 | 0.369 | 1.596 | 2.046 | 0.390 | 0.575 | 0.426 | 0.773 | 0.323 | 0.272 |
| DIMP2 | 7.634 | 6.109 | 4.049 | 7.088 | 3.737 | 4.046 | 2.297 | 4.075 | 8.105 | 9.614 | 4.430 | 7.385 | 4.241 | 6.098 | 2.645 | 3.254 |
| DMOP2 | 6.296 | 3.114 | 3.465 | 7.669 | 0.702 | 0.237 | 0.507 | 0.426 | 2.168 | 21.833 | 0.389 | 1.312 | 0.108 | 0.078 | 0.043 | 0.099 |
| DMOP2$_{iso}$ | 0.119 | 0.122 | 0.137 | 0.097 | 0.119 | 0.117 | 0.140 | 0.129 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 |
| DMOP2$_{dec}$ | 8.249 | 10.087 | 2.993 | 8.118 | 1.368 | 13.831 | 0.898 | 1.463 | 1.970 | 1.172 | 0.459 | 1.205 | 0.246 | 0.280 | 0.178 | 0.310 |
| DMOP3 | 10.584 | 0.293 | 1.978 | 5.253 | 0.308 | 0.298 | 3.350 | 3.979 | 1.360 | 25.525 | 0.270 | 1.087 | 0.127 | 0.156 | 0.029 | 0.064 |
| HE2 | 1.735 | 0.340 | 0.094 | 0.362 | 1.087 | 0.249 | 0.073 | 0.184 | 2.655 | 0.461 | 0.247 | 1.572 | 0.926 | 0.295 | 0.172 | 0.355 |
| HE7 | 0.174 | 0.075 | 0.068 | 0.090 | 0.079 | 0.051 | 0.042 | 0.041 | 0.197 | 0.079 | 0.099 | 0.128 | 0.101 | 0.055 | 0.056 | 0.065 |
| HE9 | 0.319 | 0.243 | 0.296 | 0.280 | 0.271 | 0.242 | 0.259 | 0.251 | 0.350 | 0.270 | 0.361 | 0.341 | 0.302 | 0.293 | 0.308 | 0.289 |

For each problem with each parameter, all algorithms get optimized populations by NSGA-II in the last generation of each change. MIGD is the average of IGDs of these populations with one run under 20 changes. The best values that the algorithm obtains are shown with dark background.

Table 3.8: MIGD values of four DMOEAs with the optimization algorithm MOPSO on all benchmark problems.

| Prob. | C1 | | | | C2 | | | | C3 | | | | C4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | RND | COPY | Trans. | ImTr. | RND | COPY | Tran. | ImTr. | RND | COPY | Trans. | ImTr. | RND | COPY | Trans. | ImTr. |
| FDA4 | 0.437 | 0.252 | 0.054 | 0.055 | 0.288 | 0.168 | 0.057 | 0.060 | 0.158 | 0.096 | 0.062 | 0.064 | 0.109 | 0.074 | 0.066 | 0.066 |
| FDA5 | 1.033 | 0.536 | 0.436 | 0.177 | 0.770 | 0.418 | 0.302 | 0.156 | 0.535 | 0.180 | 0.198 | 0.165 | 0.379 | 0.151 | 0.158 | 0.138 |
| FDA5$_{iso}$ | 0.306 | 0.150 | 0.248 | 0.157 | 0.218 | 0.135 | 0.260 | 0.148 | 0.149 | 0.152 | 0.162 | 0.147 | 0.135 | 0.125 | 0.151 | 0.147 |
| FDA5$_{dec}$ | 0.993 | 0.385 | 0.705 | 0.245 | 0.802 | 0.272 | 0.697 | 0.192 | 0.417 | 0.156 | 0.438 | 0.163 | 0.289 | 0.158 | 0.209 | 0.143 |
| DIMP2 | 13.497 | 9.013 | 10.690 | 7.771 | 13.062 | 8.332 | 9.935 | 6.400 | 10.806 | 7.838 | 8.310 | 6.731 | 8.432 | 7.061 | 5.897 | 6.334 |
| DMOP2 | 2.076 | 2.000 | 3.393 | 0.062 | 1.336 | 2.044 | 4.303 | 0.025 | 0.782 | 0.860 | 0.757 | 0.019 | 0.782 | 0.515 | 2.671 | 0.009 |
| DMOP2$_{iso}$ | 0.005 | 0.006 | 0.006 | 0.006 | 0.005 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 |
| DMOP2$_{dec}$ | 3.429 | 1.463 | 4.930 | 0.251 | 3.093 | 1.634 | 3.268 | 0.262 | 1.928 | 0.958 | 3.086 | 0.155 | 1.181 | 0.673 | 2.161 | 0.113 |
| DMOP3 | 1.732 | 0.810 | 0.243 | 0.016 | 0.942 | 0.344 | 0.147 | 0.004 | 0.659 | 0.140 | 0.059 | 0.006 | 0.203 | 0.058 | 0.059 | 0.006 |
| HE2 | 2.500 | 0.124 | 0.285 | 0.059 | 1.866 | 0.076 | 0.219 | 0.057 | 0.812 | 0.060 | 0.148 | 0.058 | 0.287 | 0.058 | 0.112 | 0.058 |
| HE7 | 0.263 | 0.062 | 0.060 | 0.050 | 0.230 | 0.053 | 0.053 | 0.042 | 0.185 | 0.049 | 0.048 | 0.033 | 0.137 | 0.042 | 0.034 | 0.033 |
| HE9 | 0.381 | 0.303 | 0.322 | 0.274 | 0.368 | 0.294 | 0.313 | 0.285 | 0.329 | 0.296 | 0.280 | 0.297 | 0.305 | 0.301 | 0.299 | 0.301 |
| Prob. | C5 | | | | C6 | | | | C7 | | | | C8 | | | |
| Methods | RND | COPY | Trans. | ImTr. | RND | COPY | Tran. | ImTr. | RND | COPY | Trans. | ImTr. | RND | COPY | Trans. | ImTr. |
| FDA4 | 0.255 | 0.904 | 0.053 | 0.053 | 0.054 | 0.052 | 0.053 | 0.052 | 0.276 | 0.122 | 0.059 | 0.063 | 0.111 | 0.069 | 0.067 | 0.067 |
| FDA5 | 0.964 | 2.238 | 0.442 | 1.426 | 0.444 | 0.712 | 0.143 | 0.450 | 0.671 | 0.199 | 0.329 | 0.116 | 0.305 | 0.140 | 0.129 | 0.129 |
| FDA5$_{iso}$ | 0.258 | 0.419 | 0.148 | 0.376 | 0.124 | 0.280 | 0.133 | 0.324 | 0.194 | 0.133 | 0.225 | 0.141 | 0.136 | 0.131 | 0.153 | 0.125 |
| FDA5$_{dec}$ | 1.493 | 1.702 | 1.289 | 0.750 | 1.041 | 1.083 | 0.733 | 0.934 | 0.793 | 0.158 | 0.386 | 0.160 | 0.291 | 0.153 | 0.164 | 0.145 |
| DIMP2 | 12.972 | 9.121 | 7.753 | 9.081 | 8.932 | 6.033 | 7.373 | 7.061 | 12.233 | 6.654 | 10.029 | 6.664 | 7.758 | 4.099 | 7.090 | 3.433 |
| DMOP2 | 24.948 | 38.943 | 33.468 | 18.085 | 19.351 | 39.751 | 19.077 | 20.844 | 1.345 | 0.352 | 0.680 | 0.021 | 0.555 | 0.062 | 1.736 | 0.005 |
| DMOP2$_{iso}$ | 0.113 | 0.114 | 0.114 | 0.092 | 0.113 | 0.114 | 0.113 | 0.113 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 | 0.006 |
| DMOP2$_{dec}$ | 8.044 | 31.433 | 14.416 | 30.323 | 2.640 | 24.972 | 14.606 | 25.805 | 2.775 | 0.745 | 0.888 | 0.204 | 1.243 | 0.101 | 1.592 | 0.139 |
| DMOP3 | 23.435 | 33.862 | 28.885 | 20.397 | 18.399 | 19.220 | 19.124 | 18.014 | 1.024 | 0.122 | 0.122 | 0.013 | 0.307 | 0.029 | 0.023 | 0.007 |
| HE2 | 1.349 | 0.078 | 0.290 | 0.065 | 0.342 | 0.061 | 0.070 | 0.060 | 1.757 | 0.076 | 0.269 | 0.058 | 0.400 | 0.059 | 0.110 | 0.058 |
| HE7 | 0.215 | 0.051 | 0.066 | 0.050 | 0.124 | 0.053 | 0.037 | 0.039 | 0.244 | 0.053 | 0.047 | 0.043 | 0.125 | 0.044 | 0.036 | 0.026 |
| HE9 | 0.326 | 0.270 | 0.256 | 0.258 | 0.271 | 0.269 | 0.233 | 0.268 | 0.365 | 0.302 | 0.309 | 0.295 | 0.300 | 0.302 | 0.287 | 0.295 |

For each problem with each parameter, all algorithms get optimized populations by MOPSO in the last generation of each change. MIGD is the average of IGDs of these populations with one run under 20 changes. The best values that the algorithm obtains are shown with dark background.

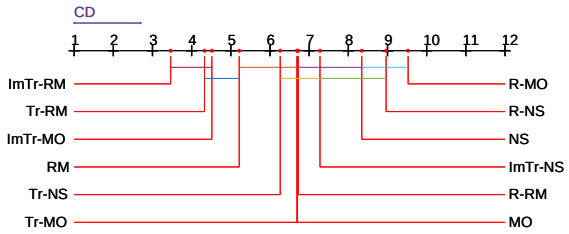



Figure 3.1: Friedman ranking among MIGD of obtained solutions by all DMOEAs from left to right, 4 responding strategies with respect to 3 EAs including RMMEDA, NSGA-II and MOPSO (referred as RM, NS and MO for simplicity). Any pair of approaches whose distance between them is larger than CD are considered to be different.

Additionally, in order to show the significant superiority of the proposed method to other algorithms, Friedman and Nemenyi statistical tests are implemented on all benchmark problems regarding MIGD of 12 algorithms (4 responding strategies with 3 EAs). The MIGD values that all algorithms get on one problem with one parameter setting are regarded as an observation of the test. Therefore, there are 96 (12 problems and 8 parameters) observed data. Friedman detects significant differences in average value with a p-value of 1.3726e-55. The Nemenyi post-tests are shown in Figure 3.1, and are discussed over the next sub-sections. According to the test, average MIGD value of the ImTr-RMMEDA is significantly better than that of the other approaches except Tr-RMMEDA and ImTr-MOSPSO.

#### 3.3.2.3 Impact of EAs on Different DMOEAs

Some DMOEAs with different EAs have differentiate performance, while others have the similar performance, which can be seen from Figure. 3.1. As a whole, MIGD values that RMMEDA obtains are better than those from MOPSO in most cases regarding all problems and different frequency and severity of changes, while NSGA-II gets the worst MIGD results. For example, for problem FDA4 with the parameter setting C2, each DMOEA with RMMEDA has better MIGD than that of NSGA-II, no matter what the MIGD order of different DMOEAs with the same EA. This shows that RMMEDA are more suitable to solve these DMOPs than other two EAs. The above observations are confirmed by the Friedman and Nemenyi tests in Figure 3.1.

When transferred solutions are not converged, solutions with Gaussian kernel-based transfer have better diversity, enabling better results to be achieved after the change than when using linear kernel. That's why the improved Tr-NSGA-II is worse than the original Tr-NSGA-II.

#### 3.3.2.4 Performance of DMOEAs on Different Benchmarks

In general, it is clear from Tables 3.7-3.8 that transfer learning-based DMOEAs are all better than static MOEAs and DMOEAs with random initial population. This shows that transferred solutions can maintain the balance of convergence and diversity to some degree, compared with static MOEAs and random DMOEAs. When comparing the performance of Tr-DMOEAs and ImTr-DMOEAs, ImTr-DMOEAs are basically better than Tr-DMOEAs. Specifically, for HEs problems, ImTr-DMOEAs are all better than Tr-DMOEAs except HE2 and HE7 when the EA is NSGA-II. This shows that transfer learning-based DMOEA with the linear kernel and combined solutions is more capable of solving DMOPs with fixed POS.

Regarding tri-objective optimization problems, ImTr-DMOEAs performs better than Tr-DMOEAs on FDA5$_{iso}$ and FDA5$_{dec}$ while worse than Tr-DMOEAs on FDA4 and FDA5. This implies that the improved Tr-DMOEAs can solve tri-objective problems with complex property such as isolated or deceptive POF. For DIMP2 whose decision variable has its own rate of change, it is difficult for any algorithm to solve, compared with other problems. The original Tr-DMOEA shows its superiority on this problem. In terms of other bi-objective problems, the Tr-DMOEAs with linear kernel function are all superior to those with the Gaussian kernel. This demonstrates that the improved Tr-DMOEA is better capable of solving tri-objective problems except the one with very complicated features like DIMP2, for which all algorithms struggle.

#### 3.3.2.5 Influence of Different Settings of $\tau_t$ and $n_t$ on Tr-DMOEAs

Overall, it is clear from Table 3.7-3.8 that MIGD values obtained by all DMOEAs become better with the increase of changing frequency, when they are under same changing severity. The reason is obvious, which is that the more the iterations within each change, the better the performance. Additionally, no matter what kind of changing sizes are, two kinds of Tr-DMOEAs are all better than the random and static ones. The only difference between the original and the improved Tr-DMOEAs is their performance with different EAs, which has been introduced in section 3.3.2.3. Therefore, it can be concluded that the improved Tr-DMOEA can address most investigated problems no matter what kinds of changing frequency and severity are, as long as solutions before the change have been converged.

## 3.4 Summary

This Chapter studied transfer in DMO, analyzing when and how transfer optimization works in DMO. It has been computationally observed that transfer optimization works poorly on problems with fixed PS and when environmental changes are small. It has also been shown that the Gaussian kernel function in the existing method Tr-DMOEA [25] is inadequate for DMO since it results in half of transferred solutions to be far away from the optimal solutions. Based on these two observations, a new method has been proposed to avoid transfer when it is expected to fail and replace the Gaussian kernel with a linear one. In particular, the linear kernel overcomes the limitation of the Gaussian one and does not result in transferred solutions far away from the true PF. Experimental results have shown that our proposed method is effective in solving DMOPs, compared with other state-of-the-art algorithm Tr-DMOEAs [25].

Even though the quality of transferred solutions and optimized solutions has been improved by studying the research questions of when and how to transfer in DMO, it is unclear how much computational budget the knowledge transfer costs as an extra component embedded into normal process of DMOEAs. It is therefore important to uncover how much computation time transfer costs and whether the efficiency of transfer can be improved as well as whether the solution quality would be affected if the efficiency could be improved, which will be studied in the next Chapter.

# Chapter 4

# Improving the Efficiency of Transfer in Tr-DMOEA for Changing Shape/Position of PSs/PFs [1]

Although quality of transferred and optimized solutions has been enhanced by the proposed improved version of Tr-DMOEAs in Chapter 3, it is unclear how efficient transfer optimization is in DMO as an extra component added to DMOEAs. It is also unclear whether the cost of transfer optimization is so high as to defeat the original purpose of using transfer (i.e., the purpose of quickly finding good solutions for the new environment) and whether the efficiency could be improved while preserving the solution quality.

Therefore, this Chapter aims to study how efficient and effective transfer optimization is in dynamic multi-objective optimization (DMO), answering the **Research question 2 (RQ2)** in Section 1.2: "What is the computational cost of knowledge transfer and could it be so high as to defeat the purpose of using knowledge transfer in DMO to find good solutions as quickly as possible when changes occur? How to improve the efficiency of knowledge transfer in DMO while preserving the quality of solutions found when solving DMOPs with changing PSs and/or PFs position and/or shape?". This research question can be divided into the following three sub-research questions:

1. What is the computational cost of knowledge transfer in DMO?
2. Could it be so high as to defeat the purpose of using knowledge transfer in DMO to find good solutions as quickly as possible when changes occur?
3. How to improve the efficiency of knowledge transfer in DMO while preserving

[1] This chapter is written based on content from the published paper [2] as [*2] in Section 1.4.

the quality of solutions found?

We utilize the state-of-the-art method transfer learning-based dynamic multi-objective evolutionary algorithms (Tr-DMOEAs) [25] as an example of transfer learning-based algorithm to investigate the answers to these three sub-research questions. Specifically, Section 4.1 gives a computation time analysis of transfer optimization, showing that the optimization of the 'inner' problem introduced by transfer optimization to generate transferred solutions (equation (2.13) of Chapter 2) is very time-consuming. Considering the high computational cost of transfer optimization, it is likely that the original purpose of using transfer optimization in DMO might be negated. In other words, the computation time saved in optimization may be defeated by computationally expensive transfer optimization. As a result, there is less gain than expected in the overall computational efficiency. To verify this, experiments have been conducted in Section 4.2, regarding using computational cost of transfer optimization to optimize randomly generated solutions. The results have demonstrated that the convergence and diversity of final solutions generated from the random solutions are significantly better than those generated from transferred solutions under the same total computational budget. Two alternatives of the 'inner' optimizer are computationally investigated, enhancing the efficiency in Section 4.3. One of the alternative obtains better solution quality of optimized solutions than the original Tr-DMOEAs.

## 4.1 Efficiency of Transfer Optimization in DMO

In this section, we quantify and present the computational cost of knowledge transfer in Tr-DMOEA from the perspectives of time complexity and computation time in seconds.

The time complexity of TCA and primal dual interior point has been analyzed in [25]. The major time cost of TCA is spent on the eigenvalue decomposition. It costs $O(d(m+n)^2)$ time when $d$ nonzero eigenvectors are to be extracted, where $m$ and $n$ are the numbers of the solutions which are generated in the search space of source and target problems. For the primal dual interior point method, suppose the constraint matrix $A$ has $n$ rows and $m$ columns, and $n < m$, it has $O(\sqrt{m}L)$ iterations and $O(m^3L)$ arithmetic operations, where $L$ is the total number of bits of the input. It is clear that the time complexity of the interior point is larger than that of TCA.

In order to verify the cost of TCA and interior point method from the perspective of computation time, an initial experiment is conducted regarding how much computation time these two parts consume in a single run. The experimental design is set as those in Section 3.1.1 except the following:

- Here, only one problem DMOP2 is used as the test problem as it is only used to reflect the proportion of computation time of each component in Tr-DMOEA.
- Population iterates for 50 generations both before and after an environmental change, and there is only one change.
- The changing severity is set as 10.
- Computation time of each function of the whole algorithm will be recorded by the Profile environment of MATLAB 2018b.

Computation time of each part of the Tr-RMMEDA is recorded and presented in Table 4.1. It is clear from the table that the interior point method consumes the most computation time, which confirms the complexity analysis of TCA and interior point in previous one section.

Table 4.1: Computation time of each process of Tr-RMMEDA with one run, one change and one parameter on problem DMOP2.

| Process | Computation time | Proportion |
|---|---|---|
| Interior point | 427.455 s | 93.9% |
| RMMEDA | 25.53 s | 5.6 % |
| TCA | 2.408 s | 0.5% |
| Sum | 455.393 s | 100 % |

## 4.2 Solutions Quality after Optimization for Which Transfer Cost Is Used

It has been experimentally shown that the existing Tr-DMOEA is extremely time-consuming. It is still unclear whether it is worthy to consume such long time to use transfer optimization in DMO. This section is presented to explore this.

### 4.2.1 Computation Time of Transfer Used for Optimization

In order to figure out whether it is worthwhile to use transfer optimization in DMO, this section designs an experiment to verify it. The main idea behind this experiment is to use the computation time of transfer to optimize randomly generated solutions.

Whenever there is a change, transfer leaning is used to get the transferred initial population, while another initial population is randomly generated in the search space. The costs of transfer optimization and random generation are recorded,

termed as $C_{tr}$ and $C_{ran}$. The cost is the running time determined based on the stopwatch timer in Matlab, where the Matlab command 'tic' and 'toc' starts and ends the timer, respectively. Then, the cost $C_{tr}$ - $C_{ran}$ is used to optimize the randomly generated population. During the optimization, both transferred population and random one will iterate for $\tau_t$ generations to get two optimized solutions.

All test problems in [31] and all parameter settings in Table 3.1 are used, which is the same as those in Section 3.1.1. In Tr-DMOEA, all the transfer procedures including the interior point method are used, which are the same as the original.

### 4.2.2 Quality Comparison of Transferred And Random Solutions with Same Cost Budget

The specific experimental design has been introduced in the previous section. In this section we compare the quality of optimized solutions with transfer and optimized solutions from random generation, both of which are obtained under the same computation time budget.

Table 4.2: Mean and standard deviations of MIGD values of transferred solutions and those solutions optimized using the computation time of transfer after being randomly generated.

| Prob. | C1 | | C2 | | C3 | | C4 | |
|---|---|---|---|---|---|---|---|---|
| Methods | Transfer | Random | Transfer | Random | Transfer | Random | Transfer | Random |
| FDA4 | 8.80e-02(2.50e-04) | 7.30e-02(2.45e-08) | 8.20e-02(2.05e-04) | 7.27e-02(2.01e-07) | 9.72e-02(3.95e-04) | 7.29e-02(5.38e-09) | 8.53e-02(2.38e-04) | 7.30e-02(4.06e-09) |
| FDA5 | 2.62e-01(1.83e-03) | 1.77e-01(4.85e-06) | 2.03e-01(1.68e-03) | 1.73e-01(8.88e-07) | 1.94e-01(1.61e-03) | 1.75e-01(3.22e-06) | 1.89e-01(2.54e-03) | 1.74e-01(2.77e-06) |
| $FDA5_{iso}$ | 5.80e-02(6.97e-05) | 2.59e-01(1.57e-06) | 5.91e-02(3.24e-05) | 2.56e-01(1.67e-06) | 6.15e-02(5.93e-05) | 2.69e-01(2.35e-06) | 6.18e-02(5.33e-05) | 2.52e-01(1.85e-06) |
| $FDA5_{dec}$ | 5.41e-01(2.23e-02) | 2.06e-01(7.60e-06) | 5.79e-01(2.71e-02) | 1.98e-01(4.12e-06) | 5.20e-01(1.16e-02) | 2.00e-01(3.92e-07) | 5.13e-01(9.76e-03) | 1.90e-01(1.41e-06) |
| DIMP2 | 9.61e+00(2.37e+00) | 3.57e+00(9.26e-04) | 9.97e+00(1.07e+00) | 3.53e+00(6.77e-05) | 9.03e+00(3.57e+00) | 3.54e+00(5.98e-04) | 9.77e+00(1.32e+00) | 3.56e+00(6.11e-04) |
| DMOP2 | 4.05e-01(3.69e-02) | 1.46e-02(1.59e-04) | 3.02e-01(1.78e-02) | 1.81e-02(2.69e-04) | 2.34e-01(5.82e-03) | 5.12e-03(4.52e-06) | 2.29e-01(5.17e-03) | 1.15e-02(8.32e-05) |
| $DMOP2_{iso}$ | 2.19e-03(7.48e-09) | 4.45e-03(3.35e-10) | 2.32e-03(7.88e-08) | 4.44e-03(6.10e-10) | 2.49e-03(1.48e-07) | 4.44e-03(1.48e-10) | 2.55e-03(1.35e-07) | 4.44e-03(2.34e-10) |
| $DMOP2_{dec}$ | 6.45e-01(1.24e-01) | 2.54e-02(1.87e-05) | 5.12e-01(4.01e-02) | 2.28e-02(2.81e-05) | 3.48e-01(5.99e-03) | 2.35e-02(2.17e-05) | 2.94e-01(5.18e-03) | 1.48e-02(4.17e-06) |
| DMOP3 | 3.14e-01(5.55e-02) | 3.28e-03(5.27e-11) | 1.38e-01(3.28e-03) | 3.28e-03(5.63e-11) | 1.39e-01(1.32e-02) | 3.29e-03(1.43e-11) | 7.20e-02(4.84e-03) | 3.30e-03(8.28e-11) |
| HE2 | 4.59e-01(2.49e-02) | 5.98e-02(6.63e-08) | 4.97e-01(1.38e-02) | 5.89e-02(2.62e-08) | 3.23e-01(1.07e-03) | 5.77e-02(5.15e-08) | 2.70e-01(5.19e-03) | 5.66e-02(8.82e-10) |
| HE7 | 2.24e-01(1.94e-04) | 3.70e-02(1.41e-08) | 2.29e-01(2.97e-04) | 3.72e-02(3.44e-09) | 2.48e-01(3.49e-04) | 3.70e-02(1.38e-08) | 2.64e-01(3.56e-04) | 3.71e-02(7.82e-10) |
| HE9 | 3.30e-01(2.51e-04) | 2.34e-01(7.89e-08) | 3.06e-01(4.54e-04) | 2.34e-01(1.43e-07) | 2.83e-01(2.29e-04) | 2.33e-01(1.82e-07) | 2.80e-01(4.85e-04) | 2.32e-01(7.33e-08) |

| Prob. | C5 | | C6 | | C7 | | C8 | |
|---|---|---|---|---|---|---|---|---|
| Methods | Transfer | Random | Transfer | Random | Transfer | Random | Transfer | Random |
| FDA4 | 7.23e-02(1.18e-04) | 5.99e-02(4.24e-09) | 6.86e-02(1.20e-04) | 6.01e-02(5.67e-09) | 7.01e-02(1.55e-04) | 7.32e-02(5.29e-09) | 7.35e-02(3.09e-04) | 7.33e-02(5.46e-09) |
| FDA5 | 1.35e+00(4.76e-01) | 1.43e-01(4.38e-07) | 1.86e+00(2.89e-01) | 1.44e-01(1.24e-06) | 1.24e+00(5.75e-01) | 1.70e-01(5.22e-07) | 6.70e-01(9.62e-03) | 1.74e-01(7.32e-06) |
| $FDA5_{iso}$ | 5.33e-01(1.37e-02) | 1.78e-01(1.47e-05) | 5.44e-01(6.02e-02) | 1.85e-01(6.78e-06) | 6.49e-01(2.01e-02) | 2.51e-01(9.83e-06) | 5.51e-01(4.31e-03) | 2.49e-01(9.90e-06) |
| $FDA5_{dec}$ | 2.28e+00(2.43e-01) | 1.59e-01(9.46e-06) | 2.33e+00(2.77e-01) | 1.59e-01(6.53e-06) | 2.14e+00(3.27e-01) | 1.87e-01(1.27e-06) | 1.21e+00(2.80e-01) | 1.90e-01(7.73e-06) |
| DIMP2 | 1.09e+01(9.25e-01) | 3.84e+00(1.07e-04) | 9.81e+00(9.88e-01) | 3.83e+00(1.06e-04) | 1.30e+01(1.59e+00) | 3.78e+00(3.37e-04) | 1.27e+01(1.63e+00) | 3.78e+00(6.19e-04) |
| DMOP2 | 1.10e+02(7.79e+01) | 1.81e+01(4.03e-04) | 1.13e+02(1.15e+02) | 1.80e+01(8.19e-05) | 2.70e+00(2.18e+00) | 5.15e-03(4.60e-06) | 3.64e+00(1.41e+00) | 5.26e-03(4.97e-06) |
| $DMOP2_{iso}$ | 4.36e-01(5.21e-08) | 9.08e-02(4.66e-10) | 4.36e-01(9.49e-08) | 1.12e-01(8.16e-11) | 2.67e-03(2.45e-07) | 4.44e-03(1.10e-10) | 2.89e-03(5.71e-07) | 4.45e-03(9.63e-11) |
| $DMOP2_{dec}$ | 5.38e+00(3.82e+01) | 1.74e-01(7.42e-10) | 3.94e+00(5.68e+00) | 1.94e-01(1.38e-10) | 2.04e+00(8.07e-01) | 2.33e-02(7.25e-05) | 2.69e+00(2.12e+00) | 1.71e-02(1.12e-05) |
| DMOP3 | 1.12e+02(4.89e+01) | 1.80e+01(2.02e-08) | 1.13e+02(2.98e+02) | 1.80e+01(4.12e-09) | 1.44e+00(8.44e-01) | 3.33e-03(2.49e-11) | 2.22e+00(9.56e-01) | 3.30e-03(6.02e-11) |
| HE2 | 1.25e-01(7.15e-04) | 5.64e-02(1.19e-08) | 1.21e-01(5.20e-04) | 5.49e-02(4.11e-10) | 6.06e-01(5.69e-03) | 5.84e-02(3.57e-08) | 5.02e-01(2.14e-03) | 5.66e-02(6.04e-10) |
| HE7 | 1.72e-01(1.14e-03) | 3.42e-02(1.17e-08) | 1.82e-01(1.94e-04) | 3.42e-02(2.40e-08) | 2.53e-01(4.60e-04) | 3.70e-02(4.02e-09) | 2.73e-01(5.03e-04) | 3.70e-02(1.33e-08) |
| HE9 | 2.76e-01(1.01e-04) | 2.09e-01(2.78e-07) | 2.58e-01(6.38e-04) | 2.08e-01(1.83e-07) | 3.19e-01(7.59e-05) | 2.34e-01(2.96e-08) | 2.79e-01(2.94e-04) | 2.31e-01(1.58e-07) |

There are 20 independent runs. Within each run, for each problem with each parameter setting, one solution set is the transferred solutions. Another solution set is obtained by optimization on randomly generated solutions using the computation time that transfer optimization consumes. The better values with smaller average that the method gets are shown with dark background. Wilcoxon Rank Sum test was used to indicate whether there is significant difference between the compared algorithms and the test result is h = 1 and p = 1.2624e-08. Specific can be found in the second paragraph of Section 4.2.2.1.

#### 4.2.2.1 Solution Quality Comparison in the First Generation

Here, we compare the quality of transferred solutions and those solutions which are optimized using the transfer computation time. IGD is used to measure the diversity and convergence of those two solution sets. MIGD is the average of these 20 IGD values under 20 changes for two solution sets. The mean and standard deviation of MIGD values are presented in Table 4.2, in which 'Transfer' and 'Optimize' refer to the MIGD values of these two solution sets. The better values that the method gets are shown with dark background.

In order to indicate the significance between transferred solutions and optimized solutions using transfer computation time, the Wilcoxon Rank Sum test with the significance level 0.05 is carried out across problem instances, as recommended by Demsar [90]. MIGD value of 'Transfer' and 'Optimize' for each problem with one parameter is regarded as one observation data for the test. The result with $h = 1$ and $p = 1.2624e\text{-}08$ shows that random solutions optimized using transfer computation time are significantly better than transferred solutions.

Table 4.3: Mean and standard deviations of MIGD values of optimized solutions with transfer and random solutions that first iterate for some generations which consume the same computation time as transfer optimization, and then are optimized the same generations as those with transfer.

| Prob. | C1 | | C2 | | C3 | | C4 | |
|---|---|---|---|---|---|---|---|---|
| Methods | Transfer | Random | Transfer | Random | Transfer | Random | Transfer | Random |
| FDA4 | 5.85e-02(6.23e-05) | 7.30e-02(2.52e-08) | 5.73e-02(4.51e-06) | 7.26e-02(3.46e-08) | 6.38e-02(6.46e-06) | 7.28e-02(5.53e-09) | 6.56e-02(1.88e-06) | 7.32e-02(4.77e-09) |
| FDA5 | 1.17e-01(3.18e-04) | 1.82e-01(2.80e-06) | 7.23e-02(2.49e-05) | 1.74e-01(1.04e-06) | 7.33e-02(2.75e-06) | 1.82e-01(3.32e-07) | 7.77e-02(3.69e-06) | 1.76e-01(2.84e-06) |
| $FDA5_{iso}$ | 6.12e-02(8.74e-06) | 2.61e-01(2.33e-06) | 7.45e-02(9.93e-06) | 2.57e-01(4.63e-06) | 8.48e-02(4.85e-06) | 2.77e-01(1.60e-05) | 8.76e-02(3.31e-06) | 2.62e-01(2.35e-06) |
| $FDA5_{dec}$ | 4.86e-01(9.24e-03) | 2.05e-01(4.34e-06) | 4.00e-01(8.08e-03) | 1.94e-01(3.20e-06) | 3.05e-01(5.72e-03) | 2.00e-01(1.61e-06) | 1.91e-01(1.96e-03) | 1.93e-01(1.43e-06) |
| DIMP2 | 7.02e+00(8.61e-01) | 3.57e+00(9.26e-04) | 5.43e+00(4.27e-01) | 3.53e+00(6.75e-05) | 3.64e+00(1.26e-01) | 3.54e+00(5.96e-04) | 3.18e+00(6.77e-03) | 3.56e+00(6.10e-04) |
| DMOP2 | 2.77e-01(2.33e-02) | 1.46e-02(1.57e-04) | 1.17e-01(1.35e-03) | 1.80e-02(2.68e-04) | 8.78e-03(4.43e-05) | 5.12e-03(4.50e-06) | 4.17e-03(5.86e-06) | 1.15e-02(8.40e-05) |
| $DMOP2_{iso}$ | 2.58e-03(5.02e-08) | 4.45e-03(1.12e-10) | 3.12e-03(1.45e-07) | 4.45e-03(1.06e-10) | 3.73e-03(6.53e-08) | 4.47e-03(1.32e-10) | 4.06e-03(2.81e-08) | 4.49e-03(1.42e-10) |
| $DMOP2_{dec}$ | 4.77e-01(4.35e-02) | 2.41e-02(1.87e-05) | 2.87e-01(2.41e-02) | 2.27e-02(2.69e-05) | 6.02e-02(4.78e-04) | 2.29e-02(2.37e-05) | 1.27e-02(1.87e-04) | 1.48e-02(4.15e-06) |
| DMOP3 | 2.19e-01(4.21e-02) | 3.29e-03(3.28e-11) | 4.90e-02(3.69e-04) | 3.30e-03(3.16e-11) | 4.21e-03(5.04e-07) | 3.29e-03(3.09e-11) | 3.51e-03(5.93e-09) | 3.30e-03(1.43e-11) |
| HE2 | 4.22e-01(1.83e-02) | 5.97e-02(7.59e-08) | 3.93e-01(8.25e-03) | 5.88e-02(2.45e-08) | 2.19e-01(2.25e-03) | 5.77e-02(4.34e-08) | 1.04e-01(6.91e-04) | 5.66e-02(4.21e-10) |
| HE7 | 1.62e-01(9.35e-05) | 3.71e-02(1.93e-08) | 1.23e-01(8.81e-05) | 3.72e-02(4.49e-09) | 7.07e-02(5.22e-05) | 3.70e-02(3.81e-09) | 5.01e-02(9.82e-06) | 3.70e-02(9.77e-09) |
| HE9 | 3.02e-01(1.43e-04) | 2.34e-01(7.38e-08) | 2.72e-01(1.82e-04) | 2.34e-01(1.30e-07) | 2.43e-01(4.81e-05) | 2.33e-01(1.45e-07) | 2.37e-01(2.59e-05) | 2.33e-01(1.64e-07) |
| Prob. | C5 | | C6 | | C7 | | C8 | |
| Methods | Transfer | Random | Transfer | Random | Transfer | Random | Transfer | Random |
| FDA4 | 5.06e-02(2.99e-07) | 6.01e-02(1.04e-08) | 5.63e-02(2.88e-06) | 5.99e-02(1.14e-08) | 5.08e-02(1.33e-06) | 7.31e-02(7.13e-09) | 5.62e-02(1.89e-06) | 7.28e-02(2.19e-08) |
| FDA5 | 7.45e-01(5.23e-02) | 1.43e-01(3.21e-06) | 3.35e-01(3.64e-03) | 1.39e-01(2.60e-06) | 8.96e-01(2.55e-01) | 1.71e-01(1.07e-06) | 2.84e-01(2.00e-03) | 1.72e-01(6.59e-06) |
| $FDA5_{iso}$ | 3.88e-01(8.47e-03) | 1.80e-01(1.55e-05) | 2.24e-01(1.08e-03) | 1.91e-01(4.60e-06) | 4.34e-01(6.13e-03) | 2.51e-01(1.01e-05) | 2.48e-01(1.82e-03) | 2.61e-01(8.09e-06) |
| $FDA5_{dec}$ | 1.57e+00(4.95e-01) | 1.61e-01(2.65e-06) | 4.35e-01(9.41e-03) | 1.57e-01(3.00e-06) | 1.39e+00(3.41e-01) | 1.89e-01(5.32e-06) | 4.09e-01(9.33e-03) | 1.89e-01(1.93e-06) |
| DIMP2 | 7.49e+00(7.21e-01) | 3.84e+00(1.03e-04) | 4.02e+00(3.28e-01) | 3.83e+00(1.06e-04) | 9.18e+00(1.04e+00) | 3.78e+00(3.36e-04) | 5.89e+00(1.17e-01) | 3.78e+00(6.20e-04) |
| DMOP2 | 9.16e+01(4.48e+01) | 1.81e+01(4.05e-04) | 7.42e+01(1.45e+01) | 1.80e+01(8.33e-05) | 7.34e-01(1.39e-01) | 5.13e-03(4.52e-06) | 4.46e-02(1.11e-02) | 5.23e-03(4.74e-06) |
| $DMOP2_{iso}$ | 4.36e-01(1.11e-08) | 9.08e-02(1.79e-10) | 4.36e-01(1.96e-09) | 1.12e-01(7.45e-11) | 3.39e-03(1.99e-07) | 4.46e-03(4.45e-10) | 4.35e-03(7.25e-08) | 4.50e-03(2.86e-10) |
| $DMOP2_{dec}$ | 7.66e-01(1.60e+00) | 1.74e-01(4.99e-10) | 4.32e-01(8.19e-08) | 1.94e-01(1.74e-10) | 5.40e-01(3.17e-02) | 2.30e-02(6.80e-05) | 8.76e-03(3.02e-05) | 1.68e-02(9.73e-06) |
| DMOP3 | 9.05e+01(1.17e+01) | 1.80e+01(1.38e-08) | 7.33e+01(1.01e+00) | 1.80e+01(1.28e-09) | 3.78e-01(7.07e-02) | 3.33e-03(1.51e-11) | 5.44e-03(3.04e-05) | 3.31e-03(2.85e-11) |
| HE2 | 6.95e-02(1.68e-04) | 5.64e-02(1.11e-08) | 5.36e-02(2.14e-06) | 5.49e-02(4.23e-09) | 5.38e-01(5.53e-03) | 5.84e-02(3.39e-08) | 1.57e-01(1.19e-03) | 5.66e-02(5.72e-10) |
| HE7 | 8.26e-02(1.95e-04) | 3.43e-02(7.46e-09) | 3.73e-02(1.17e-06) | 3.42e-02(2.07e-08) | 1.43e-01(2.91e-04) | 3.69e-02(3.33e-09) | 5.21e-02(8.34e-06) | 3.70e-02(2.96e-09) |
| HE9 | 2.21e-01(1.54e-04) | 2.09e-01(2.57e-07) | 1.80e-01(4.12e-05) | 2.08e-01(1.71e-07) | 2.86e-01(2.78e-05) | 2.34e-01(3.44e-08) | 2.54e-01(9.71e-05) | 2.31e-01(1.70e-07) |

There are 20 independent runs. Within each run, for each problem with each parameter setting, 'Transfer' solution set means the optimized solutions with transfer. 'Random' solution set means random generated solutions that first iterate for some generations which consume the same computation time as transfer optimization, and then are optimized the same generations as those with transfer. The better values with smaller average that the method gets are shown with dark background. Wilcoxon Rank Sum test was used to indicate whether there is significant difference between the compared algorithms and the test result is h = 1 and p = 0.0085. Specific can be found in the second paragraph of Section 4.2.2.2.

#### 4.2.2.2 Solution Quality Comparison after Optimization

Here, we compare the quality of optimized solutions with transfer and those solutions which originate from random generation. Also, IGD is used to measure the diversity and convergence of those two solution sets. MIGD is the average of these 20 IGD values under 20 changes for two solution sets. The mean and standard deviation of MIGD values are presented in Table 4.3, in which 'Transfer' and 'Optimize' refer to the MIGD values of these two solution sets. The better values that the method gets are shown with dark background.

In order to indicate the significance between optimized solutions with transfer and those from random generation, the Wilcoxon Rank Sum test with the significance level 0.05 is carried out. MIGD value of 'Transfer' and 'Optimize' for each problem with one parameter is regarded as one observation data for the test. The result with $h = 1$ and $p = 0.0085$ shows that solutions which originates from random generation are significantly better than optimized solutions with transfer in the last generation of optimization.

It can be concluded that under the same computation time budget, random solution after optimization will obtain better results than those from transfer optimization, no matter whether the optimization process is conducted on transferred solutions or not. In the original paper, Tr-DMOEA is better than the algorithm with random solutions. The difference is that in the original paper, the same evaluation times and generations are given to these two algorithms, while the same computation time budget is given in this paper. This shows that it is not worth to consume such long time on transfer optimization, while achieving worse results than consuming these computation time on optimizing. In other words, it vanishes the original purpose of using transfer optimization in DMO, reducing the efforts of optimization.

On the other side, when comparing Tables 4.2 and 4.3, it is clear that the improvement of transferred solutions are more than that of optimized solution from randomization after $\tau_t$ generations' optimization. In addition, as $\tau_t$ increases, the quality of transferred solutions becomes better while those from randomization remain unchanged. These two observations show that transferred solutions are far from the optimum of problems, while solutions from randomization have already been nearly Pareto optimal.

## 4.3 Computational Analysis of Three Tr-DMOEA Variants

Given that the computation time of solving the inner optimization problem in Tr-DMOEA is very large, it is unclear whether the efficiency of solving the inner problem could be enhanced. To explore this, other two popular optimization methods are used here, which are the active set and sequential quadratic programming (SQP) methods. A set of experiments are conducted to validate the efficiency and effectiveness of these three Tr-DMOEA variants. In this section, we aim to determine which transfer algorithm with which 'inner' optimizer obtains fastest computation time or best solution quality among three algorithms. Therefore, we use Friedman test for comparison of multiple groups to see whether there are significant differences among the three Tr-DMOEAs variants.

### 4.3.1 Computation Time Comparison of Three Tr-DMOEA Variants

The computation time of three different optimization methods in these three Tr-DMOEA variants is recorded when solving the inner problem, for all test problems with all parameter setting.

Given that the computation time of solving the inner optimization problem in Tr-DMOEA is very large, it is unclear whether the efficiency of solving the inner problem (equation (3.2)) could be enhanced. To explore this, other two popular optimization methods are used here, which are the active set [97] and sequential quadratic programming (SQP) [97] methods. We have conducted a set of experiments to validate the efficiency and effectiveness of these three Tr-DMOEA variants.

Here, we give a brief description of active set and SQP method:

- The active set method [97] considers an optimization problem with $n$ constraints $g_1(x) \geq 0, ...g_n(x) \geq 0$. For a point $x$ in the feasible region, a constraint is called active at $x$ if $g_i(x) = 0$ and inactive if $g_i(x) > 0$. The procedures for the active set are as follows: solve the equality problem defined by the active set (approximately) with a solution $x_*$; compute the Lagrange multipliers of the active set; remove a subset of the constraints with negative Lagrange multipliers; search for the new boundary based on $x_*$ along the feasible region boundary formed by the active set and add the constraint related to the new boundary to the approximated active set. Iterate these procedures until the optima is found.

- SQP [97] is an iterative method for constrained nonlinear optimization. At each iteration, a basic sequential quadratic programming algorithm defines an appropriate search direction as a solution to the quadratic programming subproblem.

Note that all three optimization methods (interior point, SQP and active set) used in this paper are from the optimization toolbox of MATLAB 2018b. The specific implementation of these three methods and the difference between the SQP and the active-set algorithms can be found in MATLAB's online documents[2].

All experimental settings are the same as those in Tr-DMOEA [25] and Section 3.1.1 except the following:

1. In three Tr-DMOEA variants, three different optimization methods interior point, active set and SQP methods are used, respectively. The other steps of the Tr-DMOEA are the same as in the previous work [25].

The computation time of three different optimization methods in these three Tr-DMOEA variants is only recorded when solving the inner problem (equation (3.2)), for all test problems with all parameter setting. The specific computation time comparisons of these three optimization methods in DMO, the comparison results of computation time of three Tr-DMOEA variants are shown in Table 4.4. Due to space limitations, only results of C5 and C6 are presented. Results of other parameter settings are similar to those of C5 and C6.

It is clear from Table 4.4 that for all test problems with those parameter settings, the interior point method consumes the most time among the three compared methods to solve the inner optimization problem, while the active set method is the most efficient optimization algorithm. In addition, the time interior points method consumes is several times over that another two methods consume. This shows that another two inner optimization methods can greatly improve the efficiency of transfer optimization in DMO.

Friedman and Nemenyi statistical tests [90] were carried out across benchmark problems. The computation time that all algorithms get on one problem with one parameter setting is regarded as an observation of the test. Therefore, there are 96 (12 problems and 8 parameters) observed data. Friedman detects significant differences in average computation time with a p-value of 2.0311e-42. The Nemenyi post-tests are shown in Figure 4.1(a). This shows that the SQP and active set method are significantly more efficient than the interior point method.

[2]https://ww2.mathworks.cn/help/optim/ug/constrained-nonlinear-optimization-algorithms.html.

Table 4.4: Mean and standard deviations of computation time (in seconds) of three optimization methods for minimizing the distance of source and target problems in the latent space with 20 environment changes for all problems with all parameter settings.

| Prob. | C1 | | | C2 | | |
|---|---|---|---|---|---|---|
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 5.77e+03(2.00e+05) | 6.92e+02(2.06e+03) | 4.81e+02(1.45e+03) | 6.10e+03(1.14e+05) | 1.12e+03(2.64e+03) | 7.87e+02(2.63e+03) |
| FDA5 | 3.42e+03(1.02e+05) | 4.62e+02(1.06e+03) | 2.75e+02(2.25e+02) | 4.40e+03(1.55e+05) | 5.87e+02(1.11e+03) | 3.72e+02(4.02e+02) |
| FDA5$_{iso}$ | 1.98e+03(2.32e+04) | 6.85e+02(1.07e+03) | 4.31e+02(2.36e+02) | 2.01e+03(2.06e+04) | 6.56e+02(6.55e+02) | 4.19e+02(2.80e+02) |
| FDA5$_{dec}$ | 2.30e+03(5.37e+03) | 2.74e+02(5.47e+02) | 1.82e+02(4.79e+01) | 2.92e+03(4.60e+04) | 3.46e+02(7.24e+02) | 2.25e+02(3.59e+02) |
| DIMP2 | 8.31e+02(2.54e+04) | 1.97e+01(1.21e+01) | 1.81e+01(7.75e+00) | 9.26e+02(1.32e+04) | 2.74e+01(1.39e+01) | 1.87e+01(4.72e+00) |
| DMOP2 | 7.91e+02(1.47e+04) | 7.67e+01(2.67e+02) | 4.35e+01(1.88e+01) | 9.75e+02(1.38e+04) | 1.53e+02(6.71e+02) | 6.99e+01(1.10e+02) |
| DMOP2$_{iso}$ | 1.29e+03(3.91e+03) | 5.85e+02(2.26e+04) | 3.65e+02(5.10e+02) | 1.29e+03(3.35e+03) | 5.81e+02(2.22e+04) | 3.59e+02(5.84e+02) |
| DMOP2$_{dec}$ | 5.81e+02(3.87e+03) | 4.47e+01(2.80e+02) | 2.42e+01(6.92e+00) | 7.01e+02(2.95e+03) | 4.70e+01(2.17e+02) | 2.67e+01(1.58e+01) |
| DMOP3 | 1.02e+03(2.28e+04) | 1.05e+02(1.86e+02) | 5.90e+01(5.06e+01) | 1.62e+03(5.37e+04) | 1.89e+02(4.49e+02) | 9.26e+01(1.04e+02) |
| HE2 | 1.44e+03(1.23e+04) | 1.98e+02(1.02e+03) | 1.39e+02(2.11e+03) | 1.92e+03(2.26e+04) | 2.46e+02(1.43e+03) | 1.27e+02(1.36e+03) |
| HE7 | 4.43e+03(1.81e+05) | 7.29e+02(2.47e+03) | 6.59e+02(3.10e+03) | 6.12e+03(3.26e+05) | 1.03e+03(3.34e+03) | 9.40e+02(5.27e+03) |
| HE9 | 2.58e+03(1.17e+05) | 6.94e+02(6.85e+02) | 5.00e+02(3.62e+03) | 3.42e+03(2.02e+05) | 9.27e+02(2.03e+03) | 6.61e+02(3.32e+03) |
| Prob. | C3 | | | C4 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 6.29e+03(1.18e+05) | 1.18e+03(2.09e+03) | 8.41e+02(2.25e+03) | 6.34e+03(1.21e+05) | 1.16e+03(1.38e+03) | 8.24e+02(2.59e+03) |
| FDA5 | 6.09e+03(2.28e+05) | 8.27e+02(2.15e+03) | 5.29e+02(7.62e+02) | 6.86e+03(2.92e+05) | 1.17e+03(2.97e+03) | 7.28e+02(6.51e+02) |
| FDA5$_{iso}$ | 2.00e+03(1.56e+04) | 6.32e+02(5.44e+02) | 4.03e+02(1.26e+02) | 1.99e+03(1.36e+04) | 6.18e+02(5.16e+02) | 3.96e+02(9.27e+01) |
| FDA5$_{dec}$ | 3.72e+03(1.74e+04) | 3.58e+02(1.10e+03) | 3.20e+02(5.65e+01) | 4.83e+03(7.96e+03) | 4.05e+02(9.50e+02) | 3.66e+02(7.80e+02) |
| DIMP2 | 1.22e+03(1.94e+04) | 3.14e+01(2.85e+01) | 1.91e+01(1.77e+01) | 6.38e+03(7.18e+05) | 1.18e+02(1.28e+02) | 6.53e+01(1.14e+02) |
| DMOP2 | 4.40e+03(2.53e+05) | 4.05e+02(1.99e+03) | 2.22e+02(1.36e+02) | 5.84e+03(2.92e+05) | 4.65e+02(2.99e+03) | 2.79e+02(1.48e+02) |
| DMOP2$_{iso}$ | 1.33e+03(3.75e+03) | 5.72e+02(2.10e+04) | 3.50e+02(6.69e+02) | 1.40e+03(4.97e+03) | 5.66e+02(2.11e+04) | 3.46e+02(5.97e+02) |
| DMOP2$_{dec}$ | 1.59e+03(3.23e+04) | 1.06e+02(1.85e+03) | 5.54e+01(5.94e+01) | 4.27e+03(5.86e+04) | 3.13e+02(4.46e+03) | 2.11e+02(2.07e+03) |
| DMOP3 | 6.07e+03(6.10e+05) | 6.21e+02(4.68e+03) | 3.03e+02(4.43e+02) | 7.11e+03(1.11e+06) | 6.48e+02(7.40e+03) | 3.19e+02(8.00e+02) |
| HE2 | 3.60e+03(1.60e+05) | 6.33e+02(1.95e+04) | 2.20e+02(3.47e+03) | 1.09e+04(1.23e+06) | 1.39e+03(3.46e+04) | 6.10e+02(4.51e+03) |
| HE7 | 1.09e+04(5.87e+05) | 1.91e+03(1.35e+04) | 1.70e+03(1.65e+04) | 1.23e+04(6.97e+05) | 2.65e+03(2.11e+04) | 2.39e+03(1.63e+04) |
| HE9 | 6.96e+03(8.56e+05) | 1.75e+03(4.42e+03) | 1.23e+03(1.10e+04) | 8.10e+03(1.14e+06) | 2.15e+03(1.33e+03) | 1.62e+03(1.36e+04) |
| Prob. | C5 | | | C6 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 6.38e+03(8.93e+04) | 8.77e+02(1.28e+03) | 4.59e+02(7.89e+02) | 6.42e+03(1.34e+05) | 8.31e+02(7.18e+02) | 4.25e+02(7.10e+02) |
| FDA5 | 3.48e+03(1.46e+05) | 3.64e+02(5.91e+02) | 2.00e+02(6.25e+02) | 5.67e+03(2.38e+05) | 7.62e+02(8.12e+02) | 3.70e+02(1.58e+02) |
| FDA5$_{iso}$ | 1.36e+03(8.46e+03) | 4.18e+02(4.03e+02) | 2.69e+02(1.64e+02) | 1.31e+03(1.35e+04) | 4.18e+02(2.77e+02) | 2.69e+02(7.39e+01) |
| FDA5$_{dec}$ | 3.03e+03(1.56e+04) | 3.46e+02(8.70e+02) | 1.44e+02(2.95e+02) | 4.74e+03(5.48e+04) | 7.01e+02(4.91e+03) | 3.11e+02(2.48e+02) |
| DIMP2 | 9.62e+02(2.11e+04) | 2.80e+01(2.25e+01) | 2.11e+01(1.29e+01) | 6.74e+03(1.78e+06) | 1.49e+02(4.52e+02) | 7.74e+01(2.76e+02) |
| DMOP2 | 7.26e+02(8.73e+03) | 3.15e+02(1.35e+03) | 1.03e+02(1.43e+02) | 5.35e+03(2.68e+05) | 3.69e+02(1.11e+03) | 2.05e+02(2.84e+01) |
| DMOP2$_{iso}$ | 1.34e+03(5.58e+03) | 5.97e+02(2.33e+04) | 4.44e+02(1.25e+03) | 1.44e+03(5.92e+03) | 5.78e+02(2.20e+04) | 4.33e+02(1.47e+03) |
| DMOP2$_{dec}$ | 5.81e+02(3.71e+03) | 5.07e+01(5.06e+01) | 1.86e+01(3.25e+00) | 3.84e+03(6.01e+04) | 3.47e+02(4.55e+02) | 1.97e+02(1.84e+01) |
| DMOP3 | 8.64e+02(2.20e+04) | 2.98e+02(1.46e+03) | 1.13e+02(2.51e+02) | 5.45e+03(4.75e+05) | 3.88e+02(4.98e+02) | 1.99e+02(1.07e+02) |
| HE2 | 2.11e+03(5.02e+04) | 4.35e+02(5.15e+03) | 3.28e+02(3.65e+03) | 1.35e+04(1.52e+06) | 2.29e+03(7.51e+04) | 1.55e+03(5.45e+04) |
| HE7 | 5.84e+03(3.36e+05) | 1.28e+03(5.17e+03) | 1.15e+03(6.44e+03) | 1.07e+04(1.94e+06) | 2.81e+03(1.29e+04) | 2.47e+03(4.82e+04) |
| HE9 | 2.91e+03(1.76e+05) | 8.51e+02(5.40e+02) | 6.68e+02(1.71e+03) | 6.47e+03(7.73e+05) | 1.79e+03(6.05e+02) | 1.50e+03(1.06e+04) |
| Prob. | C7 | | | C8 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 6.35e+03(1.39e+05) | 1.20e+03(1.38e+03) | 8.72e+02(2.05e+03) | 6.63e+03(1.68e+05) | 1.24e+03(1.40e+03) | 8.97e+02(1.66e+03) |
| FDA5 | 4.82e+03(1.65e+05) | 6.98e+02(2.06e+03) | 4.28e+02(2.39e+02) | 6.31e+03(3.35e+05) | 1.21e+03(3.13e+03) | 7.28e+02(5.18e+02) |
| FDA5$_{iso}$ | 1.93e+03(2.63e+04) | 6.90e+02(7.43e+02) | 4.55e+02(1.07e+02) | 1.95e+03(2.89e+04) | 6.67e+02(8.07e+02) | 4.42e+02(5.77e+01) |
| FDA5$_{dec}$ | 2.84e+03(5.16e+04) | 3.23e+02(5.02e+02) | 2.27e+02(7.11e+01) | 4.63e+03(1.20e+05) | 4.18e+02(9.77e+02) | 3.95e+02(1.09e+02) |
| DIMP2 | 9.09e+02(1.58e+04) | 2.59e+01(2.39e+01) | 1.76e+01(1.04e+01) | 6.13e+03(6.20e+05) | 1.32e+02(4.46e+02) | 7.06e+01(1.16e+02) |
| DMOP2 | 1.06e+03(1.37e+04) | 1.14e+02(3.17e+02) | 5.67e+01(2.14e+01) | 5.91e+03(1.73e+05) | 4.30e+02(2.07e+03) | 2.58e+02(1.12e+02) |
| DMOP2$_{iso}$ | 1.31e+03(4.63e+03) | 5.61e+02(1.85e+04) | 3.61e+02(6.70e+02) | 1.41e+03(5.73e+03) | 5.45e+02(1.82e+04) | 3.49e+02(5.41e+02) |
| DMOP2$_{dec}$ | 6.95e+02(5.07e+03) | 3.87e+01(4.13e+01) | 3.11e+01(2.02e+01) | 4.33e+03(5.18e+04) | 3.20e+02(4.13e+03) | 2.03e+02(2.91e+03) |
| DMOP3 | 1.70e+03(8.16e+04) | 1.38e+02(2.18e+02) | 7.47e+01(4.06e+01) | 6.83e+03(6.14e+05) | 5.70e+02(3.91e+03) | 2.97e+02(3.71e+02) |
| HE2 | 1.94e+03(3.58e+04) | 2.41e+02(1.14e+03) | 1.29e+02(1.05e+03) | 1.15e+04(6.54e+05) | 1.43e+03(2.50e+04) | 5.58e+02(4.74e+03) |
| HE7 | 6.24e+03(1.61e+05) | 1.03e+03(2.26e+03) | 9.33e+02(7.25e+03) | 1.20e+04(1.00e+06) | 2.67e+03(1.18e+04) | 2.37e+03(2.70e+04) |
| HE9 | 3.40e+03(1.96e+05) | 9.31e+02(8.21e+02) | 6.73e+02(2.55e+03) | 8.02e+03(1.16e+06) | 2.06e+03(9.06e+02) | 1.61e+03(1.55e+04) |

There are 20 independent runs. Within each run, for each problem with each parameter setting, there are 20 changes. After each change, three optimization algorithms (interior point, SQP and active set) are used to find the minimal distance, respectively. The computation time of each algorithm is recorded for each change. The values in this table are the mean and standard deviation of total computation time of 20 changes under 20 runs. The smallest computation time for each problem is highlighted in bold.

### 4.3.2 Solutions Quality Comparison of Three Tr-DMOEA Variants

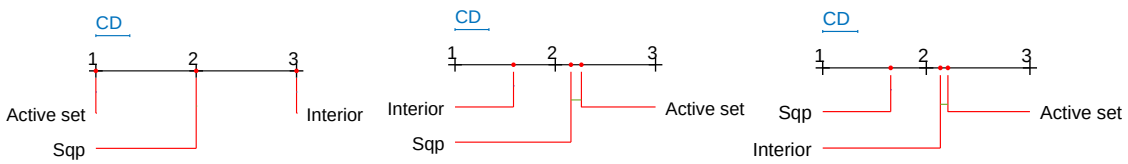
In order to evaluate the influence of the improved efficiency on the solution quality, in this section, transferred solutions of these Tr-DMOEAs in the first generation after change are first compared. The optimized solution of these Tr-DMOEAs in the last generation after change are then compared.

#### 4.3.2.1 Solutions Quality Comparison in the First Generation

Inverted Generational Distance (IGD) [89] is used to compare the quality of solution sets obtained by these three Tr-DMOEAs. IGD can measure the diversity and convergence of a solution set found by an algorithm, so it can give us a comprehensive understanding about the performance of the compared algorithms. MIGD [89] is a modified version of IGD, which is the average IGD values in all changes. The smaller the MIGD the better the algorithm.

For each Tr-DMOEA on one problem with one parameter setting, in the first generation after each change, a solution set obtained by transfer optimization in the experiment of Section 4.3.1 is used to calculate the IGD value. MIGD is the average of these 20 IGD values under 20 changes. Therefore, these three Tr-DMOEA variants will obtain one MIGD value on each problem for each parameter. Each algorithm is independently run 20 times on each problem instance with each parameter setting. The mean and standard deviation of MIGD values of transferred solutions obtained by the three Tr-DMOEAs in the first generation after changes are shown in Table 4.5.

In order to show the significant superiority of one method to others, Friedman and Nemenyi statistical tests were carried out across benchmark problems following Demsar's recommendation [90]. The MIGD values that all algorithms get on one




(a) Computation time (b) MIGD at 1st g (c) MIGD at the last g

Figure 4.1: Friedman ranking for three optimization methods (active set, SQP and interior point) from left to right regarding (a) computation time, (b) MIGD of solutions obtained at the first generation and (c) MIGD of optimized solutions at the last generation. Any pair of approaches whose distance between them is larger than CD are considered to be significantly different based on the Nemeyi post-hoc test.

Table 4.5: Mean and standard deviations of MIGD values of transferred solutions in the first generation after change obtained by three Tr-RMMEDA variants with different inner optimization algorithms.

| Prob. | C1 | | | C2 | | |
|---|---|---|---|---|---|---|
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 8.39e-02(7.14e-05) | 1.90e-01(1.11e-04) | 2.02e-01(1.50e-04) | 8.38e-02(4.84e-06) | 1.33e-01(2.45e-05) | 1.40e-01(1.42e-05) |
| FDA5 | 9.65e-01(1.30e-02) | 9.79e-01(1.31e-02) | 1.01e+00(1.25e-02) | 8.84e-01(1.06e-02) | 8.90e-01(1.37e-02) | 8.35e-01(7.58e-03) |
| $FDA5_{iso}$ | 6.14e-01(9.75e-03) | 4.99e-01(2.51e-03) | 5.16e-01(3.36e-03) | 3.93e-01(6.96e-04) | 4.01e-01(4.27e-04) | 4.08e-01(2.56e-04) |
| $FDA5_{dec}$ | 1.27e+00(1.76e-02) | 8.75e-01(2.67e-03) | 8.21e-01(6.81e-03) | 1.19e+00(1.88e-02) | 7.95e-01(3.68e-03) | 7.45e-01(5.01e-03) |
| DIMP2 | 1.55e+01(2.88e-01) | 1.80e+01(6.98e-01) | 1.72e+01(4.33e-01) | 1.51e+01(1.47e-01) | 1.72e+01(1.86e-01) | 1.68e+01(2.54e-01) |
| DMOP2 | 1.67e+00(3.18e-02) | 5.45e+00(1.58e+00) | 5.40e+00(1.29e+00) | 1.62e+00(4.41e-02) | 5.33e+00(5.58e-01) | 5.17e+00(8.46e-01) |
| $DMOP2_{iso}$ | 2.92e-03(3.01e-09) | 4.03e-03(4.11e-09) | 4.03e-03(4.95e-09) | 3.33e-03(5.76e-09) | 4.14e-03(1.68e-09) | 4.13e-03(3.92e-09) |
| $DMOP2_{dec}$ | 2.22e+00(5.59e-01) | 2.70e+01(4.46e+00) | 2.64e+01(6.88e+00) | 1.73e+00(8.48e-02) | 2.90e+01(6.25e+00) | 2.76e+01(5.46e+00) |
| DMOP3 | 8.95e-01(2.13e-02) | 3.93e+00(7.51e-01) | 3.39e+00(7.73e-01) | 9.25e-01(2.21e-02) | 3.90e+00(2.35e-01) | 3.59e+00(4.43e-01) |
| HE2 | 7.20e-01(6.56e-03) | 4.97e-01(1.44e-02) | 5.25e-01(1.23e-02) | 6.59e-01(4.75e-03) | 4.18e-01(5.47e-04) | 4.79e-01(6.48e-04) |
| HE7 | 3.49e-01(2.22e-05) | 4.07e-01(4.94e-05) | 4.10e-01(9.44e-05) | 3.36e-01(2.58e-05) | 3.90e-01(4.78e-05) | 3.90e-01(2.97e-05) |
| HE9 | 3.44e-01(9.09e-05) | 3.31e-01(7.45e-05) | 3.52e-01(4.66e-05) | 3.21e-01(8.39e-05) | 3.10e-01(6.60e-05) | 3.31e-01(7.45e-05) |
| Prob. | C3 | | | C4 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 8.93e-02(3.92e-06) | 1.24e-01(2.01e-05) | 1.31e-01(1.02e-05) | 9.57e-02(3.04e-06) | 1.26e-01(1.21e-05) | 1.32e-01(1.56e-05) |
| FDA5 | 5.69e-01(8.86e-03) | 7.04e-01(6.25e-03) | 6.42e-01(8.40e-03) | 4.55e-01(3.40e-03) | 4.95e-01(1.12e-03) | 5.10e-01(8.62e-04) |
| $FDA5_{iso}$ | 3.91e-01(5.50e-04) | 4.08e-01(4.88e-04) | 4.16e-01(2.12e-04) | 3.86e-01(2.09e-04) | 4.07e-01(2.31e-04) | 4.06e-01(2.90e-04) |
| $FDA5_{dec}$ | 1.04e+00(7.86e-03) | 7.84e-01(3.41e-03) | 6.84e-01(2.05e-03) | 7.38e-01(6.50e-03) | 7.22e-01(7.10e-03) | 6.38e-01(1.88e-03) |
| DIMP2 | 1.43e+01(2.84e-01) | 1.63e+01(1.56e-01) | 1.60e+01(2.25e-01) | 1.20e+01(1.18e-01) | 1.39e+01(1.88e-01) | 1.44e+01(1.14e-01) |
| DMOP2 | 1.57e+00(2.69e-02) | 2.92e+00(5.16e-01) | 3.13e+00(4.02e-01) | 1.48e+00(5.84e-02) | 2.80e+00(7.04e-01) | 2.31e+00(5.11e-01) |
| $DMOP2_{iso}$ | 3.94e-03(3.08e-09) | 4.30e-03(2.87e-09) | 4.31e-03(2.43e-09) | 4.26e-03(3.89e-09) | 4.39e-03(2.50e-09) | 4.38e-03(3.94e-09) |
| $DMOP2_{dec}$ | 1.58e+00(5.19e-02) | 2.24e+01(8.19e+00) | 2.26e+01(1.43e+01) | 1.33e+00(2.94e-02) | 1.19e+01(3.96e+00) | 1.25e+01(4.98e+00) |
| DMOP3 | 9.35e-01(2.73e-02) | 2.23e+00(4.86e-01) | 2.29e+00(2.26e-01) | 8.16e-01(1.26e-02) | 2.16e+00(5.11e-01) | 2.05e+00(2.23e-01) |
| HE2 | 5.07e-01(2.40e-04) | 3.38e-01(2.18e-04) | 4.44e-01(3.12e-04) | 4.54e-01(2.47e-04) | 2.93e-01(2.81e-04) | 4.22e-01(2.54e-04) |
| HE7 | 3.22e-01(1.10e-05) | 3.68e-01(1.79e-05) | 3.70e-01(4.12e-05) | 3.21e-01(2.71e-05) | 3.67e-01(1.28e-05) | 3.65e-01(3.24e-05) |
| HE9 | 2.92e-01(4.16e-05) | 2.86e-01(1.76e-05) | 3.04e-01(4.42e-05) | 2.77e-01(2.84e-05) | 2.75e-01(1.58e-05) | 2.88e-01(1.70e-05) |
| Prob. | C5 | | | C6 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 9.85e-02(3.99e-06) | 1.00e-01(6.48e-06) | 1.03e-01(1.32e-05) | 1.06e-01(7.04e-06) | 1.12e-01(1.35e-05) | 1.18e-01(2.15e-05) |
| FDA5 | 1.28e+00(7.92e-03) | 6.23e-01(1.65e-03) | 5.72e-01(1.46e-03) | 1.22e+00(8.41e-03) | 5.15e-01(7.53e-04) | 5.07e-01(3.91e-04) |
| $FDA5_{iso}$ | 5.39e-01(2.67e-03) | 5.42e-01(6.59e-03) | 6.79e-01(4.93e-03) | 5.07e-01(1.15e-03) | 5.31e-01(2.75e-03) | 6.42e-01(5.70e-03) |
| $FDA5_{dec}$ | 2.11e+00(4.11e-02) | 1.13e+00(1.44e-02) | 9.91e-01(1.92e-02) | 2.69e+00(4.38e-02) | 6.93e-01(7.07e-04) | 6.36e-01(2.28e-03) |
| DIMP2 | 1.53e+01(2.25e-01) | 1.79e+01(4.41e-01) | 1.69e+01(2.74e-01) | 1.21e+01(1.53e-01) | 1.44e+01(9.16e-02) | 1.44e+01(2.01e-01) |
| DMOP2 | 3.92e+01(3.28e-01) | 1.82e+01(2.64e-04) | 1.82e+01(4.64e-04) | 3.52e+01(1.01e+00) | 1.81e+01(2.52e-04) | 1.81e+01(1.85e-04) |
| $DMOP2_{iso}$ | 8.50e-02(1.29e-05) | 8.85e-02(8.31e-06) | 8.68e-02(6.61e-06) | 1.07e-01(2.36e-05) | 1.10e-01(7.90e-06) | 1.10e-01(1.18e-05) |
| $DMOP2_{dec}$ | 8.86e+00(2.38e+00) | 1.57e+05(4.70e+11) | 8.08e+05(1.95e+12) | 6.72e+00(4.26e-01) | 2.76e+00(4.07e+00) | 2.85e+00(4.32e+00) |
| DMOP3 | 3.70e+01(3.75e-01) | 1.82e+01(4.23e-04) | 1.82e+01(5.34e-04) | 3.39e+01(8.30e-01) | 1.82e+01(5.21e-04) | 1.82e+01(5.19e-04) |
| HE2 | 7.20e-01(4.59e-02) | 5.89e-01(1.69e-02) | 4.72e-01(2.68e-02) | 2.36e-01(9.37e-05) | 2.81e-01(2.57e-04) | 2.25e-01(3.64e-04) |
| HE7 | 3.05e-01(3.48e-05) | 3.46e-01(6.97e-05) | 3.48e-01(6.82e-05) | 2.85e-01(2.29e-05) | 3.18e-01(2.12e-05) | 3.17e-01(3.26e-05) |
| HE9 | 3.12e-01(5.27e-05) | 3.07e-01(1.73e-05) | 3.19e-01(8.23e-05) | 2.69e-01(3.58e-05) | 2.84e-01(1.08e-05) | 2.85e-01(1.55e-05) |
| Prob. | C7 | | | C8 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 8.26e-02(3.44e-06) | 1.26e-01(1.90e-05) | 1.32e-01(1.79e-05) | 9.36e-02(3.58e-06) | 1.18e-01(1.12e-05) | 1.24e-01(1.82e-05) |
| FDA5 | 7.74e-01(1.27e-02) | 7.86e-01(1.48e-02) | 7.55e-01(7.42e-03) | 4.00e-01(1.35e-03) | 4.42e-01(1.89e-03) | 4.72e-01(1.23e-03) |
| $FDA5_{iso}$ | 3.75e-01(7.57e-04) | 3.74e-01(2.47e-04) | 3.90e-01(3.95e-04) | 3.67e-01(1.09e-04) | 3.82e-01(1.56e-04) | 3.89e-01(1.88e-04) |
| $FDA5_{dec}$ | 1.12e+00(1.00e-02) | 7.13e-01(3.99e-03) | 6.78e-01(6.90e-03) | 6.73e-01(5.80e-03) | 6.39e-01(1.85e-03) | 5.89e-01(2.75e-04) |
| DIMP2 | 1.57e+01(1.81e-01) | 1.77e+01(2.93e-01) | 1.65e+01(4.96e-01) | 1.24e+01(8.00e-02) | 1.39e+01(2.37e-01) | 1.39e+01(1.81e-01) |
| DMOP2 | 1.51e+00(4.80e-02) | 6.06e+00(6.84e-01) | 5.63e+00(5.73e-01) | 1.40e+00(1.45e-02) | 3.00e+00(2.95e-01) | 3.25e+00(3.64e-01) |
| $DMOP2_{iso}$ | 3.23e-03(8.26e-09) | 4.13e-03(2.70e-09) | 4.12e-03(4.46e-09) | 4.23e-03(3.91e-09) | 4.38e-03(4.08e-09) | 4.37e-03(3.70e-09) |
| $DMOP2_{dec}$ | 1.81e+00(1.52e-01) | 2.61e+01(8.02e+00) | 2.50e+01(7.57e+00) | 1.41e+00(2.59e-02) | 1.26e+01(6.02e+00) | 1.24e+01(5.68e+00) |
| DMOP3 | 8.30e-01(2.18e-02) | 4.44e+00(4.49e-01) | 4.01e+00(4.61e-01) | 8.08e-01(1.46e-02) | 2.70e+00(2.19e-01) | 2.41e+00(3.29e-01) |
| HE2 | 6.15e-01(1.07e-03) | 4.31e-01(7.16e-04) | 4.79e-01(8.21e-04) | 4.42e-01(1.70e-04) | 2.86e-01(2.24e-04) | 4.45e-01(2.36e-04) |
| HE7 | 3.38e-01(2.13e-05) | 3.89e-01(5.18e-05) | 3.88e-01(3.77e-05) | 3.22e-01(2.02e-05) | 3.67e-01(1.23e-05) | 3.65e-01(1.25e-05) |
| HE9 | 3.19e-01(7.06e-05) | 3.08e-01(3.58e-05) | 3.26e-01(4.94e-05) | 2.71e-01(3.72e-05) | 2.69e-01(2.00e-05) | 2.81e-01(2.12e-05) |

For each problem with each parameter setting, three Tr-RMMEDA variants with different inner optimization methods (interior point, SQP and active set) get initial populations after each change. MIGD is the average of IGDs of these initial populations with 20 independent runs under 20 changes. The better values with smaller average that the method gets are shown with dark background.

problem with one parameter setting are regarded as an observation of the test. Therefore, there are 96 (12 problems and 8 parameters) observed data. Friedman detects significant differences in average value with a p-value of 1.7723e-6. The Nemenyi post-tests are shown in Figure 4.1(b).

Only results of C5 and C6 are presented due to space limitation. It has been observed that in most cases the Tr-DMOEA with the interior point method achieves the best results, compared with the two other Tr-DMOEA variants. The Friedman and Nemenyi statistical tests show that Tr-DMOEA with interior point significantly outperforms the other two variants.

#### 4.3.2.2 Solutions Quality Comparison after Optimization

Similarly, in the experiment of section 4.3.1, each Tr-DMOEA will get a solution set on one problem with one parameter setting, after optimization for $\tau_t$ generations. Each algorithm is independently run 20 times on each problem instance with each parameter setting. The obtained solution set is used to calculate the IGD value. Also, the IGD values of 20 changes are averaged to get the MIGD for each problem with one parameter. The mean and standard deviation of MIGD values of transferred solutions obtained by three Tr-DMOEAs in the last generation after change are shown in Table 4.6.

Friedman and Nemenyi statistical tests [90] were carried out across benchmark problems. The MIGD values that all algorithms get on one problem with one parameter setting are regarded as an observation of the test. Therefore, there are 96 (12 problems and 8 parameters) observed data. Friedman detects significant differences in average value with a p-value of 1.1162e-4. The Nemenyi post-tests are shown in Figure 4.1(c).

Only results of C5 and C6 are presented in Table 4.6. It has been observed that in most cases the Tr-DMOEA with SQP method achieves the best results, compared with two other Tr-DMOEA variants. The Friedman and Nemenyi statistical tests show that Tr-DMOEA with SQP significantly outperforms the other two variants.

After getting the computation time and solution quality comparison results of three Tr-DMOEA variants, it can be concluded that the interior point method in the original Tr-DMOEA is ineffective and extremely inefficient. Although interior point method can achieve the best transferred solutions among three compared methods, it consumes several or even more than ten times of what two other methods consume. In addition, transferred solutions found by interior point method become significantly worse than those found by SQP method after optimization, which shows that it is not beneficial for the interior point method to improve the efficiency of optimization, compared with SQP method.

Table 4.6: Mean and standard deviations of MIGD values of optimized solutions in the last generation after change obtained by three Tr-RMMEDA variants with different inner optimization algorithms.

| Prob. | C1 | | | C2 | | |
|---|---|---|---|---|---|---|
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 5.40e-02(2.18e-06) | 1.34e-01(1.91e-06) | 1.39e-01(2.16e-06) | 5.46e-02(6.04e-09) | 8.31e-02(1.68e-07) | 8.51e-02(2.18e-07) |
| FDA5 | 6.36e-01(1.02e-03) | 5.85e-01(5.84e-04) | 5.86e-01(5.56e-05) | 4.76e-01(1.80e-04) | 4.64e-01(2.92e-05) | 4.49e-01(5.14e-05) |
| $FDA5_{iso}$ | 3.52e-01(5.14e-05) | 3.25e-01(3.65e-05) | 3.29e-01(7.48e-06) | 2.63e-01(4.51e-06) | 2.64e-01(2.07e-05) | 2.68e-01(1.01e-05) |
| $FDA5_{dec}$ | 1.17e+00(3.47e-04) | 7.38e-01(1.36e-04) | 7.10e-01(6.78e-04) | 9.82e-01(1.28e-03) | 6.63e-01(6.10e-04) | 6.38e-01(2.29e-05) |
| DIMP2 | 8.71e+00(8.73e-03) | 9.30e+00(3.25e-03) | 9.33e+00(1.76e-03) | 7.15e+00(3.55e-03) | 7.72e+00(8.54e-03) | 8.09e+00(9.38e-03) |
| DMOP2 | 5.05e-01(7.24e-04) | 5.21e-01(1.47e-03) | 5.39e-01(1.04e-03) | 2.24e-01(1.23e-04) | 1.82e-01(3.29e-04) | 1.79e-01(2.47e-04) |
| $DMOP2_{iso}$ | 2.84e-03(2.95e-10) | 4.01e-03(7.77e-11) | 3.98e-03(4.60e-10) | 3.23e-03(5.81e-10) | 4.12e-03(4.22e-10) | 4.10e-03(9.83e-11) |
| $DMOP2_{dec}$ | 7.34e-01(2.61e-03) | 8.88e-01(4.42e-04) | 9.97e-01(2.18e-03) | 4.34e-01(2.89e-04) | 5.86e-01(3.66e-04) | 6.03e-01(3.32e-05) |
| DMOP3 | 2.66e-01(1.10e-04) | 3.44e-01(1.57e-03) | 2.83e-01(4.56e-05) | 1.27e-01(1.28e-04) | 9.17e-02(2.27e-05) | 1.00e-01(1.29e-05) |
| HE2 | 5.56e-01(3.26e-04) | 2.97e-01(1.95e-03) | 3.26e-01(2.74e-03) | 4.67e-01(4.78e-05) | 1.85e-01(1.78e-05) | 2.14e-01(2.63e-05) |
| HE7 | 1.71e-01(4.92e-06) | 1.66e-01(3.39e-07) | 1.68e-01(3.56e-06) | 1.28e-01(4.42e-06) | 1.26e-01(3.99e-06) | 1.21e-01(1.02e-06) |
| HE9 | 2.93e-01(8.54e-07) | 2.81e-01(2.13e-06) | 2.90e-01(2.04e-06) | 2.77e-01(1.19e-06) | 2.67e-01(2.07e-07) | 2.74e-01(9.62e-07) |
| Prob. | C3 | | | C4 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 6.02e-02(1.75e-08) | 6.76e-02(1.57e-08) | 7.09e-02(8.40e-08) | 6.53e-02(1.29e-08) | 6.73e-02(2.11e-08) | 7.02e-02(3.55e-08) |
| FDA5 | 2.80e-01(9.65e-06) | 3.37e-01(3.35e-06) | 3.46e-01(5.58e-05) | 1.83e-01(1.47e-05) | 2.16e-01(9.00e-06) | 2.22e-01(3.90e-05) |
| $FDA5_{iso}$ | 2.27e-01(2.52e-05) | 2.23e-01(3.51e-05) | 2.29e-01(2.15e-05) | 2.08e-01(5.54e-06) | 2.12e-01(3.28e-06) | 2.13e-01(9.72e-07) |
| $FDA5_{dec}$ | 7.03e-01(3.46e-04) | 6.39e-01(4.84e-05) | 6.25e-01(3.39e-05) | 5.07e-01(3.88e-05) | 5.61e-01(5.41e-05) | 5.50e-01(1.10e-05) |
| DIMP2 | 5.07e+00(1.25e-03) | 5.38e+00(1.15e-03) | 5.60e+00(9.32e-04) | 3.97e+00(4.63e-04) | 3.92e+00(2.80e-04) | 4.12e+00(1.57e-03) |
| DMOP2 | 4.25e-02(2.06e-05) | 7.41e-03(2.09e-08) | 1.04e-02(3.83e-07) | 9.01e-03(4.86e-06) | 3.28e-03(5.92e-11) | 3.31e-03(1.57e-10) |
| $DMOP2_{iso}$ | 3.89e-03(1.64e-10) | 4.29e-03(1.28e-10) | 4.29e-03(6.09e-11) | 4.25e-03(4.42e-10) | 4.35e-03(3.82e-10) | 4.35e-03(2.65e-10) |
| $DMOP2_{dec}$ | 1.42e-01(1.06e-05) | 1.95e-01(3.23e-05) | 2.46e-01(1.06e-04) | 6.38e-02(5.45e-06) | 9.01e-02(1.36e-05) | 1.05e-01(4.00e-05) |
| DMOP3 | 1.79e-02(9.52e-06) | 6.21e-03(4.65e-09) | 6.72e-03(5.04e-09) | 3.57e-03(1.92e-09) | 3.36e-03(2.68e-11) | 3.31e-03(1.15e-10) |
| HE2 | 2.82e-01(5.81e-05) | 1.22e-01(7.34e-06) | 1.28e-01(9.31e-06) | 1.42e-01(9.07e-06) | 6.70e-02(1.53e-06) | 6.43e-02(1.36e-06) |
| HE7 | 7.77e-02(2.43e-07) | 7.27e-02(1.34e-07) | 7.27e-02(2.55e-07) | 5.22e-02(4.36e-08) | 5.18e-02(3.53e-08) | 5.24e-02(6.09e-08) |
| HE9 | 2.55e-01(2.58e-07) | 2.50e-01(9.32e-08) | 2.56e-01(1.91e-07) | 2.45e-01(1.77e-07) | 2.43e-01(2.41e-08) | 2.47e-01(1.47e-07) |
| Prob. | C5 | | | C6 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 5.06e-02(8.71e-09) | 4.87e-02(1.80e-08) | 4.97e-02(2.80e-08) | 5.65e-02(8.32e-09) | 5.53e-02(1.22e-08) | 5.60e-02(9.64e-09) |
| FDA5 | 5.33e-01(1.27e-04) | 3.84e-01(8.14e-06) | 3.79e-01(2.42e-05) | 1.91e-01(3.94e-06) | 2.08e-01(1.19e-05) | 2.31e-01(8.89e-06) |
| $FDA5_{iso}$ | 2.22e-01(4.64e-06) | 1.83e-01(1.02e-05) | 2.05e-01(2.02e-05) | 1.61e-01(1.42e-06) | 1.51e-01(5.54e-06) | 1.55e-01(3.01e-06) |
| $FDA5_{dec}$ | 8.39e-01(1.48e-03) | 6.48e-01(5.58e-05) | 6.53e-01(3.82e-04) | 2.79e-01(7.93e-06) | 3.51e-01(2.10e-05) | 3.53e-01(6.43e-06) |
| DIMP2 | 7.28e+00(7.10e-04) | 7.93e+00(4.81e-03) | 8.34e+00(4.94e-03) | 4.21e+00(2.69e-04) | 4.11e+00(3.85e-03) | 4.29e+00(1.44e-04) |
| DMOP2 | 2.36e+01(1.07e-02) | 1.80e+01(1.24e-07) | 1.81e+01(2.31e-06) | 1.85e+01(1.15e-02) | 1.80e+01(7.61e-09) | 1.80e+01(1.48e-10) |
| $DMOP2_{iso}$ | 8.98e-02(1.90e-10) | 9.04e-02(1.94e-10) | 9.05e-02(9.52e-10) | 1.12e-01(8.90e-11) | 1.12e-01(3.06e-10) | 1.12e-01(1.22e-10) |
| $DMOP2_{dec}$ | 1.20e+00(4.20e-03) | 3.50e-01(8.52e-05) | 4.16e-01(1.15e-03) | 2.37e-01(4.19e-05) | 1.11e-01(3.23e-08) | 1.18e-01(3.16e-05) |
| DMOP3 | 2.32e+01(1.65e-02) | 1.80e+01(2.98e-07) | 1.80e+01(1.93e-06) | 1.86e+01(3.18e-03) | 1.80e+01(1.53e-11) | 1.80e+01(1.36e-08) |
| HE2 | 3.27e-01(1.26e-04) | 2.28e-01(2.91e-04) | 1.88e-01(2.47e-04) | 7.80e-02(3.22e-07) | 7.12e-02(2.55e-06) | 6.78e-02(2.48e-06) |
| HE7 | 1.12e-01(4.66e-07) | 1.11e-01(4.26e-07) | 1.09e-01(2.36e-07) | 4.82e-02(4.88e-08) | 4.65e-02(3.88e-08) | 4.66e-02(2.40e-08) |
| HE9 | 2.48e-01(3.91e-07) | 2.39e-01(2.06e-07) | 2.44e-01(3.44e-06) | 2.21e-01(5.59e-08) | 2.17e-01(6.19e-08) | 2.18e-01(1.94e-07) |
| Prob. | C7 | | | C8 | | |
| Methods | Interior | SQP | active | Interior | SQP | active |
| FDA4 | 5.51e-02(4.88e-08) | 8.18e-02(9.17e-07) | 8.43e-02(1.14e-06) | 6.55e-02(1.24e-08) | 6.74e-02(3.35e-08) | 7.05e-02(5.60e-08) |
| FDA5 | 4.35e-01(3.35e-04) | 4.48e-01(2.13e-05) | 4.39e-01(1.46e-05) | 1.72e-01(2.15e-06) | 2.13e-01(6.17e-06) | 2.28e-01(1.94e-05) |
| $FDA5_{iso}$ | 2.55e-01(4.51e-06) | 2.53e-01(8.10e-06) | 2.52e-01(1.18e-05) | 2.06e-01(3.14e-06) | 2.07e-01(3.68e-06) | 2.09e-01(6.68e-06) |
| $FDA5_{dec}$ | 8.62e-01(5.28e-04) | 6.18e-01(9.67e-04) | 6.03e-01(3.83e-04) | 4.92e-01(3.02e-05) | 5.24e-01(8.01e-05) | 5.21e-01(6.20e-05) |
| DIMP2 | 7.50e+00(4.41e-03) | 8.07e+00(4.01e-03) | 8.38e+00(8.77e-03) | 4.20e+00(7.72e-04) | 4.13e+00(2.30e-03) | 4.30e+00(2.06e-03) |
| DMOP2 | 2.08e-01(4.81e-05) | 1.59e-01(3.71e-05) | 1.73e-01(2.15e-05) | 7.75e-03(9.37e-06) | 3.28e-03(1.93e-11) | 3.34e-03(5.65e-11) |
| $DMOP2_{iso}$ | 3.16e-03(3.30e-10) | 4.11e-03(1.87e-10) | 4.10e-03(1.95e-10) | 4.23e-03(1.74e-10) | 4.37e-03(8.97e-11) | 4.36e-03(1.83e-10) |
| $DMOP2_{dec}$ | 4.22e-01(5.27e-04) | 5.52e-01(2.70e-04) | 5.74e-01(3.34e-04) | 7.30e-02(5.47e-06) | 8.16e-02(1.39e-05) | 9.11e-02(3.66e-05) |
| DMOP3 | 1.13e-01(8.69e-05) | 8.96e-02(9.48e-07) | 9.57e-02(2.16e-06) | 3.87e-03(6.97e-08) | 3.36e-03(2.28e-11) | 3.37e-03(3.47e-11) |
| HE2 | 4.51e-01(2.91e-05) | 1.92e-01(1.32e-05) | 2.18e-01(3.27e-05) | 1.40e-01(2.14e-06) | 6.51e-02(8.76e-07) | 6.64e-02(7.53e-02) |
| HE7 | 1.27e-01(2.44e-06) | 1.23e-01(1.12e-06) | 1.23e-01(6.50e-07) | 5.23e-02(4.98e-08) | 5.18e-02(3.87e-08) | 5.22e-02(3.27e-08) |
| HE9 | 2.77e-01(5.88e-07) | 2.66e-01(2.19e-07) | 2.71e-01(4.97e-07) | 2.44e-01(1.59e-06) | 2.42e-01(5.71e-07) | 2.45e-01(1.55e-08) |

For each problem with each parameter setting, three Tr-RMMEDA variants with different inner optimization methods (interior point, SQP and active set) get optimized populations by RMMEDA in the last generation of each change. MIGD is the average of IGDs of these populations with 20 independent runs under 20 changes. The better values with smaller average that the method gets are shown with dark background.

Considering the influence of $\tau_t$ on the quality of optimized solutions obtained by three Tr-DMOEAs, it is clear from Table 4.6 that when changes are medium (C1 - C4), the solution quality of that with SQP increases as $\tau_t$ increases from 5 to 10 and to 25. However, when $\tau_t = 50$, Tr-DMOEA with interior point method remains the best, the same as what these methods perform in the first generation after change. As for another two cases when changes are huge (C5 - C6) and small (C7 - C8), the Tr-DMOEA with interior point becomes worse while another two variants become better.

## 4.4 Summary

This Chapter studied how efficient transfer optimization is in DMO through time complexity and computation time analysis, showing that the interior point method in transfer optimization [25] is extremely time-consuming when solving the inner problem (equation (2.13) of Chapter 2). Considering the high computational cost of the transfer optimization, another experiment was conducted to verify whether the purpose of using transfer optimization in DMO vanishes, which leverages the computation time of transfer to optimize randomly generated solutions. The results show that randomly generated solutions after being optimized using transfer time are greatly better than transferred ones, which therefore encourages the proposal of more efficient transfer optimization algorithms for DMO. Even though the existing Tr-DMOEA is computationally consuming, it should be noted that it could still be a good alternative to solve problems with extremely expensive objective functions [98]. Besides, this chapter also computationally studied two other inner optimization methods to determine whether the efficiency of transfer optimization can be improved and whether the improved efficiency will affect the effectiveness. Experimental results shows that the SQP method achieved a better balance between the transfer optimization efficiency and effectiveness of optimized solutions.

Knowledge transfer has been widely used to solve DMOPs with changing position and/or shape of PSs and/or PFs and achieved much success when solving this type of DMOP, while few or even no studies have been conducted to solve other types of DMOPs, even though other types of DMOPs, such as DMOPs with a changing NObj widely exist in project scheduling [9–11], software development life cycle [99], etc. It is therefore significant to study how to apply knowledge transfer to solve other types of DMOPs, which will be studied in the next Chapter.

# Chapter 5

# Diversity Enhancing Knowledge Transfer via PS Expansion/Contraction in DMO with a Changing Number of Objectives (NObj) [1]

Even though numerous studies have been carried out to solve DMOPs with changing position and/or shape of PSs and/or PFs, few studies have been done to solve DMOPs with a changing number of objectives (NObj). The most recent work is the proposal of the Dynamic Two Archive Evolutionary Algorithm (DTAEA) [3]. The main idea was to simultaneously maintain two co-evolving populations, i.e, a convergence archive (CA) and a diversity archive (DA) during the evolution. Whenever environmental changes occur, CA and DA are reconstructed to preserve as much convergence and diversity as they can in the new environment.

Considering that the reconstruction of CA and DA in DTAEA involves copying (optimal) solutions from the past problem instance to the next after changes, DTAEA can be regarded as a kind of knowledge transfer-based algorithm, as it makes use of knowledge acquired from solving the previous problem instance to solve the new problem instance. However, in this Chapter, we show that DTAEA cannot handle DMOPs with a changing NObj containing more complex problem features including PF shapes (convex, discontinuous and mixed shape of convex and concave) and fitness landscapes (nonseparability and deceptiveness) well. Specifically, the knowledge transfer (i.e. CA and DA reconstruction) in DTAEA is incapable of providing enough diversity in these complex scenarios. The reason is that the change in the NObj changes the distribution of reconstructed CA on the true PF for

[1] This chapter is written based on content from the submitted paper as [*3] in Section 1.4.

the new problem instance and the uniformly sampled solutions in the search space by the DA reconstruction are not uniformly distributed in the objective space due to problem features in the more complex problems.

DMOPs with a changing NObj usually result in expansion or contraction of the PF or PS manifold. In order to expand and contract the PS, it is intuitive to obtain the expansion and contraction directions first for increasing and decreasing the NObj. Expansion directions are defined from one extreme point to its nondominated but far away solutions. Contractions directions are defined as the inverse direction from one extreme point to its closest nondominated solution and the directions between from any one nondominated solution to another one. Therefore, in this Chapter, we attempt to answer **Research question 3 (RQ3)** presented in Section 1.2: "How to heuristically determine PS expansion and contraction directions to expand and contract PSs when solving DMOPs with a changing NObj, so as to enhance diversity via knowledge transfer right after changes and further help the optimization itself?" To answer this research question, the following two sub-research questions will be answered:

1. How can we expand and contract the PS to increase diversity when solving DMOPs with a changing NObj, so as to improve knowledge transfer right after changes?
2. How does knowledge transfer via expansion and contraction of the PS help the optimization process itself? In reality of optimization, people are always interested in the solutions found after optimization, which are the ones that will be chosen to be deployed. Therefore, it is significant to figure out how the knowledge transfer used to initialize the population in the new environment helps in finding good solutions after optimization.

In order to answer these research questions, this Chapter develops a knowledge transfer algorithm for solving DMOPs with a changing NObj, motivated from the aforementioned limitations of the existing transfer approach DTAEA. The proposed knowledge transfer algorithm aims to enhance population diversity of the knowledge transfer via PS expansion and contraction after determining the expansion and contraction directions in a heuristic way. This strategy works better than DTAEA because DMOPs with a changing NObj usually result in the expansion or contraction of the dimension of the PS manifold [3].

Note that diversity is the appearance of a population in the objective space of a problem and an initial population can be influenced by knowledge transfer after an environmental change. In this chapter, we found that the population obtained by knowledge transfer in the existing work DTAEA lacks diversity. Then, we proposed a novel knowledge transfer algorithm KTDMOEA to enhance diversity of transferred solutions by expanding and contracting PSs. This is a transfer optimization method with the specific purpose of improving population diversity when the NObj increases.

Therefore, our proposed KTDMOEA contributes to the diversity improvement of the population. In addition, our proposed diversity enhancement method is able to benefit the knowledge transfer in DTAEA.

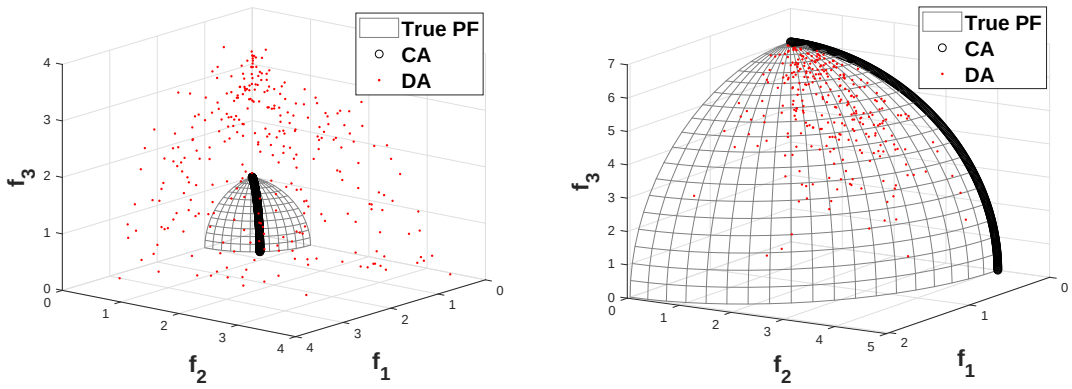
Also note that the distinction of KTDMOEA and existing transfer methods is that KTDMOEA makes use of the idea of transfer by heuristic methods, rather than using a specific transfer learning algorithm from machine learning to do the transfer, like transfer component analysis in Tr-DMOEA [25], manifold transfer learning in [26], TrAdaboost in [27, 81], etc. Specifically, the heuristic methods are heuristically searching for the expansion and contraction directions when increasing and decreasing the NObj, so as to conduct the PS expansion and contraction. In addition, the specific knowledge transfer method used in KTDMOEA is a solution-based transfer optimization algorithm. In particular, the method transfers the PS obtained in the previous environment by expanding and contracting the PS when increasing and decreasing the NObj.

The organization of this Chapter is described as follows. Section 5.1 investigates the state-of-the-art transfer algorithm (DTAEA) for solving DMOPs with a changing NObj and shows that it lacks sufficient diversity when the fitness landscape and Pareto front shape present nonseparability, deceptiveness or other challenging features. Then, a knowledge transfer dynamic multi-objective evolutionary algorithm (KTDMOEA) is proposed in Section 5.2 to enhance population diversity after changes by expanding/contracting the Pareto set in response to an increase/decrease in the NObj. This enables a solution set with good convergence and diversity to be obtained after optimization. Section 5.3 presents the detailed experimental setup for verifying the effectiveness of the proposed KTDMOEA. Then, experimental studies and analyses of the results are elaborated in Section 5.4. Finally, a summary is presented in Section 5.5.

## 5.1 A Preliminary Investigation of State-of-the-art

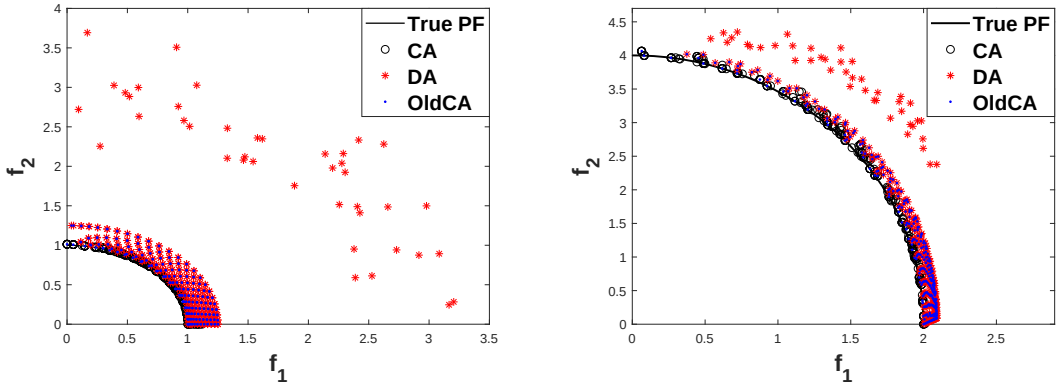
Even though DTAEA has been computationally demonstrated in [3] to be effective on DMOPs with a changing NObj based on knowledge transfer, the test problems that were used to evaluate DTAEA are somewhat limited, as problem features in those problems are relatively simple, such as linear or concave PF shape and fitness landscape with multimodality, bias or even nothing.

In order to evaluate whether DTAEA is able to solve DMOPs with a changing NObj and more complex problem features including PF shapes (convex, discontinuous and mixed shape of convex and concave) and fitness landscapes (nonseparability and deceptiveness), a benchmark problem WFG4 is arbitrarily selected from the WFG suite [100] as an example to conduct an experimental investigation of the per-




(a) F2 (b) WFG4

Figure 5.1: Distribution of the reconstructed CA and DA obtained by DTAEA in the first generation right after changes when increasing NObj from 2 to 3 on F2 and WFG4.




(a) F2 (b) WFG4

Figure 5.2: Distribution of the reconstructed CA and DA obtained by DTAEA in the first generation right after changes when decreasing NObj from 3 to 2 on F2 and WFG4.

formance of DTAEA. In contrast, F2 [3] is arbitrarily selected from the DTLZ suite [60]. When increasing the NObj, the problems are set as bi-objective problems and then given 1000 generations to evolve by DTAEA before increasing the NObj from 2 to 3. When decreasing the NObj, the problems are set as tri-objective problems and then given 1000 generations to evolve by DTAEA before decreasing the NObj from 3 to 2.

Figures 5.1 and 5.2 show the distribution of the old CA, the reconstructed CA and DA obtained by DTAEA on the two problems F2 and WFG4 in the first generation right after changes for the cases of increasing the NObj from 2 to 3 and decreasing the NObj from 3 to 2, respectively. It should be noted that when increasing the NObj, solutions in the old CA are all copied to the new CA. Therefore, in Figure 5.1, 'CA' represents both solutions in the old CA and the new CA. It is clear from Figure 5.1 that when increasing the NObj from 2 to 3, the new CA does not have good diversity on both F2 and WFG4. As for the reconstructed DA, it has a good

level of diversity on F2. As shown in Figure 5.1, solutions randomly generated in the search space are covering the whole area over the true PF. However, on WFG4, solutions in the reconstructed DA only cover a part of the PF over it.

Similarly, it can be seen from Figure 5.2 that solutions in DA also have good diversity on F2 when decreasing NObj from 3 to 2. However, on WFG4, there are some areas close to the high values of the second objective in the objective space without any solutions covered by the DA. It is clear that on both problems and the two cases (increasing and decreasing the NObj), the reconstructed CA and DA do not provide enough diversity. Therefore, the CA and DA reconstruction of DTAEA cannot provide enough diversity on DMOPs with more complex problem features. The reason is that the problem features in the more complex problems cause uniformly sampled solutions in the search space not to be uniformly distributed in the objective space.

## 5.2 Proposed Knowledge Transfer DMOEA (KTDMOEA)

In this section, we present our proposed knowledge transfer dynamic multi-objective evolutionary algorithm (denoted as KTDMOEA), which is designed to tackle more complex DMOPs with a changing NObj. The main component of KTDMOEA is the proposed diversity enhanced knowledge transfer, which is designed to improve population diversity right after changes in DMOPs with a changing NObj. KTDMOEA's flowchart is shown in Figure 5.3. As the flowchart exhibits, KTDMOEA maintains a single population. Whenever there is a change in the NObj, the process of knowledge transfer is evoked to reconstruct the population such that it has increased diversity; otherwise, other procedures in the evolution process are carried out. The novel process of knowledge transfer through PS expansion/contraction proposed in this paper will be elaborated in Section 5.2.1. Then, Section 5.2.2 presents the overall evolutionary process of the proposed KTDMOEA.

Note that the proposed KTDMOE does not require prediction of environmental changes, since it only relies on optimal solutions obtained in the previous one environment to do the transfer.

### 5.2.1 Diversity Enhancing Knowledge Transfer

As DMOPs with increasing/decreasing NObj result in the expansion/contraction of PS/PF, we propose to enhance diversity through PS expansion and contraction for increasing and decreasing NObj, respectively. This strategy is targeted at enhancing knowledge transfer right after changes. In this section, the specific details

of PS expansion and contraction are given in Section 5.2.1.1 and Section 5.2.1.2, respectively.

#### 5.2.1.1 Expand the PS when Increasing the NObj

Increasing the NObj usually results in the expansion of the dimension of PS/PF manifold. Therefore, the PS is proposed to be expanded when increasing the NObj, so as to increase the population diversity.

The idea of PS expansion in the decision space is illustrated in Figure 5.24(f). Note this figure is just drawn to demonstrate the process of PS expansion, the specific PF and expansion direction in real problems may be different. As shown in the figure, suppose the blue point is one extreme point in the PS before the change ($PS_t$); blue line is the Pareto optimal set at time step t with two NObj; the expansion direction is found by generating several solutions around the blue point and connecting the blue point to the point nondominated to it; the plane formed by 4 black lines is the Pareto optimal set at time step $t+1$ with three NObj and the red arrows are the expansion directions. Solutions evenly selected in $PS_t$, which are the points in the starting points of the red arrows, are regarded as the PS expansion base solutions to cover the whole PS right after the change ($PS_{t+1}$).

The framework of PS expansion is exhibited in Algorithm 5.1. As Algorithm 5.1 presents, in order to achieve PS expansion, the first step is to search for the potential PS expansion directions, whose procedure is given in Algorithm 5.2 and explained as follows. Given a set of Pareto optimal solutions at the time step t ($\mathbf{PS}_t$), the algorithm firstly finds solutions with the maximum objective value for each objective as the set of extreme points (denoted as $\mathbf{P}_e$) in line 1 of Algorithm 5.2. The use of extreme points helps to ensure that the found expansion directions are not misleading. If the middle points in the PF are selected, the found expansion directions might be along the direction of the PF, thus wasting computational resources. Then, in line 2, a random extreme point $\mathbf{x}_e$ is selected from the set $\mathbf{P}_e$ as the initial point of expansion directions. Later on, a solution set $\mathbf{P}_{var}$ with the same size as the population size is randomly produced around $\mathbf{x}_e$ via the polynomial mutation [101]

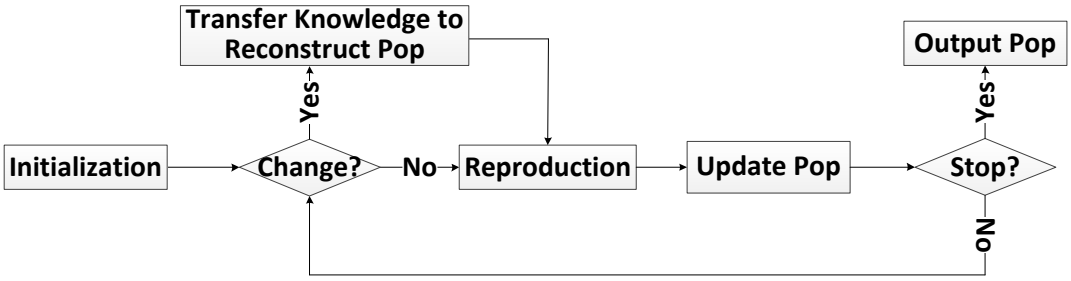
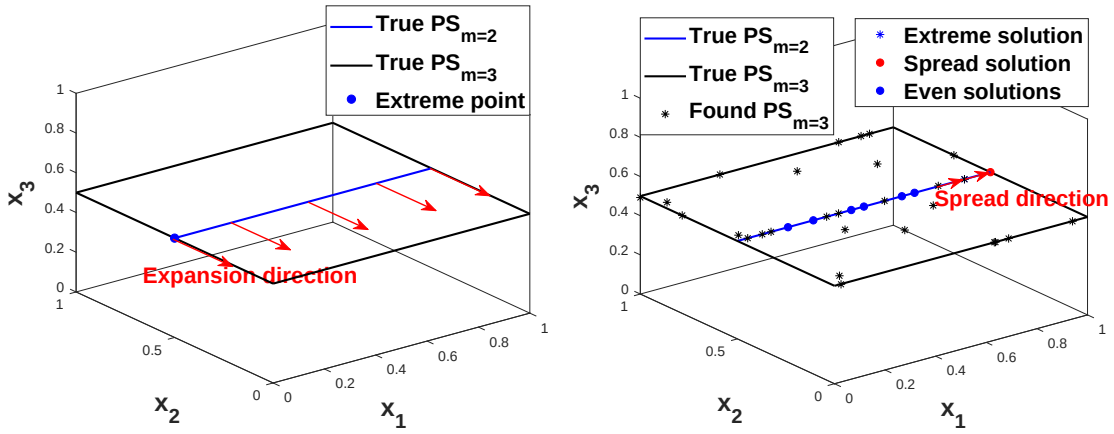



Figure 5.3: Flow chart of KTDMOEA.




(a) PS expansion (b) PS contraction

Figure 5.4: Brief illustration of how to expand and contract the PS for increasing and decreasing NObj, respectively.

**Algorithm 5.1: Expand the PS to Generate Transferred Solutions**

**Input:** Pareto optimal solution set at time t ($\mathbf{PS}_t$); set of the searched expansion directions $\mathbf{D}$ with size $N_{dir}$; population size $N$; the NObj for the old problem before the change $M_t$; number of solutions to generate along each expansion direction $\theta$;

**Output:** Transferred solutions $\mathbf{P}_{tr}$

1 Search a set of expansion direction as $\mathbf{D}$ via Algorithm 5.2;

2 Evenly select solutions from $\mathbf{PS}_t$ in the objective space with the size of $N_{base} = \lfloor \frac{N-M_t}{N_{dir}*\theta} \rfloor$ as $\mathbf{P}_{base}$;

3 Generate $\theta$ solutions along each direction in $\mathbf{D}$ to fill $\mathbf{P}_{tr}$ through the following equation (5.2);

4 Evenly select solutions from $\mathbf{PS}_t$ in the objective space with the size of $N - N_{base} * \theta * N_{dir}$ to fill $\mathbf{P}_{tr}$;

**Return** $\mathbf{P}_{tr}$.

to be regarded as the candidate sets of end points of expansion directions in line 3, which is also called detective population.

Procedures from lines 4 to 10 in Algorithm 5.2 are conducted to make sure the remaining solutions in $\mathbf{P}_{non}$ are nondominated and in different sub-spaces from the extreme point such that the extreme point and them can form the right expansion directions. Therefore, in line 4, all solutions in $\mathbf{P}_{var}$ are evaluated in the new environment and dominated solutions are discarded after sorting them using nondominated sorting [4]. Then, if all solutions in $\mathbf{P}_{var}$ are nondominated by those in $PS_t$, just set $\mathbf{P}_{non}$ as $\mathbf{P}_{var}$ ; else delete all solutions from $\mathbf{P}_{var}$ that are dominated by those in $\mathbf{PS}_t$ and regard the set of remaining solutions as $\mathbf{P}_{non}$ in line 8. Then, in line 9, use evenly generated weight vectors following the method in [3] to estimate density of $\mathbf{P}_{non}$ and $\mathbf{PS}_t$ with the method introduced in Section 5.2.2. Delete those solutions from $\mathbf{P}_{non}$ that are in the same subarea as those in $\mathbf{PS}_t$ in line 10. Later on, if there

**Algorithm 5.2: Search of Expansion Direction**

**Input:** Pareto optimal solution set at time t ($\mathbf{PS}_t$)
**Output:** Set of the searched expansion directions $\mathbf{D}$ or NULL.

1 Find the solutions with the maximum objective value for each objective as the set of extreme points ($\mathbf{P}_e$);
2 Randomly select one solution from $P_e$ and regard it as $\mathbf{x}_e$;
3 Produce a solution set $\mathbf{P}_{var}$ (also called detective population) with the same size as the population size through randomly generating solutions via the polynomial mutation [101] around the selected extreme point $x_e$;
4 Evaluate all solutions in $\mathbf{P}_{var}$ and delete dominated solutions after conducting nondominated sorting on them;
5 **if** *All solutions in* $\boldsymbol{P}_{var}$ *are nondominated by* $\boldsymbol{PS}_t$ **then**
6 Set $\mathbf{P}_{non}$ as $\mathbf{P}_{var}$;
7 **else**
8 Delete solutions from $\mathbf{P}_{var}$ that are dominated by those in the $\mathbf{PS}_t$ and set the remaining solutions as $\mathbf{P}_{non}$;
9 Use evenly generated weight vectors following the method in [3] to estimate density of $\mathbf{P}_{non}$ and $\mathbf{PS}_t$ with the method introduced in Section 5.2.2;
10 Delete solutions in $\mathbf{P}_{non}$ located at the same subarea with solutions of $\mathbf{PS}_t$ ;
11 **if** $\boldsymbol{P}_{non}$ *is NULL* **then**
**Return** NULL.
12 **else**
13 Use the remaining solutions in $\mathbf{P}_{non}$ and the extreme point $\mathbf{x}_e$ to form a set of lines that represent the directions (denoted as $\mathbf{D}$): $\mathbf{D}_j = \frac{\mathbf{P}^j_{non} - \mathbf{x}_e}{\|\mathbf{P}^j_{non} - \mathbf{x}_e\|}, (j = 1, ..., |\mathbf{P}_{non}|)$;
14 Delete duplicated directions from $\mathbf{D}$;
**Return D**.

is no solution in $\mathbf{P}_{non}$, this means no expansion direction is found and return NULL; else, in line 13, use the points to form a set of lines that represents the directions (denoted as $\mathbf{D}$) by regarding the extreme point $\mathbf{x}_e$ in line 2 as the starting point and those solutions in $\mathbf{P}_{non}$ as the end point: $\mathbf{D}_j = \frac{\mathbf{P}^j_{non} - \mathbf{x}_e}{\|\mathbf{P}^j_{non} - \mathbf{x}_e\|}, (j = 1, ..., |\mathbf{P}_{non}|)$. Then, delete duplicated expansion directions from $\mathbf{D}$ and return $\mathbf{D}$.

After getting the expansion directions, the next step is to expand the PS to generate transferred solutions following the expansion directions. The detailed procedures of this algorithm are shown in Algorithm 5.1. Given the Pareto optimal solution set at time t $\mathbf{PS}_t$, evenly select solutions from it with the size of $N_{base}$ as $\mathbf{P}_{base}$, where

$$N_{base} = \lfloor \frac{N - M_t}{N_{dir} * \theta} \rfloor, \tag{5.1}$$

where $N$ is the size of population; $M_{t+1}$ is the NObj at time step t+1; $N_{dir}$ is the number of expansion directions in $\mathbf{D}$ and $\theta$ is the number of solutions to generate

along each expansion direction, which is a parameter to be set by the user. $N - M_t$ is designed to enable those $M_t$ extreme points in $\mathbf{PS}_t$ to be preserved to the next environment. Then in line 3 of Algorithm 5.1, generate $\theta$ solutions along each expansion direction in $\mathbf{D}$ to fill the transferred solution set $\mathbf{P}_{tr}$ through the following Equation (5.2), which produces a transferred solution based on a base solution and an expansion direction.

$$\mathbf{x}_{new}^{(j-1)*N_{dir}+i} = \mathbf{x}_i + C_i^j * rand() * \mathbf{D}_j (i = 1, ..., N_{base}; j = 1, ..., N_{dir}), \tag{5.2}$$

where $\mathbf{x}_i$ is the $i$-th solution in the base population $\mathbf{P}_{base}$; $C_i^j$ is a float variable enabling some expanded solutions to reach the boundary of the decision space, whose detailed calculation will be discussed in the next paragraph; rand() is a function to generate a random number in $(0, 1]$; $\mathbf{D}_j$ is the $j$-th expansion direction in the set $\mathbf{D}$. After generating transferred solutions through PS expansion, if $\mathbf{P}_{tr}$ is not full, just evenly select solutions from $\mathbf{PS}_t$ in the objective space with the size of $N - N_{base} * \theta * N_{dir}$.

The calculation of $C_i^j$ should follow the criterion that all solutions expanded from the solutions $\mathbf{x}_i$ in $\mathbf{P}_{base}$ are within the bound of each decision variable and they should reach the boundary of the search space as close as possible. Bearing this criterion in mind, we design the calculation of $C_i^j$. Given a base solution $\mathbf{x}_i$ and one expansion direction $\mathbf{D}_j$, suppose $para^k$ is the value that makes the $k$-th variable of the generated solution reach the boundary of this variable. Therefore, each $para^k$ can be calculated according to whether the expansion direction is positive or negative, via the following equation:

$$para^k = \begin{cases} \frac{upper^k - x_i^k}{D_j^k}, & D_j^k > 0 \\ \frac{lower^k - x_i^k}{D_j^k}, & D_j^k < 0, \end{cases} \tag{5.3}$$

where $upper^k$ and $lower^k$ are the upper bound and lower bound of the $k$-th dimension of the decision space; $x_i^k$ is the $k$-th decision value; $D_j^k$ is the value of the direction $\mathbf{D}_j$ at the $k$-th dimension. In order to ensure each generated solution is located within the region, $C_i^j = \min_{k=1,...,n} para^k$, where $n$ is the dimension of the decision space.

Note that given any solution $\mathbf{x}$ and any direction $\mathbf{D}$, a new solution $\mathbf{x}_{new}$ can be generated within the bound of the search space using the above methods from equations (5.2)-(5.3). Then, we generalize the solution generation method in equation (5.2) as follows:

$$\mathbf{x}_{new} = \mathbf{x} + C * rand() * \mathbf{D}, \tag{5.4}$$

where $C$ is the step size whose calculation is presented in the previous paragraph where equation (5.3) locates and $rand()$ is a function to generate a random number in (0, 1].

#### 5.2.1.2 Contract the PS when Decreasing the NObj

It has been observed that decreasing the NObj usually results in the contraction of the dimension of PS/PF manifold. Therefore, the PS is proposed to be contracted when decreasing the number of objective.

The idea of PS contraction in the decision space is illustrated in Figure 5.4(b). Note this figure is just drawn to demonstrate the process of PS contraction, the specific cases in real problems may be different. It tries to generate spread and uniform solutions given current nondominated solution set. Steps from lines 4 to 7 in Algorithm 5.3 are designed to help improving the spread of the population and line 8 tries to make the distribution of population more even. As illustrated in Figure 5.4(b), black start points are the found optimal solutions for the problem with 3 objectives. When decreasing the NObj from 3 to 2, the black start points in the true PS of bi-DMOP (blue line) are selected as the nondominated solutions.

One solution (denoted by the red point) is generated based on the extreme point (denoted by blue start point) following the spread direction between the extreme point's closest point and itself, so as to increase the spread of population. Other solutions (denoted by the blue points) are produced from two randomly selected solutions in the nondominated solution set, to improve the even distribution of the population (suppose there is no bias in the problem).

**Algorithm 5.3: Contract the PS to generate transferred solutions**

**Input:** Pareto optimal solution set at time t ($\mathbf{PS}_t$);
**Output:** Transferred solutions $P_{tr}$

1 Evaluate the $\mathbf{PS}_t$ in the new environment and put the nondominated solutions to $\mathbf{P}_{non}$ after conducting the nondominated sorting on $\mathbf{PS}_t$;
2 Put all solutions in $\mathbf{P}_{non}$ to $\mathbf{P}_{tr}$;
3 Find the solutions with the maximum objective value at any objective from $\mathbf{P}_{non}$ as the set of extreme points ($\mathbf{P}_e$);
4 **for** $j = 1$ *to* $|\boldsymbol{P}_e|$ **do**
5 Find a closest solution $\mathbf{P}_{non}^j$ to $\mathbf{x}_e^j$ from $\mathbf{P}_{non}$ and connect $\mathbf{P}_{non}^j$ to $\mathbf{x}_e^j$ as a direction and normalize it as $\mathbf{D}_j = \frac{\mathbf{x}_e^j - \mathbf{P}_{non}^j}{|\mathbf{x}_e^j - \mathbf{P}_{non}^j|}$;
6 Produce a new solution $\mathbf{x}_{new}$ along the direction $\mathbf{D}_j$ to make it reach the boundary of the search space;
7 Put the solution $\mathbf{x}_{new}$ to $\mathbf{P}_{tr}$;
8 Random select two solutions ($\mathbf{x}_a$ and $\mathbf{x}_b$) from $\mathbf{P}_{tr}$ and generate one solution between them to fill $\mathbf{P}_{tr}$ until the size reaches the population size N;
**Return** $\mathbf{P}_{tr}$

Given the Pareto optimal solution set $\mathbf{PS}_t$, the first step in line 1 of Algorithm 5.3 is to evaluate all solutions in $\mathbf{PS}_t$ in the new environment and put the nondominated solutions to $\mathbf{P}_{non}$ after conducting nondominated sorting [4] on $\mathbf{PS}_t$. Then, put all

solutions of $\mathbf{P}_{non}$ to the set of transferred solution $\mathbf{P}_{tr}$ in line 2. Later on, in line 3, find solutions from $\mathbf{P}_{non}$ with the maximum objective value for each objective as the set of extreme points $\mathbf{P}_e$. Subsequently, for each extreme point $\mathbf{x}_e^j$ in the set $\mathbf{P}_e$, find a closest solution $\mathbf{P}_{non}^j$ to $\mathbf{x}_e^j$ from $\mathbf{P}_{non}$ and connect $\mathbf{P}_{non}^j$ to $\mathbf{x}_e^j$ as a direction and normalize it as $\mathbf{D}_j = \frac{\mathbf{x}_e^j - \mathbf{P}_{non}^j}{|\mathbf{x}_e^j - \mathbf{P}_{non}^j|}$, as shown in line 5. Then, produce a new solution $\mathbf{x}_{new}^i$ along $\mathbf{D}_i$ to make it reach the boundary of the search space according to the following equation:

$$\mathbf{x}_{new}^j = \mathbf{x}_e^j + C_j * \mathbf{D}_j, \tag{5.5}$$

where $C_j$ is a float variable, whose calculation method is the same as that in the process of PS expansion, as shown in Equation (5.3). The newly generated solution $\mathbf{x}_{new}$ is then put in $\mathbf{P}_{tr}$. Later on, in line 8, randomly select two solutions ($\mathbf{x}_a$ and $x_b$) from $P_{tr}$ and generate one solution between them through equation (5.6) to fill $P_{tr}$ until the size of $P_{tr}$ reaches the population size $N$, using the following equation:

$$\mathbf{x}_{new} = \mathbf{x}_a + rand() * (\mathbf{x}_a - \mathbf{x}_b), \tag{5.6}$$

We believe this strategy of expanding/contracting the PF and PS works better over DTAEA [3] because increasing or decreasing the NObj usually results in the expansion and contraction of the dimension of the PS manifold. When increasing the NObj, the proposed PS expansion is able to find the expansion directions and generate solutions along these directions, increasing the population diversity in the new environment. When decreasing the NObj, the two mechanisms in PS contraction are targeted for improving the spread and evenness of the distribution. Therefore, those produced solutions by PS expansion and contraction may achieve better diversity than that of DTAEA.

### 5.2.2 KTDMOEA

As the PS expansion/contraction is designed to enhance the diversity of population after changes, the proposed KTDMOEA will not maintain a separate DA. As a result, there is no need to update DA as in DTAEA. The population update mechanism in KTDMOEA is the same as the update mechanism of CA in [3].

The flowchart of KTDMOEA is shown in Figure 5.3. The overall framework of the proposed KTDMOEA is given in Algorithm 5.4. KTDMOEA starts with generating an initial population of size $N$, as shown in line 1. While the stopping criteria are not satisfied, carry out the following steps. Detect whether the environmental changes occur. If the NObj is detected to increase, evoke the process of PS expansion on *Pop* using Algorithm 5.1. If the NObj decreases, evoke the process of PS contraction on *Pop* using Algorithm 5.3. If there is no change detected, conduct

**Algorithm 5.4: Framework of KTDMOEA**

**Input:** Population size $N$;
**Output:** The found population $Pop$

```
1  Randomly generate an initial population Pop;
2  while stopping criteria not satisfied do
3      if the number of objective changes then
4          if the number of objective increases then
5              Conduct the process of PS expansion in Algorithm 5.1 ;
6          else if the number of objective decreases then
7              Conduct the process of PS contraction in Algorithm 5.3;
8      else
9          Generate an offspring population A using the mating selection and
             genetic operators based on the parent population Pop;
10         Update Pop using offspring population A with the CA update
             mechanism in DTAEA [3] ;
   Return Pop
```

the evolutionary optimization process on $Pop$. In line 9, an offspring population $A$ is produced through the following two steps until the size of $A$ reaches the population size $N$. Firstly, randomly pick two solutions from $Pop$ as the parent solutions via the mating selection. Then, those two solutions are used to generate two offspring solutions via appropriate crossover and mutation operators. Here, we utilize the simulated binary crossover [101] and the polynomial mutation, as used by most continuous multi-objective evolutionary algorithms including DTAEA.

Lastly, the generated offspring population $A$ is used to update $Pop$ with the CA update mechanism in DTAEA [3]. Due to space limitation, the CA update mechanism of DTAEA is not introduced here. Interested readers can refer to [3].

### 5.2.3 Time Complexity Analysis

The time complexity of KTDMOEA is discussed in this subsection for one generation. When increasing or decreasing NObj, the PS expansion or contraction is evoked. In the case of increasing NObj, the main complexity of PS expansion is cost in nondominated sorting (line 9 of Algorithm 2) and density estimation (line 9 of Algorithm 2) of searching for expansion directions. The former consumes $O(N^2)$ and the latter costs $O(N|W(t)|)$, where $W(t)$ is the number of weight vectors at time step $t$. As for the PS contraction, the nondominated sorting in line 1 of Algorithm 3 costs $O(N^2)$; the for loop consumes $O(M(t)N)$, where $M(t)$ is the number of objectives at time step $t$. As KTDMOEA utilizes CA update mechanism in DTAEA to update the population, the population update in KTDMOEA costs $O(N^2)$ comparisons. In

summary, assuming that $N > W(t)$ and $N > M(t)$, complexity of KTDMOEA in one generation is $O(N^2)$.

## 5.3 Experimental Setup

In this section, experimental studies are designed to verify whether the improved knowledge transfer-based approach answers the research questions mentioned at the beginning of this Chapter. Analyses will be carried out to reveal whether existing DMOEAs for DMOPs are able to deal with a changing NObj despite not being designed to do so, and how well static MOEAs could perform on DMOPs with a changing NObj. These are important baselines.

### 5.3.1 Benchmark Problems

Two suites of multi-objective optimization test problems DTLZ [60] and WFG [100] are modified to be DMOPs with a changing NObj. Four DMOPs with a changing NObj from DTLZ1-DTLZ4 are renamed as F1-F4, the same as in [3]. These two suites of benchmark functions are used to verify that the proposed algorithm is able to deal with problems with both simple and complex problem features.

- F1-F4 in the paper of DTAEA [3]: those four problems are modified from DTLZ1 to DTLZ4 [60] just by making the NObj change over time without modifying other problems features. Those problems have simple problem features including linear and concave PF shape, and simple fitness landscape like multi-modal and bias.

- WFG1-WFG9 [100]: these functions are made that their number of objective change over time and similarly without modifying other problems features. These benchmark has not been used in other papers related to DMOPs with a changing NObj. Those problems have more complex problem features including PF shape (linear, convex, concave, degenerated and mixed shape of convex and concave ), as well as complicated fitness landscape (multimodality, bias, nonseparability, and deceptiveness).

Problem features of all of them are summarized in Table 5.1.

There are two different sequences of changes for these benchmark problems:

Table 5.1: Problem features of F1-F4 and WFG1-WFG9.

| Problems | PF shape | Fitness landscape |
|---|---|---|
| F1 | linear | multimodality |
| F2 | concave | — |
| F3 | concave | multimodality |
| F4 | concave | bias |
| WFG1 | mixed | bias |
| WFG2 | convex, disconnected | nonseparability |
| WFG3 | linear, degenerate | nonseparability |
| WFG4 | concave | multimodality |
| WFG5 | concave | deceptiveness |
| WFG6 | concave | nonseparability |
| WFG7 | concave | bias |
| WFG8 | concave | bias, nonseparability |
| WFG9 | concave | bias, multimodality, deceptiveness, nonseparability |

1. The initial NObj is set as 2. It firstly increases from 2 to 7 one by one and then decreases from 7 to 2 one by one (simply denoted as '2-7-2 one by one'), which was used in [3];

$$m(t) = \begin{cases} 2, & t{=}0 \\ 3, & t{=}1 \\ m(t-1)+1, & t \in [2{,}5] \\ m(t-1)-1, & t \in [6{,}10]. \end{cases} \tag{5.7}$$

2. The initial NObj is set as 7. It firstly decreases from 7 to 2 one by one and then increases from 2 to 7 one by one (simply denoted as '7-2-7 one by one').

$$m(t) = \begin{cases} 7, & t{=}0 \\ 6, & t{=}1 \\ m(t-1)-1, & t \in [2{,}5] \\ m(t-1)+1, & t \in [6{,}10]. \end{cases} \tag{5.8}$$

$t$ is the identifier of the change. Both sequences of changes have 10 environmental changes.

### 5.3.2 Compared Algorithms

In our experimental studies, five algorithms are selected for the comparison, so as to verify the performance of our proposal against the state-of-the-art. Considering the popularity and good performance on solving static MOPs, the elitist nondominated

sorting genetic algorithm (NSGA-II) [4] and multi-objective evolutionary algorithm based on decomposition (MOEA/D) [57] are chosen to verify whether they are able to tackle DMOPs with a changing NObj. For NSGA-II and MOEA/D, whenever there is a change, the whole population of the last generation in the old environment is just copied to the next generation after changes and then re-evaluated in the new environment to respond to changes in the NObj. Besides, two popular and state-of-the-art DMOEAs including DNSGA-II [102] and MOEA/D-KF [43] specifically designed for DMOPs with a changing shapes and/or positions of PS and/or PF are selected to be compared, so as to figure out whether DMOEAs for solving DMOPs with fixed NObj are able to deal with DMOPs with a changing NObj. These four algorithms are compared to verify whether it is necessary to develop extra approaches tailored for DMOPs with a changing NObj. In order to verify whether the improved knowledge transfer-based approach (KTDMOEA) answers the research questions mentioned at the beginning of this Chapter, as one of the popular and recently developed algorithms targeted for handling changes in the NObj, DTAEA [3] is also chosen to be compared.

The detailed descriptions of these algorithms are presented as follows:

- Static NSGA-II [4] the elitist nondominated sorting genetic algorithm; the whole old population is just copied to the next generation after changes in the NObj and then re-evaluated in the new environment.
- Static MOEA/D [57] a multi-objective evolutionary algorithm based on decomposition; the whole old population is just copied to the next generation after changes in the NObj and then re-evaluated in the new environment.
- DNSGA-II [7] the dynamic version of NSGA-II for DMOPs with changing shape and/or position of PS and/or and fixed NObj; after changes, it tackles the DMOPs with a changing NObj just like it deals with DMOPs where shape and/or position of PS and/or changes. More specifically, half of the population is randomly generated with others remained from the previous optimized population.
- MOEA/D-KF [43]: MOEA/D with Kalman Filter prediction for DMOPs with changing shape and/or position of PS and/or and fixed NObj; after changes, it tackles the DMOPs with a changing NObj just like it deals with DMOPs where shape and/or position of PS and/or changes.
- DTAEA [3]: dynamic two archive evolutionary algorithm, which is the most recently work designed for DMOPs with a changing NObj. The main idea is to simultaneously maintain two co-evolving populations, i.e. convergence archive (CA) and diversity archive (DA), which are designed to focus on population convergence and diversity respectively, so as to complement with each other in the optimization process through mating selection and population update. Besides, whenever environmental changes, CA and DA are reconstructed to

preserve as much convergence and diversity as they can in the new environment. More specifically, when increasing the NObj, solutions in the old CA are all copied to the new CA. On the contrary, when decreasing the NObj, nondominated and dominated solutions of the old CA are all copied to the new CA and new DA, respectively.

### 5.3.3 Parameter Settings

The parameters of these compared algorithms are set as follows:

- Population size: 300, the same as that of DTAEA, $\theta$ in KTDMOEA is set as 2. The impact of $\theta$ on KTDMOEA's performance will be analyzed in Section 5.4.5;
- Several different frequencies of change: $\tau_t$ is set as 5, 25 and 50 and 200; Those parameters are set for assessing the effects of different algorithms under different frequencies of change. $\tau_t$ is set as 200 larger than the setting of DMOPs with changing position/shape of PSs/PFs, since most algorithms may require more generations to perform well on problems with more than 4 objectives [62];
- All algorithms run 31 times independently, also the same as in DTAEA's work [3];
- 1000 generations are given to each algorithm before the first change so that the population before the change can converge;
- The crossover probability was $p_c = 1.0$ and its distribution index was $\eta_c = 20$. The mutation probability was $p_m = 1/\text{n}$ (where n denotes the number of decision variables) and its distribution $\eta_m = 20$. These parameters are chosen because of their good performance on solving continuous problems, which have been analyzed in [103] and [60].
- The neighbourhood size and the number $n_r$ of solutions allowed to replace in MOEA/D were set to 20 and 2, respectively, which is the same as in the original paper [57].

### 5.3.4 Performance Metrics

- Mean Hypervolume (MHV) [3] is the mean Hypervolume (HV) [104] of obtained solution sets by an algorithm at a set of discrete time steps. HV comprehensively measures the convergence and diversity of solution sets; the larger the better.

- Mean Generational Distance (MGD) [38] [15] is the mean Generational Distance (GD) [105] of obtained solution sets by an algorithm at a set of discrete time steps. GD evaluates the convergence of obtained solution sets; the smaller the better.

- Mean Maximum Spread (MMS) is the mean Maximum Spread (MS) [105] of obtained solution sets by an algorithm at a set of discrete time steps. MS assesses the diversity of solution sets; the larger the better.

Note that these three metrics are used to measure the solution quality of a found solution set. They can be also used to measure comprehensive performance of an algorithm by averaging the metric values of all obtained solutions under multiple environmental changes.

## 5.4 Experimental Results and Analyses

In order to achieve the objectives of the experiment, i.e. answering the research questions and verifying whether the existing static MOEAs and DMOEAs for solving DMOPs with fixed NObj are able to tackle DMOPs with a changing NObj, experimental results of all compared algorithms are presented in this section. Furthermore, further analyses regarding the further verification of the improved knowledge transfer, performance comparison of different NObj changing sequence and the impact of algorithm parameters are also given in this section.

Three metrics (MHV, MGD and MMS) are used to measure the quality of the found solutions at the first generation after the change and in the last generation before the next change by six algorithms. To show the significant superiority of the proposed KTDMOEA to other algorithms across all problem instances in general, Friedman and Nemenyi statistical tests [90] are adopted across all benchmark problems regarding the three metrics (MHV, MGD and MMS) of six compared algorithms. The larger the values of MHV and MMS, the better the algorithm. Therefore, the larger the Friedman ranking, the better the algorithm. Similarly, the smaller the Friedman ranking of MGD, the better the algorithm.

The mean metric value of 31 independent runs that each algorithm gets on one problem with one frequency of change at each environmental change is regarded as an observation of the Friedman and Nemenyi test. Therefore, there are 520 (13 problems, 4 different frequencies of change and 10 environmental changes) observations for each algorithm in the Friedman and Nemenyi tests. Additionally, in order to show the significant superiority of the proposed KTDMOEA to other algorithms on each individual problem of each parameter setting, the Wilcoxon Rank Sum test at the 5% significance level is implemented on each benchmark problem regarding

each metric of six compared algorithms at each parameter setting. Therefore, there are 31 observations obtained from 31 independent runs for each algorithm on each problem and parameter setting in the Wilcoxon Rank Sum test.

### 5.4.1 Initial Effectiveness of Knowledge Transfer

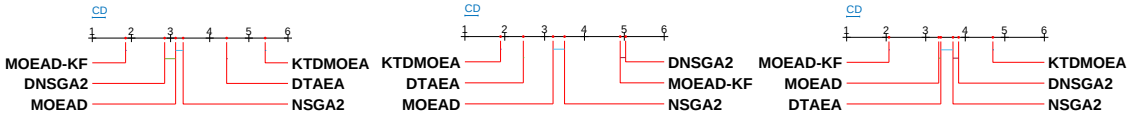
In order to verify (1) whether the proposed PS expansion/contraction mechanism is able to increase diversity so as to improve knowledge transfer after the change, and (2) whether the MOEAs not tailored for DMOPs with a changing NObj can achieve this aim after the change, the quality of the solutions obtained by all algorithms in the first generation after changes is compared.

#### 5.4.1.1 NObj Increasing from 2 to 7 and then Decreasing from 7 to 2

Figure 5.5 presents the Nemenyi post-tests results among MHV, MGD and MMS of obtained solutions at the first generation after changes by 6 algorithms. Friedman test detects significant differences in average values for MHV, MGD and MMS with a p-value of 3.57E-251, 9.14E-256, and 1.44E-117, respectively. Tables 5.2-5.4 show mean and standard deviation values of MHV, MGD and MMS of obtained solutions in the first generation after changes averaged across 10 environmental changes in the sequence of changes as '2-7-2 one by one'.

Overall, it can be observed from Figure 5.5 that when comparing all algorithms, KTDMOEA significantly outperforms all others in all three metrics. This implies that the proposed knowledge transfer technique via PS expansion/contraction indeed improves the diversity and maintain the convergence of transferred solutions right after changes, under all frequencies of changes on most problems.

Specifically, it is clear from Table 5.2 that KTDMOEA performs the best regarding MHV on all test problems except for F4 and WFG9. As for F4, it can also be seen from Tables 5.2 and 5.4 that KTDMOEA has significantly worse MHV value due to its worse MMS value, since KTDMOEA gets best MGD value on F4, as can be seen from Table 5.3. Regarding WFG9, it can be found from Tables 5.2-5.4 that




(a) MHV (b) MGD (c) MMS

Figure 5.5: Friedman ranking among MHV, MGD and MMS of obtained solutions at the first generation by 6 algorithms in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

Table 5.2: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 8.90E-01(3.40E-05)† | 8.66E-01(3.20E-04)† | 8.84E-01(1.15E-04)† | 2.17E-02(7.99E-04)† | 9.07E-01(4.51E-04)† | 9.96E-01(7.51E-11) |
| | 25 | 6.40E-01(1.51E-03)† | 8.81E-01(9.54E-05)† | 6.09E-01(7.70E-04)† | 4.70E-03(1.34E-04)† | 8.67E-01(2.83E-03)† | 9.96E-01(1.43E-09) |
| | 50 | 5.48E-01(2.65E-03)† | 8.86E-01(3.85E-05)† | 4.99E-01(2.37E-04)† | 3.72E-03(4.27E-05)† | 8.52E-01(2.36E-03)† | 9.96E-01(1.40E-10) |
| | 200 | 4.25E-01(5.63E-03)† | 9.12E-01(3.52E-05)† | 3.29E-01(1.33E-03)† | 1.25E-02(2.53E-04)† | 8.62E-01(1.15E-03)† | 9.96E-01(1.31E-10) |
| F2 | 5 | 8.54E-01(6.18E-05)† | 8.96E-01(1.55E-05)† | 8.85E-01(1.19E-05)† | 4.18E-01(6.51E-03)† | 9.17E-01(2.41E-05)† | 9.56E-01(1.63E-06) |
| | 25 | 8.98E-01(1.04E-05)† | 9.17E-01(2.87E-06)† | 8.79E-01(2.81E-05)† | 2.96E-01(4.56E-04)† | 9.26E-01(6.18E-06)† | 9.55E-01(3.23E-07) |
| | 50 | 8.89E-01(4.39E-05)† | 9.20E-01(8.29E-07)† | 8.73E-01(1.21E-05)† | 3.05E-01(7.32E-04)† | 9.28E-01(6.60E-06)† | 9.56E-01(4.57E-07) |
| | 200 | 8.80E-01(8.10E-05)† | 9.23E-01(5.57E-08)† | 8.71E-01(1.28E-05)† | 4.05E-01(1.36E-03)† | 9.29E-01(3.78E-06)† | 9.56E-01(2.94E-07) |
| F3 | 5 | 7.33E-01(1.87E-04)† | 7.20E-01(3.55E-04)† | 6.87E-01(2.83E-04)† | 4.18E-01(6.51E-03)† | 7.69E-01(6.70E-04)† | 9.57E-01(1.88E-07) |
| | 25 | 5.69E-01(4.92E-04)† | 7.28E-01(3.84E-04)† | 5.36E-01(4.44E-04)† | 0.00E+00(0.00E+00)† | 7.34E-01(4.03E-03)† | 9.54E-01(1.64E-06) |
| | 50 | 4.79E-01(7.04E-04)† | 7.36E-01(4.48E-04)† | 4.55E-01(3.63E-04)† | 0.00E+00(0.00E+00)† | 7.04E-01(4.62E-03)† | 9.57E-01(1.78E-06) |
| | 200 | 3.55E-01(1.25E-03)† | 7.94E-01(2.35E-04)† | 2.97E-01(1.55E-03)† | 0.00E+00(0.00E+00)† | 6.60E-01(4.27E-03)† | 9.57E-01(3.18E-06) |
| F4 | 5 | 8.41E-01(1.23E-04)† | 8.45E-01(2.30E-04)† | 8.60E-01(1.54E-04)† | 8.57E-01(1.01E-04)† | 9.14E-01(6.27E-05)‡ | 8.68E-01(1.52E-03) |
| | 25 | 8.29E-01(3.06E-04)† | 8.99E-01(1.06E-04)‡ | 8.22E-01(1.14E-04)† | 9.06E-01(1.50E-05)‡ | 9.28E-01(9.70E-06)‡ | 8.81E-01(4.75E-04) |
| | 50 | 7.76E-01(2.64E-04)† | 9.12E-01(4.23E-05)‡ | 8.22E-01(6.96E-05)† | 9.11E-01(3.20E-05)‡ | 9.29E-01(1.18E-05)‡ | 8.78E-01(8.21E-04) |
| | 200 | 7.80E-01(4.40E-05)† | 9.23E-01(3.27E-07)‡ | 8.16E-01(8.54E-05)† | 9.21E-01(2.05E-05)‡ | 9.27E-01(5.15E-06)‡ | 8.81E-01(1.28E-03) |
| WFG1 | 5 | 9.14E-01(3.94E-05)† | 8.40E-01(5.88E-05)† | 8.83E-01(8.36E-05)† | 7.52E-01(9.59E-04)† | 9.09E-01(3.66E-05)† | 9.66E-01(2.39E-06) |
| | 25 | 9.27E-01(1.46E-05)† | 8.62E-01(5.10E-05)† | 9.11E-01(3.04E-05)† | 7.92E-01(5.23E-04)† | 9.27E-01(1.88E-05)† | 9.67E-01(1.54E-06) |
| | 50 | 9.28E-01(4.78E-06)† | 8.74E-01(3.56E-05)† | 9.15E-01(2.56E-05)† | 7.90E-01(6.23E-04)† | 9.27E-01(8.86E-06)† | 9.67E-01(1.34E-06) |
| | 200 | 9.26E-01(2.70E-06)† | 8.92E-01(1.15E-05)† | 9.18E-01(6.35E-06)† | 7.65E-01(1.13E-03)† | 9.26E-01(2.34E-06)† | 9.67E-01(5.27E-07) |
| WFG2 | 5 | 8.87E-01(2.44E-04)† | 8.37E-01(2.05E-04)† | 8.89E-01(1.50E-04)† | 2.39E-01(3.31E-07)† | 9.11E-01(2.23E-04)† | 9.61E-01(2.40E-06) |
| | 25 | 9.04E-01(1.41E-04)† | 8.65E-01(1.84E-04)† | 8.91E-01(1.50E-04)† | 2.39E-01(2.02E-07)† | 9.21E-01(1.76E-04)† | 9.61E-01(3.72E-06) |
| | 50 | 8.98E-01(1.99E-04)† | 8.73E-01(1.32E-04)† | 8.84E-01(1.68E-04)† | 2.39E-01(9.53E-08)† | 9.17E-01(1.51E-04)† | 9.61E-01(2.67E-06) |
| | 200 | 8.85E-01(1.56E-04)† | 8.83E-01(1.48E-04)† | 8.74E-01(2.42E-04)† | 2.39E-01(5.31E-08)† | 9.10E-01(3.15E-04)† | 9.60E-01(1.10E-05) |
| WFG3 | 5 | 6.71E-01(1.10E-04)† | 6.15E-01(9.37E-05)† | 6.48E-01(5.07E-05)† | 2.50E-01(3.15E-07)† | 6.76E-01(1.57E-04)† | 6.98E-01(6.26E-06) |
| | 25 | 6.57E-01(1.30E-04)† | 6.42E-01(6.04E-05)† | 6.49E-01(4.06E-05)† | 2.51E-01(4.83E-07)† | 6.64E-01(1.14E-04)† | 6.97E-01(1.28E-05) |
| | 50 | 6.56E-01(5.06E-05)† | 6.45E-01(5.86E-05)† | 6.48E-01(4.18E-05)† | 2.52E-01(3.02E-07)† | 6.58E-01(1.29E-04)† | 6.99E-01(1.19E-05) |
| | 200 | 6.53E-01(6.29E-05)† | 6.47E-01(6.73E-05)† | 6.43E-01(4.76E-05)† | 2.52E-01(2.61E-07)† | 6.50E-01(1.14E-04)† | 6.99E-01(7.52E-06) |
| WFG4 | 5 | 6.01E-01(2.65E-04)† | 5.70E-01(1.33E-04)† | 6.08E-01(5.39E-05)† | 2.75E-01(1.37E-04)† | 6.93E-01(3.09E-05)† | 8.07E-01(2.36E-06) |
| | 25 | 6.76E-01(1.70E-05)† | 6.15E-01(7.13E-05)† | 6.36E-01(3.20E-05)† | 3.03E-01(3.30E-05)† | 7.52E-01(6.12E-07)† | 8.07E-01(6.30E-06) |
| | 50 | 6.78E-01(2.65E-05)† | 6.35E-01(5.59E-05)† | 6.39E-01(3.05E-05)† | 3.00E-01(1.17E-05)† | 7.54E-01(2.74E-07)† | 8.16E-01(6.57E-07) |
| | 200 | 6.85E-01(2.38E-05)† | 6.71E-01(2.18E-05)† | 6.46E-01(1.59E-05)† | 3.00E-01(3.60E-06)† | 7.53E-01(3.63E-07)† | 8.17E-01(4.61E-07) |
| WFG5 | 5 | 5.37E-01(2.49E-04)† | 5.67E-01(6.48E-05)† | 5.98E-01(5.12E-05)† | 5.03E-01(1.42E-05)† | 6.70E-01(1.40E-05)† | 7.47E-01(1.06E-05) |
| | 25 | 6.36E-01(1.76E-05)† | 6.11E-01(2.89E-05)† | 6.17E-01(1.92E-05)† | 5.15E-01(4.43E-06)† | 7.05E-01(2.78E-06)† | 7.61E-01(1.07E-05) |
| | 50 | 6.42E-01(8.92E-06)† | 6.21E-01(3.16E-05)† | 6.22E-01(1.55E-05)† | 5.13E-01(6.01E-06)† | 7.11E-01(1.88E-06)† | 7.70E-01(4.35E-06) |
| | 200 | 6.46E-01(9.84E-06)† | 6.45E-01(7.12E-06)† | 6.28E-01(1.56E-05)† | 5.15E-01(3.70E-06)† | 7.17E-01(1.55E-06)† | 7.73E-01(1.33E-06) |
| WFG6 | 5 | 4.30E-01(1.42E-03)† | 4.34E-01(7.61E-04)† | 5.00E-01(1.22E-03)† | 1.41E-01(1.85E-06)† | 5.13E-01(2.43E-03)† | 7.31E-01(6.35E-04) |
| | 25 | 5.55E-01(9.80E-04)† | 5.21E-01(7.29E-04)† | 5.52E-01(9.44E-04)† | 1.43E-01(7.40E-06)† | 5.53E-01(2.84E-03)† | 7.41E-01(4.51E-04) |
| | 50 | 5.77E-01(1.27E-03)† | 5.43E-01(8.76E-04)† | 5.55E-01(1.23E-03)† | 1.43E-01(4.11E-06)† | 5.35E-01(4.33E-03)† | 7.42E-01(6.05E-04) |
| | 200 | 6.01E-01(1.13E-03)† | 5.80E-01(1.04E-03)† | 5.66E-01(9.38E-04)† | 1.44E-01(4.32E-06)† | 5.61E-01(3.06E-03)† | 7.52E-01(4.17E-04) |
| WFG7 | 5 | 6.65E-01(3.91E-05)† | 4.85E-01(2.60E-04)† | 6.17E-01(5.56E-05)† | 5.06E-01(1.35E-04)† | 7.07E-01(3.21E-05)† | 8.15E-01(1.28E-06) |
| | 25 | 6.73E-01(1.98E-05)† | 5.87E-01(4.17E-05)† | 6.28E-01(2.25E-05)† | 6.13E-01(4.76E-05)† | 7.37E-01(1.70E-06)† | 8.18E-01(1.10E-06) |
| | 50 | 6.75E-01(1.55E-05)† | 6.16E-01(2.09E-05)† | 6.30E-01(1.60E-05)† | 6.34E-01(1.70E-05)† | 7.36E-01(2.01E-06)† | 8.21E-01(4.43E-07) |
| | 200 | 6.77E-01(1.20E-05)† | 6.64E-01(2.49E-05)† | 6.32E-01(8.83E-06)† | 6.55E-01(2.50E-05)† | 7.34E-01(7.61E-07)† | 8.22E-01(5.11E-07) |
| WFG8 | 5 | 6.66E-01(5.10E-05)† | 4.76E-01(3.46E-04)† | 6.18E-01(4.49E-05)† | 5.04E-01(1.43E-04)† | 7.09E-01(6.02E-05)† | 8.16E-01(2.29E-06) |
| | 25 | 6.74E-01(1.75E-05)† | 5.91E-01(5.21E-05)† | 6.27E-01(2.46E-05)† | 6.12E-01(4.17E-05)† | 7.37E-01(2.15E-06)† | 8.19E-01(8.72E-07) |
| | 50 | 6.74E-01(1.11E-05)† | 6.17E-01(4.14E-05)† | 6.29E-01(2.09E-05)† | 6.30E-01(2.72E-05)† | 7.36E-01(2.01E-06)† | 8.21E-01(4.46E-07) |
| | 200 | 6.76E-01(1.41E-05)† | 6.63E-01(1.71E-05)† | 6.32E-01(1.03E-05)† | 6.55E-01(2.11E-05)† | 7.35E-01(1.77E-06)† | 8.22E-01(3.04E-07) |
| WFG9 | 5 | 3.12E-01(2.74E-03)† | 3.19E-01(6.21E-04)† | 3.95E-01(2.36E-04)† | 5.27E-01(1.40E-02)‡ | 3.98E-01(9.90E-04)† | 4.29E-01(3.59E-04) |
| | 25 | 3.75E-01(1.05E-03)† | 3.62E-01(4.39E-04)† | 4.39E-01(1.40E-03) | 6.17E-01(1.12E-02)‡ | 4.12E-01(6.72E-04)† | 4.37E-01(7.81E-04) |
| | 50 | 3.88E-01(9.89E-04)† | 3.74E-01(4.32E-04)† | 4.83E-01(4.01E-04)† | 5.87E-01(1.33E-02)‡ | 4.17E-01(8.27E-04)† | 4.33E-01(6.54E-04) |
| | 200 | 4.23E-01(2.99E-03)† | 4.14E-01(1.65E-03)† | 5.04E-01(7.67E-04)† | 6.07E-01(1.16E-02)‡ | 4.14E-01(1.46E-03)† | 4.44E-01(2.87E-04) |
| +-= | | 52/0/0 | 52/0/0 | 51/0/1 | 45/7/0 | 48/4/0 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.3: Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 1.34E+01(2.61E+00)† | 1.83E+01(1.28E+00)† | 6.12E+01(5.75E+00)† | 3.49E+02(1.63E+03)† | 1.00E+01(2.64E+00)† | 1.22E-02(1.60E-07) |
| | 25 | 3.91E+01(3.75E+01)† | 1.71E+01(1.28E-01)† | 8.41E+01(2.53E+01)† | 6.15E+02(1.57E+03)† | 1.77E+01(2.29E+00)† | 1.66E-01(6.98E-02) |
| | 50 | 8.32E+01(9.32E+01)† | 1.67E+01(1.13E-01)† | 1.20E+02(5.15E+01)† | 6.78E+02(1.56E+03)† | 1.83E+01(1.77E+00)† | 6.07E-02(5.33E-03) |
| | 200 | 1.24E+02(1.37E+02)† | 1.48E+01(1.49E+01)† | 1.47E+02(1.48E+02)† | 7.45E+02(6.94E+02)† | 1.79E+01(1.76E+01)† | 2.05E-01(1.49E-02) |
| F2 | 5 | 8.47E-02(1.04E-05)† | 1.88E-01(5.09E-05)† | 2.61E-01(1.30E-04)† | 2.95E+00(1.06E-01)† | 9.16E-02(3.50E-05)† | 4.32E-02(5.85E-07) |
| | 25 | 2.48E-01(2.56E-04)† | 1.47E-01(5.34E-06)† | 3.65E-01(2.67E-04)† | 3.51E+00(5.61E-02)† | 6.69E-02(4.81E-06)† | 4.11E-02(6.33E-07) |
| | 50 | 3.25E-01(1.92E-04)† | 1.33E-01(2.41E-06)† | 3.92E-01(6.37E-05)† | 3.22E+00(8.82E-02)† | 6.61E-02(8.97E-06)† | 4.14E-02(4.82E-07) |
| | 200 | 3.83E-01(4.01E-01)† | 1.08E-01(1.08E-01)† | 4.06E-01(4.03E-01)† | 3.11E+00(3.23E+00)† | 6.89E-02(6.82E-02)† | 4.23E-02(4.26E-02) |
| F3 | 5 | 3.10E+01(6.38E+00)† | 5.31E+01(8.03E+00)† | 1.67E+02(1.07E+01)† | 4.66E+02(2.24E+03)† | 2.45E+01(7.75E+00)† | 1.76E-01(9.17E-02) |
| | 25 | 6.78E+01(8.41E+01)† | 4.84E+01(1.88E+00)† | 2.02E+02(1.58E+02)† | 5.77E+02(9.26E+02)† | 4.32E+01(1.79E+01)† | 5.71E-01(4.60E-01) |
| | 50 | 1.50E+02(3.39E+02)† | 4.43E+01(1.33E+00)† | 2.67E+02(1.11E+02)† | 5.65E+02(8.04E+02)† | 4.34E+01(9.69E+00)† | 5.23E-01(5.80E-01) |
| | 200 | 2.30E+02(2.42E+02)† | 3.62E+01(3.75E+01)† | 3.53E+02(3.66E+02)† | 4.63E+02(5.93E+02)† | 4.87E+01(4.21E+01)† | 4.11E-02(4.19E-02) |
| F4 | 5 | 9.44E-02(1.03E-04)† | 8.65E-02(2.42E-04)† | 3.22E-01(2.98E-04)† | 4.46E-01(1.96E-03)† | 1.02E-01(1.72E-04)† | 2.40E-02(2.36E-04) |
| | 25 | 4.00E-01(7.24E-04)† | 1.36E-01(4.86E-05)† | 4.99E-01(1.48E-04)† | 4.05E-01(3.03E-04)† | 8.44E-02(3.69E-05)† | 2.00E-02(2.66E-06) |
| | 50 | 5.11E-01(3.54E-04)† | 1.43E-01(1.55E-05)† | 5.22E-01(8.30E-05)† | 4.24E-01(2.67E-04)† | 8.56E-02(2.85E-05)† | 1.98E-02(3.96E-06) |
| | 200 | 5.51E-01(5.38E-01)† | 1.28E-01(1.29E-01)† | 5.43E-01(5.36E-01)† | 4.24E-01(4.24E-01)† | 9.04E-02(8.05E-02)† | 2.16E-02(1.77E-02) |
| WFG1 | 5 | 2.31E-01(1.66E-04)† | 3.91E-01(1.32E-04)† | 4.85E-01(1.45E-04)† | 1.46E+00(1.02E-02)† | 2.18E-01(1.52E-04)† | 2.02E-01(5.20E-05) |
| | 25 | 3.01E-01(5.42E-05)† | 3.67E-01(8.25E-05)† | 5.56E-01(4.05E-05)† | 1.72E+00(2.61E-03)† | 2.56E-01(2.72E-05)† | 1.98E-01(2.38E-05) |
| | 50 | 3.19E-01(3.17E-05)† | 3.28E-01(4.13E-05)† | 5.65E-01(2.12E-05)† | 1.76E+00(1.46E-03)† | 2.53E-01(2.41E-05)† | 1.97E-01(2.29E-05) |
| | 200 | 3.24E-01(3.25E-01)† | 2.81E-01(2.78E-01)† | 5.69E-01(5.64E-01)† | 1.83E+00(1.82E+00)† | 2.30E-01(2.29E-01)† | 1.95E-01(2.00E-01) |
| WFG2 | 5 | 6.87E-01(7.21E-04)‡ | 5.33E-01(1.35E-03)‡ | 9.02E-01(3.75E-04)† | 7.11E-01(1.22E-05)‡ | 5.92E-01(9.74E-04)‡ | 7.35E-01(3.28E-04) |
| | 25 | 8.64E-01(1.05E-03)† | 4.76E-01(7.11E-04)‡ | 1.01E+00(9.68E-04)† | 7.09E-01(1.45E-05) | 5.40E-01(5.92E-04)‡ | 7.09E-01(2.39E-04) |
| | 50 | 9.24E-01(1.26E-03)† | 4.75E-01(7.63E-04)‡ | 1.05E+00(1.05E-03)† | 7.06E-01(1.83E-05)† | 5.27E-01(5.66E-04)‡ | 6.95E-01(1.70E-04) |
| | 200 | 9.76E-01(9.75E-01)† | 4.82E-01(4.62E-01)‡ | 1.10E+00(1.11E+00)† | 7.05E-01(6.90E-01)† | 5.50E-01(4.87E-01)‡ | 6.91E-01(7.11E-01) |
| WFG3 | 5 | 9.71E-02(1.96E-03)‡ | 1.59E+00(1.67E-02)† | 1.30E+00(4.15E-03)† | 7.39E-01(5.82E-05)† | 1.15E+00(5.21E-03)† | 6.31E-01(2.07E-02) |
| | 25 | 8.45E-01(2.17E-02)‡ | 2.00E+00(1.49E-03)† | 1.58E+00(2.58E-03)† | 7.45E-01(5.74E-05)‡ | 1.52E+00(2.69E-03)† | 1.28E+00(1.50E-02) |
| | 50 | 1.28E+00(1.98E-02)‡ | 2.00E+00(3.32E-03)† | 1.63E+00(1.21E-03)† | 7.46E-01(5.18E-05)‡ | 1.60E+00(1.64E-03)† | 1.36E+00(7.99E-03) |
| | 200 | 1.64E+00(1.64E+00)† | 1.99E+00(1.98E+00)† | 1.68E+00(1.66E+00)† | 7.40E-01(7.36E-01)‡ | 1.57E+00(1.63E+00)† | 1.57E+00(1.43E+00) |
| WFG4 | 5 | 3.98E-01(1.05E-04)† | 4.91E-01(3.81E-04)† | 5.75E-01(6.48E-05)† | 8.06E-01(7.74E-04)† | 3.56E-01(9.81E-05)† | 2.94E-01(5.78E-05) |
| | 25 | 4.77E-01(5.85E-05)† | 3.73E-01(2.06E-05)† | 5.76E-01(8.39E-05)† | 7.65E-01(2.61E-04)† | 2.44E-01(1.01E-05)‡ | 2.69E-01(2.25E-05) |
| | 50 | 4.94E-01(5.02E-05)† | 3.52E-01(1.13E-05)† | 5.73E-01(6.42E-05)† | 7.64E-01(3.28E-04)† | 2.31E-01(4.58E-06)‡ | 2.64E-01(1.52E-05) |
| | 200 | 5.06E-01(5.00E-01)† | 3.18E-01(3.18E-01)† | 5.55E-01(5.58E-01)† | 7.65E-01(7.62E-01)† | 2.11E-01(2.10E-01)‡ | 2.52E-01(2.56E-01) |
| WFG5 | 5 | 4.20E-01(6.60E-05)† | 5.44E-01(2.16E-04)† | 6.87E-01(7.97E-05)† | 1.37E+00(3.08E-04)† | 4.01E-01(4.23E-05)† | 3.47E-01(3.32E-05) |
| | 25 | 4.83E-01(1.06E-04)† | 4.02E-01(3.04E-05)† | 6.50E-01(5.66E-05)† | 1.32E+00(1.43E-04)† | 3.06E-01(1.45E-05)‡ | 3.21E-01(1.79E-05) |
| | 50 | 4.98E-01(7.16E-05)† | 3.70E-01(2.13E-05)† | 6.38E-01(4.25E-05)† | 1.31E+00(1.29E-04)† | 2.90E-01(1.07E-05)‡ | 3.16E-01(4.31E-05) |
| | 200 | 5.03E-01(5.04E-01)† | 3.31E-01(3.33E-01)† | 6.32E-01(6.16E-01)† | 1.29E+00(1.29E+00)† | 2.79E-01(2.78E-01)‡ | 3.16E-01(3.23E-01) |
| WFG6 | 5 | 6.54E-01(9.82E-03)† | 7.28E-01(6.17E-03)† | 8.93E-01(5.99E-03)† | 1.25E+00(3.27E-04)† | 7.65E-01(1.44E-02)† | 3.93E-01(1.18E-03) |
| | 25 | 6.78E-01(5.71E-03)† | 5.37E-01(3.18E-03)† | 8.29E-01(5.21E-03)† | 1.25E+00(1.42E-04)† | 7.08E-01(1.79E-02)† | 3.73E-01(1.16E-03) |
| | 50 | 6.81E-01(6.80E-03)† | 5.11E-01(3.48E-03)† | 8.35E-01(6.54E-03)† | 1.25E+00(8.26E-05)† | 7.41E-01(2.73E-02)† | 3.88E-01(2.78E-03) |
| | 200 | 5.63E-01(5.70E-01)† | 4.71E-01(4.88E-01)† | 7.57E-01(8.19E-01)† | 1.24E+00(1.26E+00)† | 7.76E-01(7.55E-01)† | 3.89E-01(4.35E-01) |
| WFG7 | 5 | 4.94E-01(1.29E-04)† | 6.76E-01(1.62E-03)† | 7.14E-01(1.10E-04)† | 8.46E-01(1.11E-03)† | 3.64E-01(5.51E-05)† | 2.70E-01(3.59E-05) |
| | 25 | 5.46E-01(6.54E-05)† | 4.50E-01(5.87E-05)† | 7.33E-01(3.82E-05)† | 8.02E-01(6.89E-04)† | 2.77E-01(1.08E-05)† | 2.50E-01(2.66E-05) |
| | 50 | 5.49E-01(5.20E-05)† | 4.26E-01(3.27E-05)† | 7.23E-01(3.42E-05)† | 8.31E-01(4.00E-04)† | 2.67E-01(8.92E-06)† | 2.43E-01(2.75E-05) |
| | 200 | 5.40E-01(5.42E-01)† | 3.74E-01(3.85E-01)† | 7.14E-01(7.17E-01)† | 6.32E-01(7.35E-01)† | 2.56E-01(2.53E-01)† | 2.12E-01(2.17E-01) |
| WFG8 | 5 | 4.95E-01(1.35E-04)† | 6.98E-01(2.34E-03)† | 7.15E-01(1.01E-04)† | 8.50E-01(7.99E-04)† | 3.62E-01(1.25E-04)† | 2.68E-01(7.45E-05) |
| | 25 | 5.45E-01(5.08E-05)† | 4.49E-01(6.93E-05)† | 7.31E-01(5.69E-05)† | 7.97E-01(9.59E-04)† | 2.76E-01(8.31E-06)† | 2.50E-01(2.90E-05) |
| | 50 | 5.49E-01(3.92E-05)† | 4.27E-01(4.31E-05)† | 7.23E-01(3.32E-05)† | 8.23E-01(5.59E-04)† | 2.67E-01(5.87E-06)† | 2.43E-01(3.27E-05) |
| | 200 | 5.50E-01(5.32E-01)† | 3.86E-01(3.78E-01)† | 7.13E-01(7.12E-01)† | 6.55E-01(6.54E-01)† | 2.56E-01(2.49E-01)† | 2.20E-01(2.09E-01) |
| WFG9 | 5 | 9.95E-01(1.02E-02)† | 8.93E-01(3.76E-03)‡ | 1.14E+00(2.11E-03)† | 9.00E-01(2.83E-03)‡ | 1.06E+00(6.13E-03)† | 9.43E-01(3.78E-03) |
| | 25 | 1.10E+00(5.35E-03)† | 9.07E-01(2.08E-03) | 1.08E+00(5.90E-03)‡ | 8.83E-01(1.85E-03) | 1.04E+00(5.04E-03)† | 9.16E-01(4.91E-03) |
| | 50 | 1.11E+00(5.48E-03)‡ | 8.99E-01(2.75E-03) | 9.96E-01(2.19E-03)‡ | 8.87E-01(2.70E-03) | 1.03E+00(5.11E-03)† | 9.13E-01(3.50E-03) |
| | 200 | 1.20E+00(9.24E-01)† | 9.06E-01(9.28E-01)‡ | 1.03E+00(8.93E-01)† | 9.13E-01(9.22E-01) | 1.17E+00(1.08E+00)† | 1.00E+00(8.55E-01) |
| +-= | | 47/5/0 | 44/6/2 | 50/2/0 | 43/5/4 | 42/10/0 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.4: Mean and standard deviation values of MMS metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 7.97E-01(4.74E-05)† | 8.21E-01(7.68E-05)† | 9.92E-01(3.67E-05)† | 9.34E-01(5.32E-04)† | 8.22E-01(3.97E-04)† | 1.00E+00(9.32E-11) |
| | 25 | 8.85E-01(2.34E-04)† | 9.04E-01(7.22E-06)† | 9.99E-01(4.63E-08)‡ | 9.36E-01(3.61E-04)† | 9.12E-01(1.81E-04)† | 9.87E-01(6.37E-05) |
| | 50 | 9.24E-01(1.51E-04)† | 9.08E-01(1.81E-06)† | 1.00E+00(4.63E-08)‡ | 9.64E-01(1.68E-04)† | 9.20E-01(2.48E-04)† | 1.00E+00(1.86E-06) |
| | 200 | 9.48E-01(7.29E-05)† | 9.09E-01(8.58E-09)† | 9.92E-01(1.02E-04)† | 9.77E-01(8.87E-05)† | 9.17E-01(7.58E-05)† | 9.99E-01(2.94E-06) |
| F2 | 5 | 8.78E-01(8.03E-05)† | 9.72E-01(1.36E-05)† | 9.90E-01(1.31E-05)† | 8.73E-01(2.09E-03)† | 9.70E-01(2.75E-05)† | 9.98E-01(6.68E-06) |
| | 25 | 9.52E-01(1.23E-05)† | 9.65E-01(6.81E-06)† | 9.98E-01(3.91E-06)‡ | 8.15E-01(1.42E-03)† | 9.51E-01(5.17E-06)† | 9.89E-01(7.80E-06) |
| | 50 | 9.60E-01(7.97E-06)† | 9.60E-01(4.87E-06)† | 9.97E-01(4.46E-06)† | 7.35E-01(1.33E-03)† | 9.48E-01(2.62E-06)† | 9.98E-01(1.96E-06) |
| | 200 | 9.65E-01(6.26E-06)† | 9.48E-01(2.22E-06)† | 9.96E-01(6.46E-06)† | 6.24E-01(3.05E-04)† | 9.44E-01(1.56E-06)† | 9.98E-01(2.95E-06) |
| F3 | 5 | 8.67E-01(3.02E-05)† | 9.11E-01(6.50E-05)† | 9.82E-01(1.42E-04)† | 8.58E-01(9.87E-04)† | 8.92E-01(3.66E-04)† | 9.99E-01(7.73E-06) |
| | 25 | 9.31E-01(7.82E-05)† | 9.50E-01(1.37E-05)† | 9.98E-01(2.86E-07)‡ | 8.31E-01(1.01E-03)† | 9.54E-01(1.01E-04)† | 9.92E-01(2.97E-05) |
| | 50 | 9.58E-01(4.59E-05)† | 9.47E-01(3.63E-06)† | 9.99E-01(2.72E-07)† | 8.66E-01(1.98E-03)† | 9.52E-01(7.95E-05)† | 1.00E+00(7.68E-07) |
| | 200 | 9.67E-01(3.88E-05)† | 9.43E-01(3.00E-07)† | 9.81E-01(2.64E-04)† | 8.42E-01(1.30E-03)† | 9.49E-01(6.17E-05)† | 1.00E+00(1.04E-06) |
| F4 | 5 | 7.96E-01(3.02E-04) | 8.09E-01(8.71E-04) | 9.49E-01(5.25E-04)‡ | 8.94E-01(2.65E-04)‡ | 9.39E-01(2.64E-04)‡ | 7.92E-01(2.23E-03) |
| | 25 | 9.23E-01(7.61E-05)‡ | 9.12E-01(3.28E-04)‡ | 9.70E-01(2.36E-04)‡ | 9.18E-01(4.45E-05)‡ | 9.41E-01(4.05E-06)‡ | 8.08E-01(8.75E-04) |
| | 50 | 9.44E-01(1.62E-05)‡ | 9.33E-01(8.91E-05)‡ | 9.74E-01(1.51E-04)‡ | 9.22E-01(7.06E-05)‡ | 9.42E-01(6.33E-12)‡ | 8.08E-01(1.70E-03) |
| | 200 | 9.49E-01(3.03E-05)‡ | 9.42E-01(5.09E-32)‡ | 9.75E-01(1.85E-04)‡ | 9.31E-01(2.88E-05)‡ | 9.42E-01(5.79E-12)‡ | 8.13E-01(3.10E-03) |
| WFG1 | 5 | 6.57E-01(2.75E-04)† | 5.81E-01(2.16E-04)† | 7.26E-01(1.66E-04)† | 4.52E-01(2.73E-04)† | 6.68E-01(2.36E-04)† | 9.42E-01(1.60E-04) |
| | 25 | 7.84E-01(1.16E-04)† | 5.75E-01(7.51E-04)† | 8.17E-01(1.32E-04)† | 4.93E-01(2.58E-04)† | 7.49E-01(3.07E-04)† | 9.50E-01(2.56E-05) |
| | 50 | 8.32E-01(3.48E-05)† | 5.40E-01(1.16E-03)† | 8.61E-01(8.56E-05)† | 5.12E-01(9.31E-05)† | 8.06E-01(1.21E-04)† | 9.65E-01(2.16E-05) |
| | 200 | 8.84E-01(7.62E-06)† | 5.16E-01(9.18E-05)† | 9.19E-01(2.63E-05)† | 5.12E-01(3.22E-05)† | 8.73E-01(1.43E-05)† | 9.68E-01(1.85E-05) |
| WFG2 | 5 | 7.33E-01(1.76E-04)† | 5.89E-01(2.46E-04)† | 8.18E-01(8.66E-05)† | 1.83E-01(4.36E-06)† | 7.91E-01(1.17E-04)† | 9.96E-01(4.43E-05) |
| | 25 | 8.61E-01(2.79E-05)† | 5.13E-01(1.74E-04)† | 9.09E-01(5.98E-05)† | 1.86E-01(6.16E-06)† | 8.57E-01(5.13E-05)† | 9.88E-01(1.65E-05) |
| | 50 | 8.74E-01(2.13E-05)† | 5.13E-01(7.30E-05)† | 9.19E-01(5.79E-05)† | 1.87E-01(4.17E-06)† | 8.62E-01(7.79E-05)† | 9.61E-01(2.00E-02) |
| | 200 | 8.77E-01(1.27E-05)† | 5.23E-01(1.91E-05)† | 9.19E-01(7.63E-05)† | 1.88E-01(7.95E-07)† | 8.56E-01(1.01E-04)† | 9.82E-01(4.79E-06) |
| WFG3 | 5 | 9.76E-01(2.17E-04)† | 9.67E-01(2.75E-04)† | 9.10E-01(2.61E-04)† | 4.72E-01(1.15E-05)† | 9.74E-01(4.45E-04)† | 9.96E-01(6.65E-05) |
| | 25 | 9.66E-01(3.99E-04)† | 9.74E-01(2.47E-04)† | 9.22E-01(2.14E-04)† | 4.72E-01(9.50E-06)† | 9.62E-01(6.46E-04)† | 9.84E-01(1.91E-04) |
| | 50 | 9.55E-01(2.22E-04)† | 9.73E-01(2.50E-04)† | 9.19E-01(2.18E-04)† | 4.70E-01(8.65E-06)† | 9.67E-01(6.24E-04)† | 9.98E-01(1.33E-05) |
| | 200 | 9.35E-01(3.43E-04)† | 9.74E-01(2.32E-04)† | 9.03E-01(1.91E-04)† | 4.73E-01(8.66E-06)† | 9.03E-01(5.88E-02)† | 9.93E-01(1.29E-03) |
| WFG4 | 5 | 8.86E-01(2.14E-04)† | 9.37E-01(1.68E-04)† | 9.48E-01(4.84E-05)† | 8.28E-01(7.42E-05)† | 9.56E-01(3.50E-05)† | 1.00E+00(4.82E-07) |
| | 25 | 9.49E-01(1.03E-05)† | 9.37E-01(5.50E-05)† | 9.66E-01(2.89E-05)† | 8.40E-01(4.81E-06)† | 9.44E-01(1.33E-06)† | 9.90E-01(3.81E-06) |
| | 50 | 9.52E-01(1.27E-05)† | 9.43E-01(4.12E-05)† | 9.67E-01(2.62E-05)† | 8.40E-01(3.69E-06)† | 9.42E-01(1.97E-07)† | 1.00E+00(1.25E-07) |
| | 200 | 9.50E-01(8.82E-06)† | 9.50E-01(1.79E-05)† | 9.65E-01(3.48E-05)† | 8.37E-01(5.31E-06)† | 9.41E-01(5.01E-07)† | 1.00E+00(3.64E-07) |
| WFG5 | 5 | 8.58E-01(1.11E-04)† | 9.36E-01(7.70E-05)† | 9.58E-01(1.45E-05)† | 8.23E-01(2.85E-05)† | 9.57E-01(2.10E-05)† | 9.84E-01(1.74E-06) |
| | 25 | 9.37E-01(6.04E-06)† | 9.26E-01(3.85E-05)† | 9.68E-01(8.62E-06)† | 8.24E-01(7.43E-06)† | 9.32E-01(1.41E-06)† | 9.79E-01(5.64E-06) |
| | 50 | 9.41E-01(4.57E-06)† | 9.26E-01(7.13E-05)† | 9.69E-01(7.51E-06)† | 8.19E-01(9.31E-06)† | 9.33E-01(1.18E-06)† | 9.85E-01(3.41E-07) |
| | 200 | 9.42E-01(7.98E-06)† | 9.23E-01(2.34E-05)† | 9.69E-01(7.27E-06)† | 8.18E-01(1.16E-05)† | 9.34E-01(4.35E-07)† | 9.86E-01(3.70E-07) |
| WFG6 | 5 | 8.29E-01(4.67E-04)† | 8.75E-01(3.44E-04)† | 9.27E-01(3.27E-04)† | 3.81E-01(4.08E-06)† | 8.96E-01(5.65E-04)† | 9.73E-01(1.21E-04) |
| | 25 | 9.08E-01(2.56E-04)† | 8.97E-01(3.46E-04)† | 9.40E-01(2.70E-04)† | 3.86E-01(3.90E-06)† | 8.77E-01(5.92E-04)† | 9.70E-01(6.02E-05) |
| | 50 | 9.14E-01(2.89E-04)† | 9.04E-01(3.19E-04)† | 9.40E-01(3.36E-04)† | 3.88E-01(3.95E-06)† | 8.69E-01(7.67E-04)† | 9.75E-01(9.92E-05) |
| | 200 | 9.21E-01(2.81E-04)† | 9.08E-01(3.38E-04)† | 9.43E-01(2.15E-04)† | 3.87E-01(1.74E-06)† | 8.78E-01(6.39E-04)† | 9.79E-01(4.16E-05) |
| WFG7 | 5 | 9.54E-01(2.11E-05)† | 9.36E-01(8.06E-05)† | 9.77E-01(1.12E-05)† | 9.72E-01(1.86E-05)† | 9.66E-01(2.16E-05)† | 9.97E-01(3.36E-06) |
| | 25 | 9.53E-01(1.52E-05)† | 9.53E-01(1.13E-05)† | 9.79E-01(1.30E-05)† | 9.58E-01(1.24E-05)† | 9.43E-01(1.41E-06)† | 9.89E-01(5.31E-06) |
| | 50 | 9.50E-01(6.99E-06)† | 9.55E-01(1.57E-05)† | 9.79E-01(1.28E-05)† | 9.47E-01(5.27E-06)† | 9.42E-01(2.63E-07)† | 9.97E-01(3.00E-06) |
| | 200 | 9.46E-01(7.39E-06)† | 9.63E-01(1.31E-05)† | 9.81E-01(7.83E-06)† | 9.32E-01(1.41E-05)† | 9.41E-01(5.96E-07)† | 9.97E-01(2.33E-06) |
| WFG8 | 5 | 9.55E-01(2.61E-05)† | 9.37E-01(6.11E-05)† | 9.78E-01(1.23E-05)† | 9.70E-01(4.57E-05)† | 9.66E-01(1.99E-05)† | 9.96E-01(3.11E-06) |
| | 25 | 9.53E-01(1.19E-05)† | 9.52E-01(1.19E-05)† | 9.79E-01(1.24E-05)† | 9.58E-01(1.13E-05)† | 9.43E-01(3.45E-06)† | 9.89E-01(5.33E-06) |
| | 50 | 9.49E-01(1.46E-05)† | 9.56E-01(8.54E-06)† | 9.79E-01(1.66E-05)† | 9.47E-01(1.41E-05)† | 9.42E-01(6.83E-07)† | 9.97E-01(2.21E-06) |
| | 200 | 9.47E-01(1.00E-05)† | 9.62E-01(1.13E-05)† | 9.81E-01(1.03E-05)† | 9.32E-01(1.34E-05)† | 9.41E-01(5.48E-07)† | 9.97E-01(2.40E-06) |
| WFG9 | 5 | 6.99E-01(8.33E-03)† | 8.05E-01(8.81E-04)† | 8.65E-01(8.84E-05)‡ | 9.22E-01(2.34E-03)‡ | 8.43E-01(2.95E-04) | 8.30E-01(2.39E-03) |
| | 25 | 7.94E-01(3.71E-03)† | 8.38E-01(2.78E-04) | 8.93E-01(4.57E-04)‡ | 9.41E-01(9.21E-04)‡ | 8.35E-01(4.57E-04) | 8.26E-01(2.89E-03) |
| | 50 | 8.12E-01(2.50E-03) | 8.42E-01(2.75E-04) | 9.12E-01(1.03E-04)‡ | 9.30E-01(9.94E-04)‡ | 8.35E-01(5.43E-04) | 8.18E-01(2.29E-03) |
| | 200 | 8.44E-01(1.28E-03) | 8.61E-01(4.60E-04)‡ | 9.19E-01(1.35E-04)‡ | 9.28E-01(9.04E-04)‡ | 8.33E-01(7.57E-04) | 8.33E-01(1.12E-03) |
| +-= | | 46/3/3 | 45/4/3 | 40/12/0 | 44/8/0 | 44/4/4 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

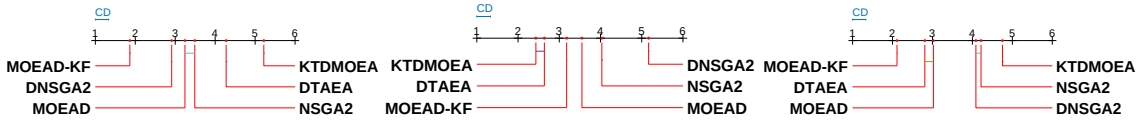
KTDMOEA gets worse MHV value due to both worse MMS and MGD values. The reasons why KTDMOEA performs worse on these problems will be elaborated in Section 5.4.1.4.

#### 5.4.1.2 NObj Decreasing from 7 to 2 and then Increasing from 2 to 7

Figure 5.6 presents the Nemenyi post-tests results among MHV, MGD and MMS of obtained solutions at the first generation by 6 algorithms. Friedman test detects significant differences in average value for MHV, MGD and MMS with a p-value of 1.97E-240, 4.72E-221, and 6.87E-113, respectively. Tables 5.5-5.7 show mean and standard deviation values of MHV, MGD and MMS of obtained solutions in the first generation after changes averaged across 10 environmental changes in the sequence of changes as '7-2-7 one by one'.

Overall, it can be found from the Friedman test results in Figure 5.6 that KTDMOEA performs significantly better than all others regarding MHV and MMS metrics. There is no significant difference between KTDMOEA and DTAEA regarding MGD. This further supports that the proposed knowledge transfer technique via PS expansion/contraction indeed improves the diversity and maintains the convergence of transferred solutions right after changes, under all frequencies of changes on most problems.

Specifically, it is clear from Table 5.5 that KTDMOEA gets worse MHV and MMS values on F4 and WFG9, which is similar to the results of the sequence of '2-7-2 one by one'. In addition, KTDMOEA gets worse MHV and MGD values than MOEAD on F1 and F3 in all cases (including all environmental changes and all parameter settings) except for cases of increasing the NObj and large values of $\tau_t$ at the same time. Additionally, it is clear from Tables 5.5-5.7 that KTDMOEA achieves worse mean MHV and MMS values on WFG3 than NSGA2 and DNSGA2. The reasons why KTDMOEA performs worse on these problems will be elaborated in Section 5.4.1.4.




(a) MHV (b) MGD (c) MMS

Figure 5.6: Friedman ranking among MHV, MGD and MMS of obtained solutions at the first generation by 6 algorithms in the changing sequence of firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

Table 5.5: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 0.00E+00(0.00E+00)† | 4.91E-01(3.45E-03)‡ | 0.00E+00(0.00E+00)† | 2.17E-02(7.99E-04)† | 3.23E-01(4.43E-03) | 3.05E-01(1.68E-02) |
| | 25 | 1.08E-01(1.22E-02)† | 8.32E-01(3.37E-04)‡ | 5.31E-02(6.50E-03)† | 0.00E+00(0.00E+00)† | 5.71E-01(2.71E-03)† | 6.76E-01(4.58E-03) |
| | 50 | 3.62E-01(2.30E-03)† | 8.48E-01(4.39E-04)‡ | 3.75E-01(1.89E-03)† | 0.00E+00(0.00E+00)† | 7.41E-01(3.67E-03)† | 8.18E-01(2.02E-03) |
| | 200 | 4.61E-01(1.98E-03)† | 8.93E-01(7.51E-05)‡ | 3.56E-01(2.16E-03)† | 0.00E+00(0.00E+00)† | 8.00E-01(3.09E-03)† | 8.80E-01(6.74E-04) |
| F2 | 5 | 8.00E-01(9.63E-04)† | 9.03E-01(1.36E-05)† | 8.57E-01(3.07E-05)† | 2.67E-01(3.02E-03)† | 9.23E-01(1.18E-05)† | 9.46E-01(2.50E-05) |
| | 25 | 8.56E-01(3.32E-04)† | 9.18E-01(1.42E-06)† | 8.70E-01(1.46E-05)† | 2.48E-01(5.80E-04)† | 9.28E-01(1.00E-06)† | 9.51E-01(1.41E-05) |
| | 50 | 8.70E-01(2.08E-04)† | 9.20E-01(1.12E-06)† | 8.73E-01(2.01E-05)† | 2.10E-01(1.49E-04)† | 9.29E-01(1.27E-06)† | 9.54E-01(1.59E-06) |
| | 200 | 8.76E-01(1.69E-04)† | 9.23E-01(1.84E-07)† | 8.70E-01(3.08E-05)† | 2.02E-01(3.23E-05)† | 9.29E-01(5.91E-07)† | 9.54E-01(4.50E-06) |
| F3 | 5 | 0.00E+00(0.00E+00)† | 2.81E-01(1.77E-03)‡ | 0.00E+00(0.00E+00)† | 0.00E+00(0.00E+00)† | 1.43E-01(2.03E-03) | 1.30E-01(1.33E-02) |
| | 25 | 7.37E-03(5.08E-04)† | 5.89E-01(1.54E-03)‡ | 1.10E-02(7.74E-04)† | 0.00E+00(0.00E+00)† | 3.35E-01(2.58E-03)† | 3.86E-01(1.36E-02) |
| | 50 | 3.06E-01(2.49E-03)† | 6.42E-01(9.67E-04)‡ | 2.99E-01(2.68E-03)† | 0.00E+00(0.00E+00)† | 4.51E-01(2.45E-03)† | 5.70E-01(3.52E-03) |
| | 200 | 3.45E-01(1.45E-03)† | 7.41E-01(1.05E-03)‡ | 2.88E-01(1.39E-03)† | 0.00E+00(0.00E+00)† | 5.35E-01(1.39E-03)† | 6.50E-01(1.78E-03) |
| F4 | 5 | 5.30E-01(1.81E-04)† | 8.96E-01(4.48E-05) | 7.75E-01(1.66E-04)† | 8.10E-01(4.23E-04)† | 9.01E-01(4.18E-05) | 8.99E-01(4.34E-04) |
| | 25 | 6.97E-01(7.08E-05)† | 9.11E-01(2.84E-05)‡ | 8.08E-01(1.64E-04)† | 8.86E-01(1.56E-04)† | 9.26E-01(3.07E-06)‡ | 9.01E-01(2.78E-04) |
| | 50 | 7.48E-01(2.03E-05)† | 9.16E-01(9.81E-06)‡ | 8.18E-01(1.54E-04)† | 8.91E-01(1.61E-04)† | 9.27E-01(2.97E-06)‡ | 9.01E-01(1.63E-04) |
| | 200 | 7.77E-01(1.84E-05)† | 9.23E-01(6.28E-07)‡ | 8.15E-01(1.16E-04)† | 8.97E-01(9.01E-05) | 9.28E-01(3.37E-06)‡ | 8.98E-01(4.11E-04) |
| WFG1 | 5 | 8.46E-01(7.62E-05)† | 7.22E-01(1.19E-04)† | 8.21E-01(6.89E-05)† | 8.01E-01(3.97E-06)† | 8.34E-01(4.45E-05)† | 8.56E-01(2.55E-05) |
| | 25 | 8.89E-01(1.92E-05)† | 7.96E-01(3.77E-05)† | 8.76E-01(1.91E-05)† | 7.55E-01(1.04E-03)† | 8.91E-01(1.56E-05)† | 9.11E-01(2.02E-05) |
| | 50 | 9.08E-01(8.21E-06)† | 8.23E-01(2.87E-05)† | 8.97E-01(2.33E-05)† | 7.67E-01(1.12E-03)† | 9.08E-01(3.55E-06)† | 9.27E-01(2.88E-06) |
| | 200 | 9.21E-01(2.62E-06)† | 8.59E-01(6.94E-06)† | 9.13E-01(6.47E-06)† | 7.42E-01(5.14E-04)† | 9.20E-01(4.77E-07)† | 9.35E-01(4.67E-06) |
| WFG2 | 5 | 8.30E-01(3.29E-04)† | 7.63E-01(2.49E-04)† | 8.48E-01(7.30E-05)† | 2.36E-01(2.08E-06)† | 8.84E-01(2.09E-04)† | 9.26E-01(1.92E-05) |
| | 25 | 8.77E-01(1.47E-04)† | 8.31E-01(1.23E-04)† | 8.69E-01(1.84E-04)† | 2.38E-01(5.00E-07)† | 9.11E-01(1.17E-04)† | 9.42E-01(1.36E-05) |
| | 50 | 8.76E-01(2.41E-04)† | 8.44E-01(1.16E-04)† | 8.68E-01(1.57E-04)† | 2.38E-01(1.12E-06)† | 9.10E-01(1.21E-04)† | 9.43E-01(1.11E-05) |
| | 200 | 8.77E-01(1.67E-04)† | 8.58E-01(1.23E-04)† | 8.66E-01(1.94E-04)† | 2.39E-01(4.82E-07)† | 9.14E-01(1.13E-04)† | 9.47E-01(8.09E-06) |
| WFG3 | 5 | 6.01E-01(1.21E-04) | 5.29E-01(3.77E-04)† | 5.86E-01(6.57E-05)† | 2.43E-01(9.19E-06)† | 5.86E-01(7.53E-05)† | 6.04E-01(5.83E-05) |
| | 25 | 6.32E-01(6.06E-05)‡ | 6.03E-01(1.16E-04)† | 6.35E-01(4.30E-05)‡ | 2.49E-01(1.33E-06)† | 6.26E-01(4.03E-05) | 6.26E-01(2.48E-05) |
| | 50 | 6.40E-01(6.71E-05)‡ | 6.11E-01(9.51E-05)† | 6.39E-01(5.67E-05)‡ | 2.49E-01(1.20E-06)† | 6.29E-01(5.77E-05)† | 6.32E-01(4.96E-05) |
| | 200 | 6.46E-01(7.20E-05)‡ | 6.18E-01(8.56E-05)† | 6.41E-01(6.79E-05) | 2.50E-01(8.68E-07)† | 6.30E-01(3.81E-05)† | 6.40E-01(2.77E-05) |
| WFG4 | 5 | 6.08E-01(1.08E-04)† | 5.73E-01(2.35E-04)† | 5.98E-01(4.40E-05)† | 2.97E-01(6.42E-05)† | 6.94E-01(4.99E-05)† | 7.87E-01(1.83E-05) |
| | 25 | 6.69E-01(2.68E-05)† | 6.19E-01(1.41E-04)† | 6.40E-01(2.67E-05)† | 3.01E-01(1.28E-05)† | 7.49E-01(3.75E-07)† | 8.11E-01(1.20E-06) |
| | 50 | 6.75E-01(2.13E-05)† | 6.37E-01(7.92E-05)† | 6.40E-01(2.55E-05)† | 3.00E-01(1.15E-05)† | 7.52E-01(4.10E-07)† | 8.14E-01(1.07E-06) |
| | 200 | 6.85E-01(1.98E-05)† | 6.66E-01(3.24E-05)† | 6.51E-01(1.87E-05)† | 3.01E-01(3.62E-06)† | 7.53E-01(2.76E-07)† | 8.16E-01(7.47E-07) |
| WFG5 | 5 | 6.09E-01(2.53E-04)† | 5.81E-01(1.20E-04)† | 5.85E-01(4.65E-05)† | 5.21E-01(2.06E-05)† | 7.32E-01(3.52E-05)† | 7.55E-01(2.15E-05) |
| | 25 | 6.41E-01(4.72E-05)† | 6.09E-01(4.79E-05)† | 6.17E-01(2.57E-05)† | 5.20E-01(2.99E-06)† | 7.23E-01(1.19E-06)† | 7.79E-01(5.47E-06) |
| | 50 | 6.46E-01(2.33E-05)† | 6.20E-01(2.34E-05)† | 6.24E-01(1.94E-05)† | 5.19E-01(3.18E-06)† | 7.24E-01(1.82E-06)† | 7.83E-01(2.50E-06) |
| | 200 | 6.52E-01(2.58E-05)† | 6.42E-01(1.66E-05)† | 6.29E-01(1.72E-05)† | 5.19E-01(1.66E-06)† | 7.25E-01(1.16E-06)† | 7.84E-01(2.35E-06) |
| WFG6 | 5 | 4.75E-01(7.66E-04)† | 4.39E-01(4.71E-04)† | 4.97E-01(6.67E-04)† | 1.44E-01(2.23E-06)† | 5.37E-01(1.94E-03)† | 6.69E-01(2.98E-03) |
| | 25 | 5.70E-01(9.47E-04)† | 5.16E-01(6.80E-04)† | 5.64E-01(7.15E-04)† | 1.49E-01(5.21E-07)† | 5.77E-01(2.20E-03)† | 7.20E-01(2.63E-03) |
| | 50 | 5.94E-01(1.01E-03)† | 5.39E-01(6.64E-04)† | 5.68E-01(8.08E-04)† | 1.50E-01(2.66E-07)† | 6.04E-01(2.13E-03)† | 7.19E-01(2.24E-03) |
| | 200 | 6.02E-01(9.73E-04)† | 5.72E-01(8.02E-04)† | 5.73E-01(8.39E-04)† | 1.51E-01(2.89E-07)† | 6.23E-01(1.97E-03)† | 7.30E-01(2.00E-03) |
| WFG7 | 5 | 6.49E-01(7.45E-05)† | 4.78E-01(3.00E-04)† | 5.96E-01(6.94E-05)† | 5.34E-01(7.04E-05)† | 7.01E-01(2.78E-05)† | 7.80E-01(1.74E-05) |
| | 25 | 6.67E-01(1.68E-05)† | 5.90E-01(5.05E-05)† | 6.23E-01(2.62E-05)† | 6.11E-01(3.92E-05)† | 7.27E-01(3.00E-06)† | 8.04E-01(1.83E-06) |
| | 50 | 6.71E-01(1.39E-05)† | 6.08E-01(2.94E-05)† | 6.27E-01(2.40E-05)† | 6.25E-01(3.28E-05)† | 7.30E-01(2.02E-06)† | 8.06E-01(1.62E-06) |
| | 200 | 6.75E-01(1.17E-05)† | 6.51E-01(1.51E-05)† | 6.31E-01(3.17E-05)† | 6.47E-01(3.12E-05)† | 7.32E-01(1.89E-06)† | 8.08E-01(1.02E-06) |
| WFG8 | 5 | 6.46E-01(4.80E-05)† | 4.79E-01(1.20E-04)† | 5.97E-01(7.20E-05)† | 5.28E-01(1.93E-04)† | 7.05E-01(4.79E-05)† | 7.78E-01(2.43E-05) |
| | 25 | 6.67E-01(1.68E-05)† | 5.90E-01(5.05E-05)† | 6.23E-01(2.62E-05)† | 6.11E-01(5.75E-05)† | 7.28E-01(1.81E-06)† | 8.04E-01(4.00E-06) |
| | 50 | 6.72E-01(1.26E-05)† | 6.09E-01(4.89E-05)† | 6.28E-01(2.55E-05)† | 6.24E-01(4.34E-05)† | 7.30E-01(1.10E-06)† | 8.06E-01(1.31E-06) |
| | 200 | 6.76E-01(1.50E-05)† | 6.50E-01(2.15E-05)† | 6.32E-01(1.74E-05)† | 6.46E-01(2.50E-05)† | 7.32E-01(7.99E-07)† | 8.08E-01(1.12E-06) |
| WFG9 | 5 | 6.37E-01(2.67E-04)‡ | 3.78E-01(1.42E-03)† | 5.71E-01(4.16E-04)‡ | 6.09E-01(1.72E-04)‡ | 5.27E-01(1.01E-02) | 5.50E-01(7.78E-03) |
| | 25 | 6.67E-01(1.68E-05) | 5.90E-01(5.05E-05) | 6.23E-01(2.62E-05) | 6.85E-01(3.76E-05) | 6.17E-01(1.05E-02) | 6.41E-01(1.49E-02) |
| | 50 | 6.78E-01(1.43E-04) | 5.97E-01(5.24E-03) | 6.39E-01(9.38E-05) | 6.95E-01(3.03E-05) | 5.61E-01(1.10E-02)† | 6.30E-01(1.34E-02) |
| | 200 | 6.86E-01(1.73E-04)‡ | 6.46E-01(4.72E-03)‡ | 6.46E-01(1.09E-04)‡ | 6.97E-01(4.80E-05)‡ | 5.59E-01(1.12E-02)† | 5.94E-01(1.14E-02) |
| +-= | | 44/5/3 | 38/11/3 | 45/4/3 | 47/2/3 | 44/2/6 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.6: Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 2.14E+02(6.65E+01)† | 2.75E+01(7.14E+00)‡ | 2.27E+02(3.90E+01)† | 5.53E+02(2.54E+03)† | 4.04E+01(1.34E+01)‡ | 6.52E+01(1.53E+02) |
| | 25 | 1.34E+02(2.21E+01)† | 2.10E+01(2.18E+00)‡ | 1.65E+02(7.64E+01)† | 6.23E+02(2.72E+03)† | 2.15E+01(1.48E+00)‡ | 3.51E+01(2.44E+01) |
| | 50 | 1.19E+02(2.87E+01)† | 2.03E+01(2.05E+00)† | 1.46E+02(1.55E+01)† | 6.80E+02(1.44E+03)† | 1.82E+01(1.99E+00) | 1.93E+01(7.47E+00) |
| | 200 | 1.30E+02(1.32E+02)† | 1.86E+01(1.96E+01)† | 1.46E+02(1.60E+02)† | 8.00E+02(7.95E+02)† | 1.96E+01(1.86E+01)† | 1.19E+01(1.32E+01) |
| F2 | 5 | 5.05E-01(2.64E-03)† | 2.01E-01(8.21E-05)† | 4.47E-01(2.04E-04)† | 3.56E+00(5.92E-02)† | 1.28E-01(3.71E-05)† | 1.04E-01(1.45E-05) |
| | 25 | 3.73E-01(4.86E-04)† | 1.44E-01(3.23E-05)† | 3.97E-01(1.27E-04)† | 3.97E+00(6.52E-02)† | 8.56E-02(7.62E-06)† | 8.20E-02(5.44E-06) |
| | 50 | 3.56E-01(2.95E-04)† | 1.32E-01(3.00E-05)† | 3.98E-01(1.46E-04)† | 3.82E+00(8.00E-02)† | 7.85E-02(6.23E-06)† | 7.66E-02(3.32E-06) |
| | 200 | 3.53E-01(3.82E-01)† | 1.05E-01(1.04E-01)† | 4.29E-01(4.15E-01)† | 4.75E+00(4.59E+00)† | 7.15E-02(7.44E-02) | 7.23E-02(7.14E-02) |
| F3 | 5 | 4.60E+02(3.67E+02)† | 6.98E+01(3.38E+01)‡ | 5.23E+02(3.39E+02)† | 6.92E+02(1.57E+03)† | 1.01E+02(7.05E+01)‡ | 1.63E+02(1.23E+03) |
| | 25 | 2.91E+02(1.19E+02)† | 5.41E+01(2.03E+01)‡ | 3.80E+02(1.28E+02)† | 6.31E+02(9.68E+02)† | 5.55E+01(1.31E+01)‡ | 9.90E+01(1.75E+02) |
| | 50 | 2.57E+02(9.35E+01)† | 5.11E+01(1.44E+01) | 3.42E+02(3.79E+01)† | 6.54E+02(1.16E+03)† | 4.61E+01(3.94E+00)‡ | 5.18E+01(7.31E+01) |
| | 200 | 2.84E+02(2.79E+02)† | 4.42E+01(4.27E+01)† | 3.51E+02(3.67E+02)† | 6.92E+02(6.46E+02)† | 4.57E+01(4.67E+01)† | 3.99E+01(4.00E+01) |
| F4 | 5 | 1.02E+00(1.19E-03)† | 1.74E-01(1.11E-04)† | 6.16E-01(2.76E-04)† | 1.82E-01(4.61E-03) | 2.14E-01(1.63E-04)† | 1.51E-01(3.42E-04) |
| | 25 | 5.68E-01(1.04E-04)† | 1.51E-01(5.29E-05)† | 5.19E-01(1.25E-04)† | 6.93E-02(1.35E-03)‡ | 1.19E-01(2.51E-05)† | 1.05E-01(2.67E-05) |
| | 50 | 5.37E-01(1.52E-04)† | 1.48E-01(2.72E-05)† | 5.26E-01(7.42E-05)† | 6.65E-02(1.08E-03)‡ | 1.09E-01(2.66E-05)† | 9.81E-02(2.21E-05) |
| | 200 | 5.39E-01(5.34E-01)† | 1.28E-01(1.33E-01)† | 5.41E-01(5.50E-01)† | 6.50E-02(8.35E-02)‡ | 1.02E-01(9.93E-02)† | 9.84E-02(1.02E-01) |
| WFG1 | 5 | 5.01E-01(3.00E-04)† | 6.59E-01(1.69E-03)† | 7.27E-01(2.03E-04)† | 1.18E+00(3.20E-03)† | 4.51E-01(1.26E-04)† | 4.10E-01(1.09E-04) |
| | 25 | 3.71E-01(1.04E-04)† | 4.32E-01(4.30E-04)† | 6.09E-01(7.97E-05)† | 1.12E+00(1.00E-03)† | 3.12E-01(6.94E-05)† | 2.67E-01(3.45E-03) |
| | 50 | 3.44E-01(4.41E-05)† | 3.65E-01(3.06E-04)† | 5.82E-01(2.45E-05)† | 1.13E+00(2.06E-04)† | 2.66E-01(3.20E-05)† | 1.93E-01(7.43E-03) |
| | 200 | 3.30E-01(3.35E-01)† | 2.99E-01(2.96E-01)† | 5.78E-01(5.70E-01)† | 1.16E+00(1.14E+00)† | 2.31E-01(2.26E-01)‡ | 2.33E-01(2.23E-01) |
| WFG2 | 5 | 9.36E-01(1.52E-03)† | 6.18E-01(1.71E-03)‡ | 1.03E+00(5.08E-04)† | 3.20E-01(2.95E-05)‡ | 6.57E-01(1.15E-03)‡ | 7.14E-01(4.86E-04) |
| | 25 | 9.09E-01(1.08E-03)† | 4.94E-01(1.05E-03)‡ | 1.04E+00(8.06E-04)† | 3.17E-01(1.09E-05)‡ | 5.49E-01(6.04E-04)‡ | 6.76E-01(2.51E-04) |
| | 50 | 9.46E-01(7.88E-04)† | 4.89E-01(1.09E-03)‡ | 1.07E+00(6.97E-04)† | 3.18E-01(2.11E-05)‡ | 5.49E-01(5.83E-04)‡ | 6.56E-01(2.61E-04) |
| | 200 | 9.54E-01(9.84E-01)† | 5.21E-01(4.77E-01)‡ | 1.12E+00(1.08E+00)† | 3.17E-01(3.11E-01)‡ | 5.25E-01(5.31E-01)‡ | 6.55E-01(6.39E-01) |
| WFG3 | 5 | 1.37E+00(6.76E-03) | 2.08E+00(6.04E-03)† | 1.65E+00(2.82E-03)† | 1.31E+00(4.37E-04)‡ | 1.58E+00(1.69E-03)† | 1.37E+00(1.35E-02) |
| | 25 | 1.15E+00(1.57E-02)‡ | 1.98E+00(3.98E-03)† | 1.62E+00(1.57E-03)† | 1.26E+00(2.19E-04)‡ | 1.58E+00(1.94E-03) | 1.57E+00(1.11E-02) |
| | 50 | 1.36E+00(1.69E-02)‡ | 1.96E+00(5.28E-03)† | 1.64E+00(1.93E-03) | 1.26E+00(1.31E-04)‡ | 1.64E+00(1.62E-03) | 1.63E+00(8.99E-03) |
| | 200 | 1.66E+00(1.64E+00) | 2.04E+00(2.01E+00)† | 1.69E+00(1.65E+00) | 1.26E+00(1.26E+00)‡ | 1.65E+00(1.66E+00) | 1.36E+00(1.70E+00) |
| WFG4 | 5 | 5.22E-01(1.18E-04)† | 5.20E-01(1.87E-04)† | 6.30E-01(1.07E-04)† | 4.22E-01(3.38E-04)† | 3.48E-01(5.51E-05)† | 3.19E-01(7.15E-05) |
| | 25 | 4.97E-01(4.10E-05)† | 3.81E-01(4.91E-05)† | 5.74E-01(5.22E-05)† | 3.76E-01(1.13E-04)† | 2.48E-01(5.19E-06)‡ | 2.72E-01(3.79E-05) |
| | 50 | 4.91E-01(5.63E-05)† | 3.55E-01(3.55E-05)† | 5.68E-01(3.88E-05)† | 3.76E-01(1.54E-04)† | 2.36E-01(5.46E-06)‡ | 2.68E-01(1.58E-05) |
| | 200 | 4.96E-01(4.77E-01)† | 3.18E-01(3.17E-01)† | 5.44E-01(5.57E-01)† | 3.67E-01(3.78E-01)† | 2.14E-01(2.13E-01)‡ | 2.58E-01(2.64E-01) |
| WFG5 | 5 | 5.48E-01(2.14E-04)† | 5.17E-01(2.46E-04)† | 7.24E-01(1.04E-04)† | 8.01E-02(9.39E-06)‡ | 3.52E-01(9.18E-05)‡ | 3.68E-01(1.12E-04) |
| | 25 | 5.12E-01(9.69E-05)† | 3.97E-01(8.79E-05)† | 6.50E-01(8.45E-05)† | 6.41E-02(6.40E-06)‡ | 2.80E-01(6.86E-06)‡ | 3.18E-01(1.49E-05) |
| | 50 | 5.06E-01(6.00E-05)† | 3.68E-01(8.45E-05)† | 6.37E-01(4.22E-05)† | 6.29E-02(6.15E-06)‡ | 2.73E-01(5.32E-06)‡ | 3.21E-01(4.53E-05) |
| | 200 | 4.95E-01(5.04E-01)† | 3.35E-01(3.34E-01)† | 6.18E-01(6.34E-01)† | 5.75E-02(6.38E-02)‡ | 2.71E-01(2.69E-01)‡ | 3.25E-01(3.28E-01) |
| WFG6 | 5 | 7.57E-01(4.50E-03)† | 7.73E-01(3.67E-03)† | 9.48E-01(4.03E-03)† | 1.16E-01(2.33E-05)‡ | 7.81E-01(1.30E-02)† | 6.34E-01(1.26E-02) |
| | 25 | 6.71E-01(5.07E-03)† | 5.56E-01(3.38E-03) | 8.12E-01(3.99E-03)† | 1.19E-01(5.86E-06)‡ | 6.65E-01(1.13E-02)† | 5.49E-01(1.53E-02) |
| | 50 | 6.55E-01(6.17E-03)† | 5.24E-01(3.47E-03) | 8.12E-01(4.88E-03)† | 1.20E-01(2.50E-06)‡ | 5.93E-01(1.18E-02) | 5.63E-01(1.71E-02) |
| | 200 | 7.67E-01(6.59E-01)† | 5.74E-01(5.01E-01) | 8.92E-01(8.02E-01)† | 1.24E-01(1.25E-01)‡ | 5.00E-01(4.40E-01) | 5.70E-01(6.46E-01) |
| WFG7 | 5 | 6.09E-01(9.82E-05)† | 6.71E-01(3.63E-03)† | 8.05E-01(1.27E-04)† | 2.69E-01(1.25E-04)‡ | 4.08E-01(6.52E-05)† | 3.44E-01(4.73E-05) |
| | 25 | 5.56E-01(4.47E-05)† | 4.95E-01(2.48E-04)† | 7.32E-01(5.07E-05)† | 1.54E-01(2.57E-05)‡ | 2.89E-01(8.37E-06)† | 2.84E-01(1.51E-05) |
| | 50 | 5.50E-01(3.17E-05)† | 4.46E-01(9.64E-05)† | 7.25E-01(2.23E-05)† | 1.32E-01(2.57E-05)‡ | 2.74E-01(5.45E-06) | 2.73E-01(1.93E-05) |
| | 200 | 5.31E-01(5.50E-01)† | 3.94E-01(3.85E-01)† | 7.22E-01(7.14E-01)† | 1.15E-01(1.22E-01)‡ | 2.53E-01(2.53E-01)‡ | 2.58E-01(2.50E-01) |
| WFG8 | 5 | 6.12E-01(1.16E-04)† | 6.85E-01(1.81E-03)† | 8.06E-01(1.28E-04)† | 2.69E-01(1.62E-04)‡ | 4.04E-01(7.38E-05)† | 3.43E-01(2.97E-05) |
| | 25 | 5.56E-01(4.47E-05)† | 4.95E-01(2.48E-04)† | 7.32E-01(5.07E-05)† | 1.53E-01(2.17E-05)‡ | 2.88E-01(5.31E-06)† | 2.85E-01(2.34E-05) |
| | 50 | 5.48E-01(3.19E-05)† | 4.46E-01(1.69E-04)† | 7.24E-01(3.33E-05)† | 1.32E-01(5.94E-06)‡ | 2.74E-01(5.38E-06) | 2.74E-01(2.32E-05) |
| | 200 | 5.38E-01(5.47E-01)† | 3.87E-01(3.91E-01)† | 7.07E-01(7.03E-01)† | 1.22E-01(1.13E-01)‡ | 2.53E-01(2.56E-01)† | 2.52E-01(2.43E-01) |
| WFG9 | 5 | 7.01E-01(1.16E-03)‡ | 9.02E-01(5.03E-03) | 8.93E-01(3.14E-03) | 1.66E-01(7.38E-05)‡ | 8.90E-01(4.60E-02) | 8.57E-01(2.42E-02) |
| | 25 | 5.56E-01(4.47E-05) | 4.95E-01(2.48E-04) | 7.32E-01(5.07E-05) | 1.24E-01(1.27E-05)‡ | 6.10E-01(3.56E-02)† | 6.40E-01(4.40E-02) |
| | 50 | 6.07E-01(5.54E-04) | 5.18E-01(1.74E-02)‡ | 7.13E-01(3.01E-04) | 1.16E-01(5.77E-06)‡ | 7.11E-01(3.63E-02) | 6.50E-01(3.84E-02) |
| | 200 | 6.21E-01(5.91E-01)‡ | 4.70E-01(3.94E-01)‡ | 7.50E-01(6.92E-01)‡ | 1.18E-01(1.11E-01)‡ | 7.61E-01(8.64E-01) | 9.14E-01(5.48E-01) |
| +-= | | 44/4/4 | 38/8/6 | 46/1/5 | 20/31/1 | 22/18/12 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

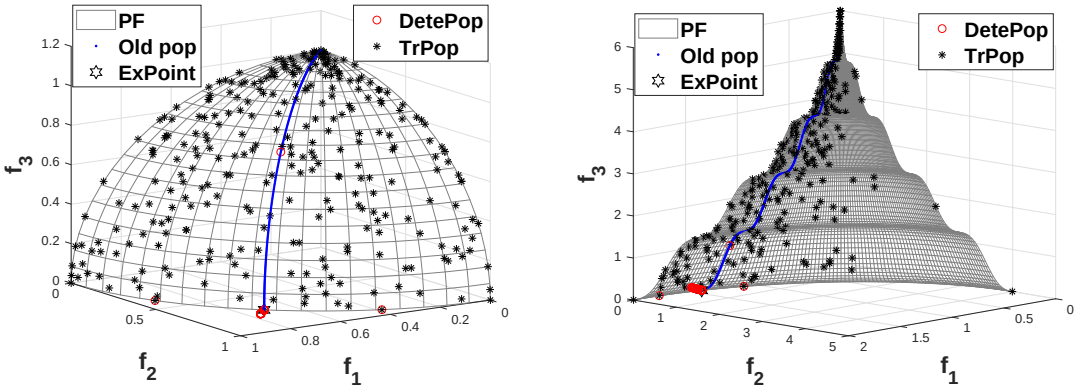
Table 5.7: Mean and standard deviation values of MMS metric for solutions obtained by all compared algorithms in the first generation right after changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 1.00E+00(0.00E+00)‡ | 9.72E-01(3.40E-04) | 9.95E-01(1.00E-04) | 9.28E-01(3.43E-04)† | 1.00E+00(7.00E-08) | 9.62E-01(5.00E-03) |
| | 25 | 1.00E+00(3.93E-06)‡ | 9.04E-01(2.63E-04)† | 1.00E+00(1.71E-07)‡ | 9.43E-01(3.67E-04)† | 6.31E-01(2.27E-01)† | 9.90E-01(5.30E-04) |
| | 50 | 9.99E-01(1.14E-05)† | 9.09E-01(1.96E-05)† | 1.00E+00(6.42E-08)‡ | 9.42E-01(3.63E-04)† | 9.48E-01(4.12E-04)† | 9.99E-01(2.09E-06) |
| | 200 | 9.49E-01(6.56E-05)† | 9.09E-01(5.95E-07)† | 9.96E-01(6.04E-05)† | 9.46E-01(2.76E-04)† | 9.16E-01(9.93E-05)† | 1.00E+00(2.27E-07) |
| F2 | 5 | 9.46E-01(3.10E-04)† | 9.93E-01(9.87E-06)† | 9.93E-01(1.40E-05)† | 9.01E-01(1.80E-03)† | 9.84E-01(5.47E-05)† | 9.96E-01(1.64E-05) |
| | 25 | 9.52E-01(1.16E-05)† | 9.69E-01(5.92E-06)† | 9.97E-01(3.78E-06) | 8.43E-01(7.93E-04)† | 6.15E-01(2.15E-01)† | 9.97E-01(1.36E-05) |
| | 50 | 9.61E-01(6.63E-06)† | 9.62E-01(4.08E-06)† | 9.98E-01(2.19E-06) | 7.50E-01(1.23E-03)† | 9.48E-01(4.15E-06)† | 9.98E-01(5.16E-06) |
| | 200 | 9.65E-01(8.20E-06)† | 9.49E-01(1.19E-06)† | 9.97E-01(3.69E-06)† | 6.33E-01(1.41E-03)† | 9.45E-01(1.94E-06)† | 9.99E-01(1.81E-06) |
| F3 | 5 | 1.00E+00(0.00E+00)‡ | 9.94E-01(8.04E-05) | 1.00E+00(0.00E+00)‡ | 8.57E-01(1.06E-03)† | 1.00E+00(8.39E-08) | 9.45E-01(7.27E-03) |
| | 25 | 1.00E+00(0.00E+00)‡ | 9.80E-01(4.91E-05)‡ | 1.00E+00(5.18E-11)‡ | 8.65E-01(5.19E-04)† | 6.42E-01(2.34E-01)† | 9.74E-01(1.61E-03) |
| | 50 | 9.97E-01(2.14E-05) | 9.52E-01(7.20E-06)† | 1.00E+00(1.98E-09)‡ | 8.42E-01(4.86E-04)† | 9.76E-01(1.22E-04)† | 1.00E+00(5.77E-08) |
| | 200 | 9.67E-01(2.56E-05)† | 9.43E-01(3.63E-07)† | 9.81E-01(2.47E-04)† | 7.88E-01(1.74E-03)† | 9.48E-01(4.56E-05)† | 1.00E+00(4.74E-07) |
| F4 | 5 | 9.83E-01(1.30E-04)‡ | 9.24E-01(2.34E-04)‡ | 9.70E-01(1.71E-04)‡ | 9.40E-01(2.26E-04)‡ | 9.46E-01(5.91E-05)‡ | 8.94E-01(1.35E-03) |
| | 25 | 9.22E-01(6.87E-05)‡ | 9.34E-01(1.30E-04)‡ | 9.67E-01(2.15E-04)‡ | 9.31E-01(1.64E-04)‡ | 6.07E-01(2.09E-01)† | 8.77E-01(5.43E-04) |
| | 50 | 9.45E-01(1.97E-05)‡ | 9.40E-01(2.19E-05)‡ | 9.72E-01(1.80E-04)‡ | 9.34E-01(6.69E-05)‡ | 9.42E-01(5.18E-12)‡ | 8.76E-01(3.82E-04) |
| | 200 | 9.48E-01(2.56E-05)‡ | 9.42E-01(3.23E-16)‡ | 9.73E-01(1.65E-04)‡ | 9.35E-01(4.63E-05)‡ | 9.42E-01(3.91E-12)‡ | 8.78E-01(1.06E-03) |
| WFG1 | 5 | 8.18E-01(2.47E-04)† | 5.84E-01(9.71E-04)† | 7.66E-01(1.96E-04)† | 4.37E-01(1.45E-04)† | 7.70E-01(4.35E-04)† | 8.59E-01(5.21E-04) |
| | 25 | 7.90E-01(8.42E-05)† | 5.14E-01(3.22E-04)† | 8.07E-01(7.24E-05)† | 4.52E-01(8.77E-05)† | 4.87E-01(1.35E-01)† | 9.05E-01(2.46E-04) |
| | 50 | 8.27E-01(5.41E-05)† | 4.83E-01(3.24E-04)† | 8.52E-01(6.29E-05)† | 4.58E-01(6.27E-05)† | 7.83E-01(1.31E-04)† | 9.39E-01(3.68E-05) |
| | 200 | 8.83E-01(8.47E-06)† | 4.79E-01(1.96E-04)† | 9.17E-01(2.71E-05)† | 4.58E-01(3.07E-05)† | 8.69E-01(1.94E-05)† | 9.57E-01(2.51E-05) |
| WFG2 | 5 | 7.69E-01(7.64E-05)† | 5.12E-01(1.46E-03)† | 8.48E-01(1.00E-04)† | 1.83E-01(6.51E-06)† | 8.16E-01(1.15E-04)† | 9.57E-01(2.88E-04) |
| | 25 | 8.57E-01(3.40E-05)† | 4.70E-01(7.10E-04)† | 9.08E-01(6.56E-05)† | 1.86E-01(3.89E-06)† | 5.48E-01(1.71E-01)† | 9.77E-01(9.82E-05) |
| | 50 | 8.70E-01(2.59E-05)† | 4.78E-01(7.22E-04)† | 9.13E-01(6.76E-05)† | 1.85E-01(3.72E-06)† | 8.60E-01(7.17E-05)† | 9.77E-01(7.14E-05) |
| | 200 | 8.75E-01(2.56E-05)† | 4.91E-01(4.94E-04)† | 9.18E-01(3.51E-05)† | 1.87E-01(1.23E-06)† | 8.62E-01(6.17E-05)† | 9.77E-01(5.01E-05) |
| WFG3 | 5 | 8.42E-01(4.17E-04)‡ | 7.41E-01(3.96E-03)† | 7.71E-01(2.54E-04)† | 4.71E-01(4.42E-05)† | 7.85E-01(5.02E-04)† | 8.24E-01(7.82E-04) |
| | 25 | 9.08E-01(2.21E-04)‡ | 8.68E-01(1.23E-03) | 8.88E-01(1.62E-04)‡ | 4.70E-01(1.53E-05)† | 5.58E-01(1.77E-01)† | 8.70E-01(5.76E-04) |
| | 50 | 9.18E-01(2.04E-04)‡ | 8.85E-01(9.60E-04) | 9.00E-01(1.92E-04)‡ | 4.69E-01(1.10E-05)† | 8.77E-01(2.92E-04) | 8.83E-01(3.79E-04) |
| | 200 | 9.17E-01(2.75E-04)‡ | 9.09E-01(4.50E-04) | 9.04E-01(2.38E-04) | 4.71E-01(1.92E-05)† | 8.90E-01(2.58E-04)† | 9.02E-01(2.23E-04) |
| WFG4 | 5 | 9.03E-01(7.52E-05)† | 8.76E-01(7.51E-04)† | 9.45E-01(3.41E-05)† | 8.40E-01(2.88E-05)† | 9.51E-01(5.92E-05)† | 9.93E-01(8.74E-06) |
| | 25 | 9.43E-01(6.92E-06)† | 9.16E-01(3.44E-04)† | 9.66E-01(2.62E-05)† | 8.38E-01(7.80E-06)† | 6.08E-01(2.10E-01)† | 9.99E-01(9.29E-07) |
| | 50 | 9.47E-01(1.01E-05)† | 9.27E-01(1.17E-04)† | 9.63E-01(1.85E-05)† | 8.38E-01(7.81E-06)† | 9.41E-01(3.94E-07)† | 9.99E-01(4.80E-07) |
| | 200 | 9.50E-01(1.11E-05)† | 9.34E-01(9.40E-05)† | 9.64E-01(1.43E-05)† | 8.36E-01(1.87E-06)† | 9.40E-01(3.61E-07)† | 9.99E-01(4.29E-07) |
| WFG5 | 5 | 9.18E-01(1.03E-04)† | 9.00E-01(3.35E-04)† | 9.51E-01(1.92E-05)† | 8.35E-01(2.12E-05)† | 9.71E-01(1.79E-05)† | 9.82E-01(7.17E-06) |
| | 25 | 9.35E-01(1.47E-05)† | 9.08E-01(1.30E-04)† | 9.64E-01(1.10E-05)† | 8.21E-01(5.82E-06)† | 6.05E-01(2.08E-01)† | 9.88E-01(2.28E-06) |
| | 50 | 9.40E-01(2.19E-05)† | 9.09E-01(9.22E-05)† | 9.67E-01(5.68E-06)† | 8.19E-01(9.54E-06)† | 9.36E-01(3.86E-06)† | 9.89E-01(7.59E-07) |
| | 200 | 9.42E-01(1.65E-05)† | 9.08E-01(1.27E-04)† | 9.69E-01(9.92E-06)† | 8.15E-01(7.20E-06)† | 9.35E-01(1.81E-07)† | 9.90E-01(5.74E-07) |
| WFG6 | 5 | 8.58E-01(1.53E-04)† | 8.01E-01(8.74E-04)† | 9.19E-01(2.74E-04)† | 3.91E-01(9.84E-06)† | 9.12E-01(4.02E-04)† | 9.49E-01(5.78E-04) |
| | 25 | 9.08E-01(2.34E-04)† | 8.76E-01(3.38E-04)† | 9.42E-01(2.22E-04)† | 3.89E-01(1.92E-06)† | 5.67E-01(1.83E-01)† | 9.68E-01(4.66E-04) |
| | 50 | 9.16E-01(2.84E-04)† | 8.90E-01(2.94E-04)† | 9.42E-01(2.28E-04)† | 3.88E-01(1.97E-06)† | 8.93E-01(3.92E-04)† | 9.66E-01(4.21E-04) |
| | 200 | 9.16E-01(2.90E-04)† | 9.00E-01(2.82E-04)† | 9.45E-01(2.34E-04)† | 3.88E-01(1.11E-06)† | 8.99E-01(4.08E-04)† | 9.71E-01(3.04E-04) |
| WFG7 | 5 | 9.54E-01(4.39E-05)† | 8.59E-01(1.36E-03)† | 9.61E-01(3.04E-05)† | 9.33E-01(3.98E-05)† | 9.63E-01(2.40E-05)† | 9.87E-01(8.06E-06) |
| | 25 | 9.46E-01(1.47E-05)† | 9.36E-01(9.15E-05)† | 9.74E-01(1.65E-05)† | 9.55E-01(4.97E-05)† | 6.07E-01(2.09E-01)† | 9.94E-01(2.47E-06) |
| | 50 | 9.46E-01(1.35E-05)† | 9.37E-01(8.20E-05)† | 9.77E-01(1.94E-05)† | 9.51E-01(2.91E-05)† | 9.39E-01(2.18E-06)† | 9.94E-01(3.26E-06) |
| | 200 | 9.46E-01(9.65E-06)† | 9.47E-01(5.28E-05)† | 9.81E-01(7.68E-06)† | 9.44E-01(6.81E-05)† | 9.40E-01(7.95E-07)† | 9.95E-01(2.88E-06) |
| WFG8 | 5 | 9.55E-01(1.74E-05)† | 8.66E-01(4.93E-04)† | 9.62E-01(2.12E-05)† | 9.28E-01(5.81E-05)† | 9.64E-01(2.89E-05)† | 9.86E-01(1.37E-05) |
| | 25 | 9.46E-01(1.47E-05)† | 9.36E-01(9.15E-05)† | 9.74E-01(1.65E-05)† | 9.54E-01(3.65E-05)† | 6.07E-01(2.09E-01)† | 9.94E-01(3.54E-06) |
| | 50 | 9.46E-01(1.00E-05)† | 9.40E-01(7.89E-05)† | 9.78E-01(1.73E-05)† | 9.51E-01(4.30E-05)† | 9.39E-01(1.23E-06)† | 9.95E-01(1.86E-06) |
| | 200 | 9.47E-01(7.13E-06)† | 9.47E-01(6.55E-05)† | 9.80E-01(1.12E-05)† | 9.45E-01(5.48E-05)† | 9.40E-01(7.76E-07)† | 9.94E-01(4.67E-06) |
| WFG9 | 5 | 9.45E-01(7.51E-05)‡ | 8.21E-01(8.50E-04)† | 9.45E-01(1.18E-04)‡ | 9.56E-01(3.33E-05)‡ | 9.05E-01(1.07E-03) | 9.15E-01(1.03E-03) |
| | 25 | 9.46E-01(1.47E-05) | 9.36E-01(9.15E-05) | 9.74E-01(1.65E-05) | 9.58E-01(2.93E-05) | 5.92E-01(2.00E-01)† | 9.44E-01(1.80E-03) |
| | 50 | 9.53E-01(6.75E-05) | 9.19E-01(7.20E-04) | 9.67E-01(4.49E-05) | 9.57E-01(6.62E-05) | 9.01E-01(1.54E-03)† | 9.42E-01(1.59E-03) |
| | 200 | 9.55E-01(6.76E-05)‡ | 9.31E-01(7.23E-04) | 9.73E-01(3.14E-05)‡ | 9.51E-01(6.47E-05)‡ | 8.93E-01(1.62E-03)† | 9.28E-01(1.38E-03) |
| +-= | | 35/14/3 | 40/4/8 | 34/12/6 | 45/5/2 | 46/2/4 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

#### 5.4.1.3 Why Does Knowledge Transfer Usually Get Better Solution Quality Right after Changes?

In this section, two examples are presented to elaborate the reason why the proposed PS expansion/contraction works well on most problems.

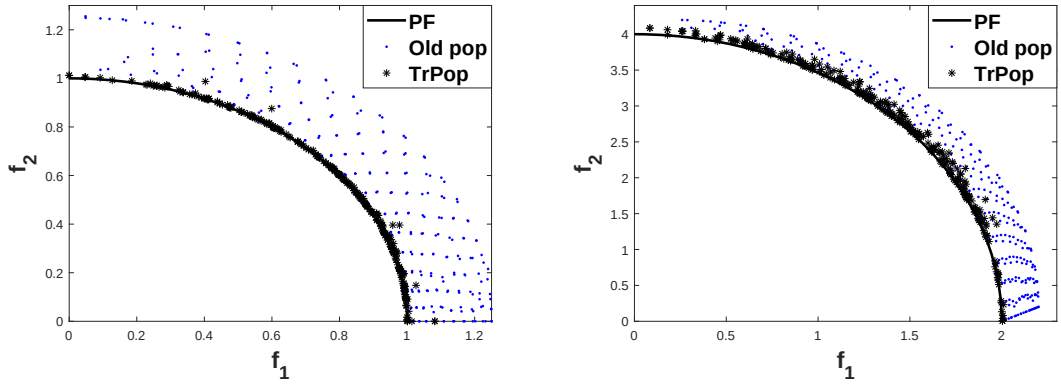



(a) F2 (b) WFG1

Figure 5.7: The distribution of the old population ('Old pop'), the extreme point ('ExPoint'), the detective population ('DetePop') and transferred population ('TrPop') via PS expansion on F2 and WFG1 at the first generation when increasing the NObj from 2 to 3.

As shown in Figure 5.7, the distributions of the old population, the detective population and transferred population via PS expansion on F2 and WFG1 is presented when increasing the NObj from 2 to 3. It is clear that nondominated solutions in the detective population are still nondominated by the selected extreme point. Following steps 5 to 10 in Algorithm 5.2, there are only two solutions in $P_{var}$, which are located in the areas away from that of the old population. Then, each of those two solutions is regarded as the ending point of the expansion direction, together with the extreme point as the starting point of the direction. Therefore, when evenly selecting solutions from the old population to conduct the PS expansion, almost all areas of F2 can be covered right after the change. Even for WFG1, a large area of the PF is covered by the transferred solutions. In addition, it is clear that some of the transferred solutions are able to reach the boundary of the PF.

Figure 5.8 presents the distributions of the old population and the transferred solutions via PS contraction on F2 and WFG1 at the first generation when decreasing the NObj from 3 to 2 in the changing NObj sequence of firstly decreasing from 7 to 2 and then increasing from 2 to 7. It is clear that on those two problems the transferred population via PS contraction has better convergence and diversity than the old population.



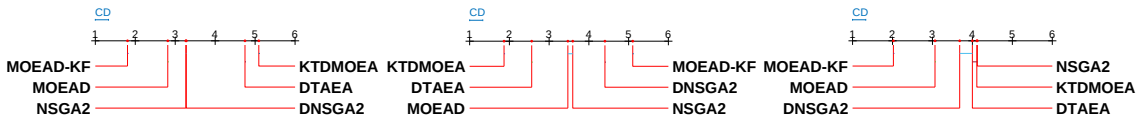

(a) F2 (b) WFG6

Figure 5.8: The distribution of the old population ('Old pop') and transferred population ('TrPop') via PS contraction on F2 and WFG1 at the first generation when decreasing the NObj from 3 to 2.

#### 5.4.1.4 Why Doesn't Knowledge Transfer Sometimes Get Better Solution Quality after Changes?

It has been found from Tables 5.2-5.7 that KTDMOEA performs worse than others on F4 and WFG9 in the above two changing sequences '2-7-2 one by one' and '7-2-7 one by one'. The reasons behind this result are analyzed as follows. For F4 with an extremely strong bias in the fitness landscape, when using the PS expansion to transfer solutions, the expansion directions cannot be found as all generated solutions around one extreme point are still close to the boundary point due to the strong bias. As for WFG9 with very complex problem features in the fitness landscape including multimodality, nonseparability, deceptiveness and bias, it is difficult for the used genetic operators (SBX and PM) in KTDMOEA to reach the global optima, while the DE operator in MOEAD-KF is able to do that. Therefore, KTDMOEA does not get better results than MOEAD-KF.

Note that when the NObj decreases from 7 to 2 and then increases from 2 to 7, KTDMOEA performs worse on WFG3 for all cases. The reason might be that when increasing the NObj, transferred solutions via PS expansion cannot increase population diversity, since for WFG3 with degenerated PF, the expanded solutions are still in the degenerated PF, without increase in the diversity. In addition, KTDMOEA performs worse on F1 and F3 at all cases except for large value of $\tau_t$ and increasing the NObj when the NObj decreasing from 7 to 2 and then increasing from 2 to 7. The reason might be that when $\tau_t$ is large, KTDMOEA ran for enough generations to reach convergence before the increase in the NObj, which is beneficial for the PS expansion.




(a) MHV (b) MGD (c) MMS

Figure 5.9: Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by 6 algorithms in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

### 5.4.2 How Does Knowledge Transfer Help Optimization?

In order to verify whether the proposed KTDMOEA can find solutions with better convergence and diversity in the last generation after optimization against all other algorithms, the solution quality of all compared algorithms after optimization and before the next change is compared.

#### 5.4.2.1 NObj Increasing from 2 to 7 And then Decreasing from 7 to 2

Figure 5.9 presents the Nemenyi post-tests results among MHV, MGD and MMS of obtained solutions at the last generation after optimization by 6 algorithms. Friedman test detects significant differences in average value for MHV, MGD and MMS with a p-value of 6.83E-215, 6.22E-160, and 4.27E-164, respectively. Tables 5.8-5.10 show mean and standard deviation values of MHV, MGD and MMS of obtained solutions in the first generation after changes averaged across 10 environmental changes in the sequence of changes as '7-2-7 one by one'.

Overall, it can be seen from those statistical test results that KTDMOEA performs significantly better than or the same as the other approaches. Specifically, it is clear from the Friedman ranking results in Figure 5.9 that KTDMOEA gets significantly best results among all compared algorithms regarding MHV and MGD values. It is the equal best, together with DTAEA and NSGA2, regarding the MS value. These three algorithms outperforms other algorithms regarding the MMS value. The statistical results show that the proposed knowledge transfer is able to help the optimization, which achieves better convergence and at least similar diversity compared to the start-of-the-arts when the NObj increasing from 2 to 7 and then decreasing from 7 to 2, under all frequencies of changes on most problems.

It can be seen from Table 5.8 that KTDMOEA results in a significantly worse mean MHV value than DTAEA on F4. And this is caused by the fact that KTDMOEA has worse MHV than DTAEA at each stage of increasing and decreasing the NObj, which can be found in Figure 5.10(d). In addition, it can be seen from Tables 5.8-5.10 that KTDMOEA gets significantly worse mean values of MHV and MMS than NSGA2, DNSGA2 and DTAEA on WFG3.

Table 5.8: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms at the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 9.21E-01(3.60E-04)† | 8.86E-01(3.70E-04)† | 9.26E-01(4.64E-04)† | 1.37E-02(4.54E-04)† | 9.71E-01(3.44E-04)† | 9.96E-01(2.39E-11) |
| | 25 | 6.08E-01(2.07E-03)† | 9.88E-01(5.08E-05)† | 5.87E-01(9.39E-04)† | 8.64E-02(3.34E-04)† | 9.75E-01(9.87E-04)† | 9.96E-01(9.75E-10) |
| | 50 | 5.44E-01(1.08E-02)† | 9.96E-01(3.76E-09)† | 4.76E-01(2.17E-03)† | 1.09E-01(3.98E-04)† | 9.92E-01(1.43E-04)† | 9.96E-01(1.07E-07) |
| | 200 | 5.96E-01(1.93E-02)† | 9.96E-01(1.47E-12)‡ | 4.22E-01(1.66E-03)† | 2.16E-01(2.32E-03)† | 9.96E-01(3.57E-09)† | 9.96E-01(5.56E-10) |
| F2 | 5 | 8.81E-01(5.64E-05)† | 9.23E-01(1.58E-05)† | 9.36E-01(3.67E-06)† | 3.94E-01(7.92E-03)† | 9.47E-01(3.57E-06)† | 9.58E-01(2.91E-07) |
| | 25 | 9.38E-01(3.44E-06)† | 9.53E-01(6.22E-07)† | 9.33E-01(1.67E-05)† | 2.80E-01(5.38E-04)† | 9.58E-01(6.69E-08)† | 9.58E-01(1.15E-09) |
| | 50 | 9.33E-01(1.66E-05)† | 9.55E-01(1.04E-07)† | 9.24E-01(1.65E-05)† | 2.70E-01(1.19E-04)† | 9.58E-01(1.21E-09)† | 9.59E-01(2.68E-10) |
| | 200 | 9.23E-01(5.36E-05)† | 9.58E-01(9.94E-09)† | 9.17E-01(5.28E-05)† | 2.72E-01(2.84E-04)† | 9.59E-01(3.37E-10)† | 9.59E-01(6.51E-11) |
| F3 | 5 | 7.74E-01(2.33E-04)† | 7.49E-01(4.18E-04)† | 7.36E-01(3.08E-04)† | 3.94E-01(7.92E-03)† | 8.40E-01(7.00E-04)† | 9.58E-01(2.07E-07) |
| | 25 | 5.71E-01(6.49E-04)† | 9.27E-01(6.77E-05)† | 5.54E-01(5.72E-04)† | 2.08E-02(1.82E-04)† | 8.86E-01(3.76E-03)† | 9.56E-01(1.58E-06) |
| | 50 | 4.61E-01(1.88E-03)† | 9.53E-01(1.60E-06)† | 4.16E-01(9.24E-04)† | 8.53E-02(1.30E-04)† | 9.24E-01(2.30E-03)† | 9.55E-01(2.12E-04) |
| | 200 | 4.53E-01(2.72E-03)† | 9.57E-01(5.36E-08)‡ | 3.96E-01(1.42E-03)† | 1.45E-01(1.31E-03)† | 9.57E-01(8.99E-07)‡ | 9.56E-01(2.08E-04) |
| F4 | 5 | 8.69E-01(1.37E-04) | 8.69E-01(3.12E-04) | 9.12E-01(9.36E-05)† | 8.97E-01(6.40E-05)† | 9.41E-01(2.14E-05)‡ | 8.29E-01(5.24E-03) |
| | 25 | 9.10E-01(1.07E-04)† | 9.36E-01(1.02E-04)† | 8.95E-01(7.57E-05) | 9.39E-01(1.24E-05)† | 9.58E-01(3.12E-08)‡ | 8.89E-01(4.17E-04) |
| | 50 | 8.81E-01(1.28E-04) | 9.50E-01(4.41E-05)† | 8.83E-01(3.13E-05) | 9.44E-01(2.77E-05)† | 9.58E-01(7.78E-10)‡ | 8.86E-01(9.96E-04) |
| | 200 | 8.65E-01(3.68E-05) | 9.58E-01(2.58E-08)† | 8.61E-01(9.69E-06) | 9.53E-01(1.21E-05)† | 9.59E-01(2.36E-10)‡ | 8.46E-01(3.53E-03) |
| WFG1 | 5 | 9.25E-01(4.37E-05)† | 8.42E-01(5.71E-05)† | 9.16E-01(7.24E-05)† | 7.99E-01(4.05E-04)† | 9.25E-01(3.94E-05)† | 9.68E-01(2.02E-06) |
| | 25 | 9.51E-01(1.14E-05)† | 8.86E-01(2.85E-05)† | 9.47E-01(1.14E-05)† | 8.10E-01(1.74E-08)† | 9.54E-01(1.21E-05)† | 9.70E-01(1.18E-06) |
| | 50 | 9.59E-01(2.15E-06)† | 9.05E-01(2.05E-05)† | 9.57E-01(2.65E-06)† | 8.10E-01(1.92E-08)† | 9.63E-01(2.93E-06)† | 9.70E-01(6.52E-07) |
| | 200 | 9.68E-01(1.42E-07)† | 9.30E-01(7.70E-06)† | 9.68E-01(1.21E-07)† | 8.10E-01(3.44E-08)† | 9.71E-01(1.01E-07)‡ | 9.70E-01(3.50E-07) |
| WFG2 | 5 | 9.33E-01(4.41E-05)† | 8.76E-01(3.84E-05)† | 9.49E-01(2.43E-06)† | 2.41E-01(9.24E-08)† | 9.55E-01(1.49E-06)† | 9.68E-01(7.02E-09) |
| | 25 | 9.63E-01(5.44E-07)† | 9.28E-01(5.60E-06)† | 9.63E-01(1.76E-07)† | 2.42E-01(2.11E-09)† | 9.67E-01(2.07E-08)† | 9.68E-01(2.36E-09) |
| | 50 | 9.65E-01(6.51E-08)† | 9.38E-01(2.35E-06)† | 9.65E-01(5.54E-08)† | 2.42E-01(1.38E-09)† | 9.68E-01(2.64E-09)† | 9.68E-01(1.28E-09) |
| | 200 | 9.65E-01(3.23E-08)† | 9.50E-01(4.80E-07)† | 9.65E-01(1.53E-08)† | 2.42E-01(7.11E-10)† | 9.68E-01(2.59E-10)‡ | 9.68E-01(3.41E-10) |
| WFG3 | 5 | 6.91E-01(1.50E-07)† | 6.33E-01(4.28E-05)† | 6.83E-01(1.20E-05)† | 2.50E-01(1.75E-07)† | 6.88E-01(4.64E-05)† | 6.99E-01(7.94E-06) |
| | 25 | 7.03E-01(1.49E-05)‡ | 6.84E-01(1.10E-05)† | 7.11E-01(1.15E-06)‡ | 2.53E-01(1.65E-08)† | 6.94E-01(1.06E-05)‡ | 6.89E-01(7.10E-04) |
| | 50 | 7.10E-01(4.63E-06)‡ | 6.93E-01(7.56E-06)† | 7.13E-01(9.10E-07)‡ | 2.54E-01(9.14E-09)† | 6.97E-01(6.42E-06)‡ | 6.94E-01(4.66E-04) |
| | 200 | 7.15E-01(7.91E-07)‡ | 7.03E-01(3.23E-06)‡ | 7.16E-01(6.77E-07)‡ | 2.54E-01(2.02E-09)† | 6.99E-01(1.52E-06) | 6.99E-01(1.88E-06) |
| WFG4 | 5 | 6.45E-01(2.84E-04)† | 6.18E-01(1.64E-04)† | 6.71E-01(4.74E-05)† | 2.92E-01(1.38E-04)† | 7.50E-01(3.79E-05)† | 8.20E-01(8.99E-07) |
| | 25 | 7.45E-01(2.25E-05)† | 7.00E-01(2.49E-05)† | 7.21E-01(3.57E-05)† | 3.47E-01(2.85E-05)† | 8.28E-01(5.53E-07)† | 8.30E-01(2.10E-07) |
| | 50 | 7.49E-01(2.17E-05)† | 7.25E-01(1.48E-05)† | 7.31E-01(2.86E-05)† | 3.48E-01(9.82E-06)† | 8.32E-01(1.89E-08)† | 8.32E-01(2.32E-08) |
| | 200 | 7.63E-01(3.69E-05)† | 7.61E-01(6.02E-06)† | 7.52E-01(1.58E-05)† | 3.47E-01(1.05E-06)† | 8.32E-01(7.75E-09)† | 8.32E-01(1.28E-08) |
| WFG5 | 5 | 5.76E-01(2.71E-04)† | 6.19E-01(3.95E-05)† | 6.60E-01(3.54E-05)† | 5.15E-01(1.57E-05)† | 7.30E-01(1.89E-05)† | 7.67E-01(3.91E-06) |
| | 25 | 7.05E-01(3.01E-05)† | 7.00E-01(1.23E-05)† | 7.00E-01(2.56E-05)† | 5.35E-01(3.14E-06)† | 7.85E-01(9.78E-07)† | 7.86E-01(1.48E-06) |
| | 50 | 7.14E-01(7.86E-06)† | 7.16E-01(5.54E-06)† | 7.08E-01(1.76E-05)† | 5.32E-01(2.28E-06)† | 7.93E-01(5.47E-07)‡ | 7.90E-01(4.27E-07) |
| | 200 | 7.19E-01(1.27E-05)† | 7.42E-01(1.40E-06)† | 7.21E-01(1.54E-05)† | 5.30E-01(1.35E-06)† | 7.99E-01(2.88E-08)‡ | 7.91E-01(1.79E-08) |
| WFG6 | 5 | 5.26E-01(3.36E-04)† | 5.17E-01(3.26E-04)† | 6.24E-01(5.36E-04)† | 1.43E-01(2.46E-06)† | 6.71E-01(3.94E-04)† | 7.46E-01(5.79E-04) |
| | 25 | 6.90E-01(1.63E-04)† | 6.85E-01(1.95E-05)† | 7.11E-01(1.71E-04)† | 1.46E-01(1.65E-05)† | 7.56E-01(1.77E-03)† | 7.61E-01(4.48E-04) |
| | 50 | 7.25E-01(7.31E-05)† | 7.16E-01(1.23E-05)† | 7.30E-01(1.04E-04)† | 1.48E-01(1.13E-05)† | 7.57E-01(1.54E-03)‡ | 7.37E-01(8.33E-04) |
| | 200 | 7.51E-01(4.16E-05)† | 7.60E-01(5.11E-06)† | 7.51E-01(2.74E-05)† | 1.49E-01(1.02E-05)† | 7.89E-01(9.68E-04)‡ | 7.71E-01(4.10E-04) |
| WFG7 | 5 | 7.37E-01(5.68E-05)† | 5.14E-01(2.37E-04)† | 7.16E-01(5.59E-05)† | 5.59E-01(2.07E-04)† | 7.84E-01(1.42E-05)† | 8.24E-01(6.34E-07) |
| | 25 | 7.66E-01(1.56E-05)† | 6.82E-01(2.69E-05)† | 7.59E-01(7.21E-06)† | 7.23E-01(3.10E-05)† | 8.31E-01(1.01E-07)‡ | 8.27E-01(3.03E-04) |
| | 50 | 7.73E-01(5.30E-06)† | 7.22E-01(1.15E-05)† | 7.72E-01(4.30E-06)† | 7.50E-01(1.87E-05)† | 8.32E-01(1.58E-08)† | 8.32E-01(1.73E-08) |
| | 200 | 7.83E-01(1.16E-06)† | 7.73E-01(2.33E-06)† | 7.83E-01(2.00E-06)† | 7.73E-01(6.17E-06)† | 8.33E-01(3.51E-09)† | 8.33E-01(1.55E-09) |
| WFG8 | 5 | 7.37E-01(6.29E-05)† | 5.05E-01(3.22E-04)† | 7.16E-01(3.51E-05)† | 5.59E-01(1.78E-04)† | 7.85E-01(1.85E-05)† | 8.24E-01(9.40E-07) |
| | 25 | 7.66E-01(1.37E-05)† | 6.83E-01(3.27E-05)† | 7.60E-01(1.09E-05)† | 7.21E-01(3.49E-05)† | 8.31E-01(6.00E-08)‡ | 8.28E-01(2.09E-04) |
| | 50 | 7.74E-01(6.88E-06)† | 7.22E-01(1.99E-05)† | 7.72E-01(4.96E-06)† | 7.46E-01(1.84E-05)† | 8.32E-01(1.69E-08)† | 8.32E-01(1.05E-08) |
| | 200 | 7.83E-01(1.51E-06)† | 7.73E-01(3.88E-06)† | 7.83E-01(1.35E-06)† | 7.73E-01(5.69E-06)† | 8.33E-01(1.94E-09)† | 8.33E-01(7.29E-10) |
| WFG9 | 5 | 3.56E-01(2.80E-03)† | 3.45E-01(4.99E-04)† | 4.40E-01(2.57E-05) | 5.62E-01(1.35E-02)‡ | 4.69E-01(6.63E-05)‡ | 4.44E-01(6.20E-04) |
| | 25 | 4.24E-01(7.02E-04) | 4.10E-01(8.64E-05)‡ | 5.47E-01(4.91E-03)‡ | 7.01E-01(6.71E-03)‡ | 4.92E-01(6.23E-05)‡ | 4.34E-01(1.31E-03) |
| | 50 | 4.43E-01(1.19E-03) | 4.27E-01(5.03E-04)† | 6.27E-01(1.57E-04)‡ | 6.95E-01(7.71E-03)‡ | 4.98E-01(6.26E-05)‡ | 4.37E-01(1.26E-03) |
| | 200 | 5.03E-01(6.47E-03)‡ | 4.86E-01(5.21E-03)‡ | 6.58E-01(1.67E-03)‡ | 7.16E-01(4.18E-03)‡ | 5.08E-01(1.12E-04)‡ | 4.75E-01(2.57E-04) |
| +-= | | 43/4/5 | 46/5/1 | 42/6/4 | 48/4/0 | 33/18/1 | 9.25E-01(3.02E-03) |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.9: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

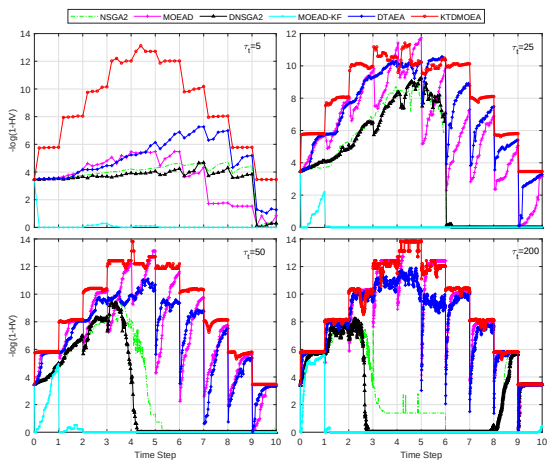
| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 3.39E+00(2.13E+00)† | 6.28E+00(1.04E+00)† | 5.46E+00(1.49E+01)† | 3.53E+02(1.73E+03)† | 9.26E-01(1.03E+00)† | 1.50E-02(1.14E-04) |
| | 25 | 2.60E+01(3.49E+01)† | 7.03E-01(6.81E-03)† | 3.14E+01(3.93E+01)† | 7.05E+02(1.24E+03)† | 1.04E+00(6.01E-01)† | 4.34E-02(2.29E-03) |
| | 50 | 6.46E+01(8.49E+01)† | 6.98E-01(4.52E-03)† | 7.15E+01(6.59E+01)† | 7.79E+02(1.58E+03)† | 2.30E-01(3.67E-02)† | 2.32E-02(5.63E-04) |
| | 200 | 1.10E+02(1.25E+02)† | 7.71E-01(6.15E-01)† | 1.13E+02(1.18E+02)† | 8.71E+02(8.01E+02)† | 1.96E-02(4.43E-02)‡ | 2.48E-02(1.26E-01) |
| F2 | 5 | 6.25E-02(2.51E-06)† | 1.60E-01(7.52E-05)† | 1.60E-01(1.80E-04)† | 3.32E+00(1.32E-01)† | 7.82E-02(3.20E-05)† | 3.96E-02(3.49E-07) |
| | 25 | 1.86E-01(2.06E-04)† | 8.82E-02(4.06E-06)† | 2.79E-01(4.13E-04)† | 6.21E+00(1.00E-01)† | 3.89E-02(6.02E-07)† | 3.38E-02(2.22E-07) |
| | 50 | 2.68E-01(1.73E-04)† | 7.08E-02(2.99E-06)† | 3.16E-01(1.84E-04)† | 6.48E+00(1.08E-01)† | 3.35E-02(2.83E-07)† | 3.20E-02(9.27E-08) |
| | 200 | 3.42E-01(3.60E-01)† | 4.62E-02(4.61E-02)† | 3.53E-01(3.46E-01)† | 6.81E+00(6.80E+00)† | 3.19E-02(3.25E-02)‡ | 3.26E-02(3.26E-02) |
| F3 | 5 | 4.30E+00(3.79E+00)† | 1.63E+01(7.59E+00)† | 9.90E+00(3.24E+01)† | 4.99E+02(1.78E+03)† | 1.79E+00(3.20E+00)† | 8.35E-02(5.89E-03) |
| | 25 | 4.03E+01(8.69E+01)† | 2.60E+00(1.51E-01)† | 5.51E+01(2.69E+02)† | 8.83E+02(2.56E+03)† | 4.19E+00(8.22E+00)† | 3.00E-01(1.37E-01) |
| | 50 | 1.17E+02(3.04E+02)† | 2.58E+00(7.76E-02)† | 1.33E+02(1.70E+02)† | 9.81E+02(1.56E+03)† | 5.40E-01(2.01E-01)† | 8.62E-02(7.98E-03) |
| | 200 | 1.99E+02(2.14E+02)† | 2.51E+00(2.36E+00)† | 2.42E+02(2.71E+02)† | 1.10E+03(1.03E+03)† | 1.42E-01(3.40E-02)† | 3.68E-02(3.14E-02) |
| F4 | 5 | 5.27E-02(4.92E-05)† | 5.15E-02(1.10E-04)† | 2.27E-01(4.20E-04)† | 4.09E-01(7.19E-04)† | 7.69E-02(1.37E-04)† | 2.09E-02(9.38E-05) |
| | 25 | 2.89E-01(6.77E-04)† | 5.39E-02(2.98E-05)† | 4.02E-01(2.79E-04)† | 3.90E-01(9.98E-05)† | 3.37E-02(5.80E-07)† | 2.31E-02(3.55E-06) |
| | 50 | 4.06E-01(3.20E-04)† | 5.57E-02(1.54E-05)† | 4.35E-01(1.41E-04)† | 3.96E-01(9.39E-05)† | 3.09E-02(1.07E-07)† | 2.33E-02(1.16E-05) |
| | 200 | 4.61E-01(4.54E-01)† | 4.38E-02(4.42E-02)† | 4.62E-01(4.58E-01)† | 4.01E-01(4.06E-01)† | 3.38E-02(3.36E-02)† | 2.53E-02(2.21E-02) |
| WFG1 | 5 | 2.16E-01(1.87E-04)† | 4.18E-01(1.66E-04)† | 2.16E-01(3.08E-04)† | 1.70E+00(1.06E-02)† | 2.13E-01(1.59E-04)† | 1.93E-01(3.91E-05) |
| | 25 | 2.71E-01(5.36E-05)† | 3.50E-01(8.82E-05)† | 2.83E-01(4.29E-05)† | 2.00E+00(2.99E-03)† | 2.24E-01(2.83E-05)† | 1.81E-01(2.00E-05) |
| | 50 | 2.78E-01(3.18E-05)† | 2.94E-01(4.58E-05)† | 2.82E-01(2.86E-05)† | 2.05E+00(1.29E-03)† | 2.02E-01(2.28E-05)† | 1.74E-01(1.42E-05) |
| | 200 | 2.81E-01(2.78E-01)† | 2.29E-01(2.23E-01)† | 2.82E-01(2.79E-01)† | 2.11E+00(2.12E+00)† | 1.65E-01(1.65E-01)‡ | 1.75E-01(1.69E-01) |
| WFG2 | 5 | 5.92E-01(1.98E-04)† | 4.25E-01(2.92E-04)‡ | 6.12E-01(3.75E-04)† | 7.10E-01(2.43E-06)† | 5.14E-01(2.61E-04)† | 5.20E-01(2.39E-04) |
| | 25 | 6.84E-01(3.58E-04)† | 3.12E-01(9.84E-06)‡ | 6.79E-01(4.56E-04)† | 7.11E-01(7.43E-07)† | 4.46E-01(5.24E-05)‡ | 4.54E-01(3.96E-05) |
| | 50 | 7.18E-01(6.81E-04)† | 3.06E-01(9.39E-06)‡ | 7.05E-01(3.69E-04)† | 7.13E-01(6.32E-07)† | 4.47E-01(3.00E-05)† | 4.42E-01(1.83E-05) |
| | 200 | 7.32E-01(7.13E-01)† | 3.04E-01(2.98E-01)‡ | 7.35E-01(7.57E-01)† | 7.14E-01(7.14E-01)† | 4.38E-01(4.36E-01) | 4.33E-01(4.34E-01) |
| WFG3 | 5 | 1.48E-02(4.31E-04)‡ | 1.84E+00(1.61E-02)† | 1.49E+00(4.87E-03)† | 7.39E-01(2.84E-05) | 1.28E+00(2.66E-03)† | 7.35E-01(2.34E-02) |
| | 25 | 1.12E+00(4.49E-02)‡ | 2.26E+00(2.83E-05)† | 1.89E+00(8.36E-04)† | 7.52E-01(4.16E-06)‡ | 1.58E+00(9.13E-04)† | 1.48E+00(5.12E-03) |
| | 50 | 1.62E+00(1.30E-02)† | 2.28E+00(7.97E-06)† | 1.95E+00(1.99E-04)† | 7.57E-01(4.76E-06)‡ | 1.68E+00(4.45E-04)† | 1.55E+00(4.07E-03) |
| | 200 | 1.95E+00(1.96E+00)† | 2.29E+00(2.29E+00)† | 1.99E+00(2.00E+00)† | 7.61E-01(7.59E-01)‡ | 1.72E+00(1.72E+00)† | 1.66E+00(1.65E+00) |
| WFG4 | 5 | 3.96E-01(7.32E-05)† | 4.72E-01(4.56E-04)† | 5.36E-01(9.44E-05)† | 7.34E-01(3.60E-04)† | 3.52E-01(1.12E-04)† | 2.88E-01(5.65E-05) |
| | 25 | 4.58E-01(7.77E-05)† | 2.96E-01(2.34E-05)† | 5.10E-01(9.25E-05)† | 6.79E-01(2.59E-04)† | 2.24E-01(1.04E-05) | 2.24E-01(9.45E-06) |
| | 50 | 4.78E-01(4.13E-05)† | 2.68E-01(1.15E-05)† | 5.04E-01(6.86E-05)† | 6.61E-01(1.19E-04)† | 2.08E-01(6.93E-06)‡ | 2.13E-01(3.17E-06) |
| | 200 | 4.92E-01(4.75E-01)† | 2.37E-01(2.33E-01)† | 4.77E-01(4.86E-01)† | 6.89E-01(6.94E-01)† | 2.19E-01(2.18E-01)† | 2.12E-01(2.13E-01) |
| WFG5 | 5 | 4.21E-01(8.09E-05)† | 5.50E-01(2.85E-04)† | 6.05E-01(1.30E-04)† | 1.33E+00(2.40E-04)† | 3.99E-01(4.97E-05)† | 3.31E-01(4.01E-05) |
| | 25 | 4.67E-01(1.18E-04)† | 3.49E-01(5.70E-05)† | 5.29E-01(7.95E-05)† | 1.29E+00(8.98E-05)† | 2.81E-01(1.06E-05)† | 2.74E-01(1.06E-05) |
| | 50 | 4.80E-01(8.30E-05)† | 3.03E-01(2.26E-05)† | 5.10E-01(3.82E-05)† | 1.28E+00(2.37E-04)† | 2.58E-01(8.89E-06)‡ | 2.62E-01(7.01E-06) |
| | 200 | 4.87E-01(5.00E-01)† | 2.61E-01(2.60E-01)† | 4.93E-01(4.89E-01)† | 1.26E+00(1.26E+00)† | 2.49E-01(2.50E-01)‡ | 2.56E-01(2.57E-01) |
| WFG6 | 5 | 4.83E-01(2.05E-03)† | 6.26E-01(1.85E-03)† | 6.68E-01(2.13E-03)† | 1.26E+00(1.17E-04)† | 5.19E-01(1.75E-03)† | 3.66E-01(1.20E-03) |
| | 25 | 5.08E-01(5.82E-04)† | 3.13E-01(3.14E-05) | 5.34E-01(6.67E-04)† | 1.26E+00(3.02E-04)† | 3.42E-01(7.66E-03) | 3.12E-01(1.36E-03) |
| | 50 | 4.99E-01(1.87E-04)† | 2.70E-01(2.38E-05)‡ | 5.09E-01(4.06E-04)† | 1.26E+00(1.76E-04)† | 3.32E-01(6.08E-03)† | 3.05E-01(1.20E-03) |
| | 200 | 4.89E-01(4.94E-01)† | 2.31E-01(2.22E-01)‡ | 5.04E-01(4.81E-01)† | 1.27E+00(1.27E+00)† | 2.56E-01(2.76E-01)† | 3.24E-01(3.78E-01) |
| WFG7 | 5 | 4.77E-01(1.15E-04)† | 6.81E-01(1.68E-03)† | 5.56E-01(2.33E-04)† | 7.91E-01(9.25E-04)† | 3.27E-01(6.74E-05)† | 2.55E-01(1.36E-05) |
| | 25 | 5.16E-01(6.90E-05)† | 3.61E-01(4.13E-05)† | 5.34E-01(3.07E-05)† | 6.53E-01(7.00E-04)† | 2.09E-01(3.98E-06)† | 2.05E-01(1.06E-05) |
| | 50 | 5.13E-01(2.76E-05)† | 3.23E-01(6.00E-05)† | 5.18E-01(4.50E-05)† | 7.06E-01(3.81E-04)† | 2.00E-01(5.23E-06)‡ | 2.03E-01(6.24E-06) |
| | 200 | 5.04E-01(4.99E-01)† | 2.64E-01(2.75E-01)† | 5.03E-01(5.05E-01)† | 4.84E-01(6.37E-01)† | 2.31E-01(2.26E-01)† | 2.31E-01(2.28E-01) |
| WFG8 | 5 | 4.79E-01(1.69E-04)† | 7.07E-01(2.73E-03)† | 5.56E-01(1.91E-04)† | 7.81E-01(1.22E-03)† | 3.27E-01(9.38E-05)† | 2.53E-01(4.19E-05) |
| | 25 | 5.12E-01(5.47E-05)† | 3.59E-01(1.08E-04)† | 5.32E-01(9.50E-05)† | 6.50E-01(5.64E-04)† | 2.09E-01(6.16E-06)† | 2.05E-01(5.90E-06) |
| | 50 | 5.15E-01(4.31E-05)† | 3.25E-01(5.71E-05)† | 5.18E-01(3.82E-05)† | 7.00E-01(3.03E-04)† | 2.00E-01(5.78E-06) | 2.03E-01(5.58E-06) |
| | 200 | 4.97E-01(4.95E-01)† | 2.78E-01(2.74E-01)† | 5.04E-01(4.99E-01)† | 5.36E-01(5.37E-01)‡ | 2.27E-01(2.26E-01)† | 2.35E-01(2.33E-01) |
| WFG9 | 5 | 9.07E-01(1.04E-02) | 8.38E-01(4.19E-04)‡ | 1.11E+00(2.00E-04)† | 9.00E-01(2.84E-03)‡ | 9.74E-01(2.26E-04)† | 9.25E-01(3.02E-03) |
| | 25 | 1.05E+00(7.24E-03)† | 8.46E-01(1.10E-04)‡ | 9.54E-01(1.70E-02)† | 8.49E-01(5.64E-04) | 9.23E-01(3.52E-04)† | 8.70E-01(2.34E-03) |
| | 50 | 1.07E+00(6.78E-03)† | 8.37E-01(1.56E-03)‡ | 8.06E-01(4.32E-04)‡ | 8.56E-01(3.78E-04) | 8.98E-01(2.22E-04)† | 8.67E-01(1.14E-03) |
| | 200 | 1.13E+00(7.76E-01)† | 8.47E-01(8.37E-01)‡ | 7.70E-01(7.11E-01)‡ | 7.51E-01(7.28E-01)‡ | 8.72E-01(9.14E-01)‡ | 9.35E-01(8.70E-01) |
| +-= | | 49/2/1 | 41/10/1 | 50/2/0 | 44/5/3 | 38/9/5 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

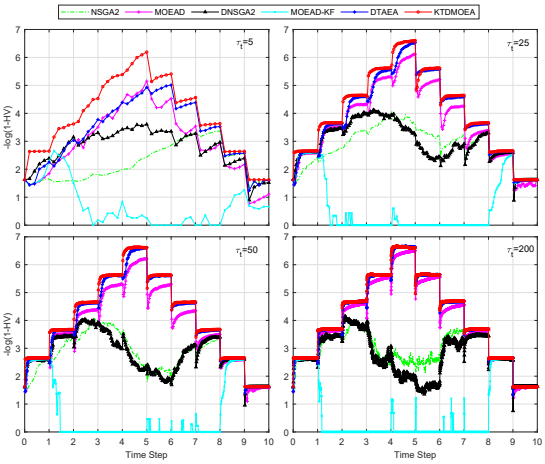
Table 5.10: Mean and standard deviation values of MMS metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 7.62E-01(4.79E-04)† | 8.44E-01(2.29E-04)† | 7.53E-01(6.87E-04)† | 9.50E-01(2.96E-04)† | 8.19E-01(7.65E-04)† | 1.00E+00(7.69E-11) |
| | 25 | 9.53E-01(1.47E-04)† | 9.82E-01(1.68E-05)† | 9.56E-01(1.29E-04)† | 9.61E-01(2.91E-04)† | 9.84E-01(4.23E-05)† | 9.96E-01(1.99E-05) |
| | 50 | 9.93E-01(2.97E-05)† | 9.95E-01(3.93E-06)† | 9.94E-01(2.06E-05)† | 9.85E-01(1.96E-04)† | 9.96E-01(7.37E-06)† | 1.00E+00(9.79E-07) |
| | 200 | 1.00E+00(6.93E-10)† | 1.00E+00(3.86E-11)‡ | 1.00E+00(3.41E-09)† | 9.99E-01(7.00E-06)† | 9.99E-01(1.22E-06)† | 1.00E+00(2.39E-08) |
| F2 | 5 | 9.10E-01(1.08E-04)† | 9.99E-01(1.06E-06)† | 9.99E-01(1.26E-06)† | 9.20E-01(2.32E-03)† | 9.99E-01(9.23E-07)† | 9.99E-01(6.29E-06) |
| | 25 | 9.99E-01(9.38E-07)† | 1.00E+00(0.00E+00)‡ | 1.00E+00(3.55E-11) | 8.35E-01(4.90E-04)† | 1.00E+00(3.61E-12) | 1.00E+00(5.12E-08) |
| | 50 | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(1.37E-12) | 7.49E-01(1.26E-03)† | 1.00E+00(3.43E-12)† | 1.00E+00(7.74E-14) |
| | 200 | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 5.89E-01(7.80E-04)† | 1.00E+00(5.06E-12) | 1.00E+00(4.14E-12) |
| F3 | 5 | 8.65E-01(1.73E-04)† | 9.51E-01(9.82E-05)† | 8.44E-01(3.20E-04)† | 9.04E-01(7.74E-04)† | 9.09E-01(7.89E-04)† | 9.99E-01(8.16E-06) |
| | 25 | 9.84E-01(3.59E-05)† | 1.00E+00(3.07E-08)‡ | 9.83E-01(3.01E-05)† | 8.70E-01(4.89E-04)† | 9.98E-01(3.35E-06) | 9.98E-01(2.87E-06) |
| | 50 | 9.98E-01(1.89E-06)‡ | 1.00E+00(0.00E+00)‡ | 9.98E-01(3.92E-06)‡ | 8.92E-01(1.09E-03)† | 1.00E+00(1.64E-07)‡ | 9.97E-01(3.30E-04) |
| | 200 | 1.00E+00(1.48E-10)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(3.38E-12)‡ | 8.12E-01(1.28E-03)† | 1.00E+00(4.05E-08)‡ | 9.97E-01(3.23E-04) |
| F4 | 5 | 8.21E-01(3.46E-04)‡ | 8.37E-01(1.11E-03)‡ | 9.69E-01(2.58E-04)‡ | 9.45E-01(3.21E-04)‡ | 9.97E-01(1.65E-05)‡ | 7.48E-01(6.87E-03) |
| | 25 | 9.76E-01(9.00E-05)‡ | 9.62E-01(4.15E-04)‡ | 9.92E-01(1.09E-04)‡ | 9.71E-01(4.45E-05)‡ | 9.99E-01(6.21E-06)‡ | 7.17E-01(8.45E-04) |
| | 50 | 9.99E-01(1.09E-05)‡ | 9.90E-01(1.29E-04)‡ | 1.00E+00(2.28E-12)‡ | 9.75E-01(7.13E-05)‡ | 1.00E+00(2.51E-12)‡ | 7.19E-01(1.97E-03) |
| | 200 | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 9.79E-01(2.65E-05)‡ | 1.00E+00(2.12E-12)‡ | 7.71E-01(6.92E-03) |
| WFG1 | 5 | 6.71E-01(3.67E-04)† | 6.27E-01(2.85E-04)† | 6.16E-01(3.62E-04)† | 4.64E-01(3.15E-04)† | 6.92E-01(2.99E-04)† | 9.39E-01(1.82E-04) |
| | 25 | 8.26E-01(1.37E-04)† | 6.05E-01(6.46E-04)† | 8.18E-01(1.91E-04)† | 5.12E-01(2.60E-04)† | 7.88E-01(3.69E-04)† | 9.55E-01(3.01E-05) |
| | 50 | 8.91E-01(7.17E-05)† | 5.70E-01(9.17E-04)† | 8.84E-01(1.09E-04)† | 5.30E-01(1.08E-04)† | 8.67E-01(1.60E-04)† | 9.72E-01(1.90E-05) |
| | 200 | 9.71E-01(8.17E-06)† | 5.42E-01(5.45E-05)† | 9.71E-01(1.26E-05)† | 5.31E-01(3.30E-05)† | 9.60E-01(1.50E-05)† | 9.75E-01(1.72E-05) |
| WFG2 | 5 | 7.60E-01(2.19E-04)† | 6.29E-01(2.33E-04)† | 8.19E-01(2.03E-04)† | 1.82E-01(3.11E-06)† | 8.53E-01(1.89E-04)† | 9.95E-01(3.37E-05) |
| | 25 | 9.43E-01(6.74E-05)† | 5.65E-01(1.36E-04)† | 9.65E-01(2.06E-05)† | 1.87E-01(1.40E-07)† | 9.58E-01(1.84E-05)† | 9.84E-01(7.96E-06) |
| | 50 | 9.80E-01(1.17E-05) | 5.64E-01(3.81E-05)† | 9.83E-01(6.13E-06)‡ | 1.88E-01(1.25E-07)† | 9.82E-01(2.02E-06)‡ | 9.54E-01(1.97E-02) |
| | 200 | 9.96E-01(3.92E-07)‡ | 5.70E-01(1.39E-05)† | 9.95E-01(3.45E-07)‡ | 1.88E-01(6.74E-07)† | 9.76E-01(2.65E-07)‡ | 9.74E-01(3.13E-07) |
| WFG3 | 5 | 9.99E-01(5.43E-07)‡ | 9.71E-01(1.93E-04)† | 9.47E-01(8.61E-05)† | 4.77E-01(8.18E-06)† | 9.91E-01(8.26E-05)† | 9.98E-01(2.30E-05) |
| | 25 | 9.97E-01(3.79E-06)‡ | 9.85E-01(1.49E-04) | 9.87E-01(7.69E-06) | 4.79E-01(1.20E-06)† | 9.90E-01(6.51E-05)† | 9.74E-01(1.39E-03) |
| | 50 | 9.97E-01(8.93E-07)‡ | 9.87E-01(1.21E-04)† | 9.93E-01(2.17E-06)‡ | 4.79E-01(6.80E-07)† | 9.95E-01(1.76E-05)‡ | 9.89E-01(8.96E-04) |
| | 200 | 9.99E-01(5.83E-08)‡ | 9.94E-01(2.66E-05)‡ | 9.98E-01(1.10E-07)‡ | 4.81E-01(2.19E-07)† | 9.34E-01(6.21E-02)† | 9.92E-01(1.29E-03) |
| WFG4 | 5 | 9.22E-01(2.84E-04)† | 9.63E-01(1.55E-04)† | 9.82E-01(9.76E-06)† | 8.46E-01(5.30E-05)† | 9.92E-01(8.82E-06)† | 1.00E+00(6.82E-08) |
| | 25 | 1.00E+00(8.80E-08)† | 9.58E-01(3.78E-05)† | 9.95E-01(2.09E-07)† | 8.55E-01(7.27E-07)† | 1.00E+00(6.68E-08)† | 1.00E+00(3.18E-08) |
| | 50 | 1.00E+00(6.37E-08)† | 9.65E-01(2.76E-05)† | 9.97E-01(1.27E-07)† | 8.57E-01(5.23E-07)† | 1.00E+00(3.75E-10)† | 1.00E+00(8.66E-12) |
| | 200 | 1.00E+00(9.75E-09)† | 9.75E-01(8.59E-06)† | 9.99E-01(1.33E-08)† | 8.57E-01(2.18E-07)† | 1.00E+00(1.23E-09)† | 1.00E+00(7.27E-12) |
| WFG5 | 5 | 8.89E-01(1.24E-04)† | 9.60E-01(7.27E-05)† | 9.72E-01(4.18E-06)† | 8.37E-01(2.89E-05)† | 9.84E-01(3.58E-06)† | 9.86E-01(9.88E-07) |
| | 25 | 9.86E-01(7.21E-07) | 9.56E-01(1.84E-05)† | 9.82E-01(2.43E-06)† | 8.47E-01(3.40E-06)† | 9.87E-01(7.13E-07)‡ | 9.86E-01(6.81E-07) |
| | 50 | 9.87E-01(3.37E-09)‡ | 9.59E-01(1.29E-05)† | 9.84E-01(2.14E-06)† | 8.48E-01(6.46E-06)† | 9.88E-01(4.61E-07)‡ | 9.87E-01(1.44E-07) |
| | 200 | 9.88E-01(6.56E-08)‡ | 9.67E-01(5.71E-06)† | 9.87E-01(1.00E-07)† | 8.52E-01(5.69E-06)† | 9.90E-01(4.70E-08)‡ | 9.87E-01(6.54E-08) |
| WFG6 | 5 | 8.90E-01(1.54E-04)† | 9.16E-01(1.76E-04)† | 9.68E-01(6.05E-05)† | 3.83E-01(4.15E-06)† | 9.69E-01(5.24E-05)† | 9.75E-01(8.72E-05) |
| | 25 | 9.85E-01(1.61E-05)‡ | 9.47E-01(1.16E-05)† | 9.87E-01(1.61E-05)‡ | 3.90E-01(3.37E-06)† | 9.78E-01(1.79E-04) | 9.74E-01(5.22E-04) |
| | 50 | 9.91E-01(4.94E-06)‡ | 9.56E-01(1.60E-05) | 9.91E-01(8.12E-06)‡ | 3.91E-01(1.22E-06)† | 9.79E-01(1.45E-04)‡ | 9.36E-01(2.33E-03) |
| | 200 | 9.96E-01(2.55E-06)‡ | 9.68E-01(6.94E-06)† | 9.95E-01(2.91E-06)‡ | 3.90E-01(2.91E-07)† | 9.87E-01(8.90E-05)‡ | 9.81E-01(3.60E-05) |
| WFG7 | 5 | 9.97E-01(3.34E-06)† | 9.56E-01(6.06E-05)† | 9.89E-01(3.26E-06)† | 9.90E-01(2.02E-05)† | 9.99E-01(8.04E-07)‡ | 9.98E-01(1.23E-06) |
| | 25 | 1.00E+00(1.91E-07)† | 9.74E-01(9.32E-06)† | 9.97E-01(1.50E-07)† | 9.94E-01(2.08E-06)† | 1.00E+00(1.77E-08)‡ | 9.90E-01(9.03E-04) |
| | 50 | 1.00E+00(4.30E-08)† | 9.77E-01(1.07E-05)† | 9.99E-01(4.36E-08)† | 9.96E-01(2.43E-06)† | 1.00E+00(4.77E-10)† | 1.00E+00(2.65E-12) |
| | 200 | 1.00E+00(1.99E-10)† | 9.85E-01(7.19E-06)† | 1.00E+00(8.89E-10)† | 9.92E-01(9.48E-06)† | 1.00E+00(1.80E-09)† | 1.00E+00(3.07E-12) |
| WFG8 | 5 | 9.97E-01(4.52E-06)† | 9.59E-01(6.27E-05)† | 9.89E-01(5.11E-06)† | 9.89E-01(4.37E-05)† | 9.99E-01(1.65E-06) | 9.99E-01(1.16E-06) |
| | 25 | 9.99E-01(1.75E-07)† | 9.73E-01(1.21E-05)† | 9.97E-01(1.92E-07)† | 9.93E-01(3.07E-06)† | 1.00E+00(2.03E-08)‡ | 9.94E-01(6.24E-04) |
| | 50 | 1.00E+00(3.86E-08)† | 9.77E-01(5.63E-06)† | 9.99E-01(5.48E-08)† | 9.95E-01(2.42E-06)† | 1.00E+00(5.92E-08)† | 1.00E+00(1.87E-12) |
| | 200 | 1.00E+00(3.60E-10)† | 9.85E-01(7.09E-06)† | 1.00E+00(1.07E-09)† | 9.92E-01(1.08E-05)† | 1.00E+00(9.93E-10)† | 1.00E+00(3.18E-12) |
| WFG9 | 5 | 7.26E-01(8.41E-03)† | 8.13E-01(1.25E-03)† | 8.80E-01(6.64E-06)‡ | 9.42E-01(1.52E-03)‡ | 8.83E-01(1.34E-05)‡ | 8.34E-01(2.50E-03) |
| | 25 | 8.29E-01(3.05E-03)‡ | 8.53E-01(1.40E-04)‡ | 9.28E-01(7.84E-04)‡ | 9.74E-01(6.27E-04)‡ | 8.87E-01(1.49E-05)‡ | 7.92E-01(3.56E-03) |
| | 50 | 8.51E-01(2.17E-03)‡ | 8.60E-01(1.82E-04)‡ | 9.53E-01(6.16E-06)‡ | 9.72E-01(5.50E-04)‡ | 8.89E-01(1.24E-05)‡ | 7.97E-01(3.31E-03) |
| | 200 | 8.96E-01(1.10E-03)‡ | 8.82E-01(7.34E-04)‡ | 9.57E-01(1.78E-04)‡ | 9.79E-01(4.02E-04)‡ | 8.92E-01(1.87E-05)‡ | 8.58E-01(5.17E-04) |
| +-= | | 29/21/2 | 35/15/2 | 44/8/0 | 48/4/0 | 26/21/5 | — |

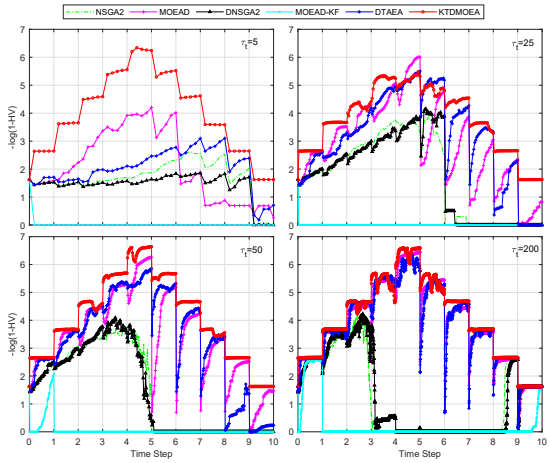
Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.



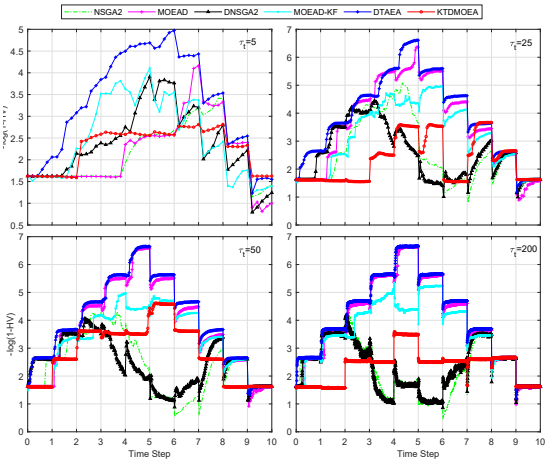

(a) F1



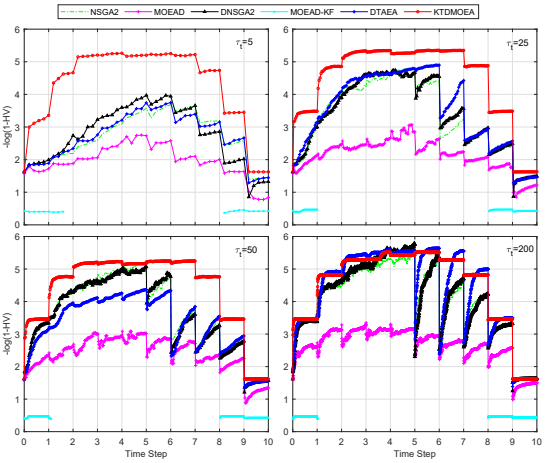

(b) F2



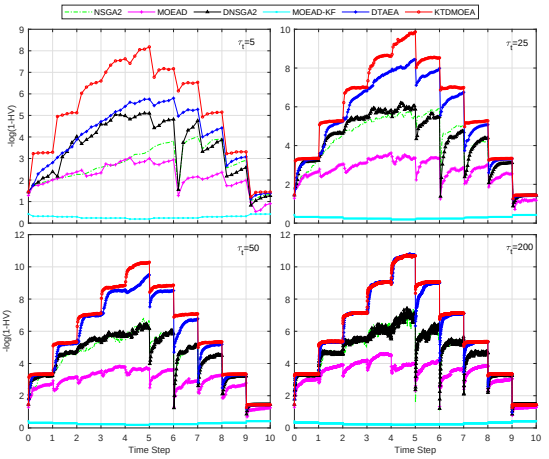

(c) F3




(d) F4

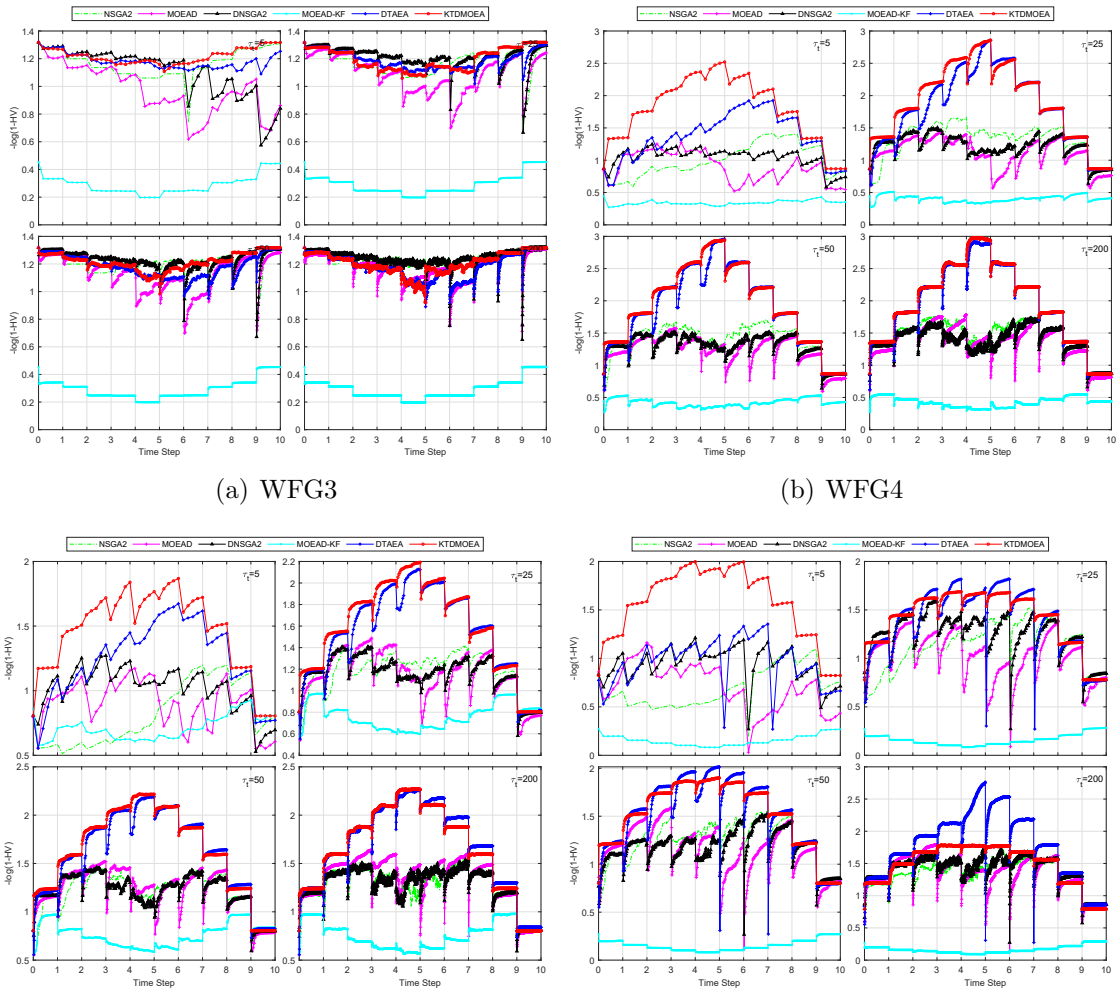



(e) WFG1




(f) WFG2

Figure 5.10: HV trajectories across the whole evolution process for the NObj changing sequence of '2-7-2 one by one'.



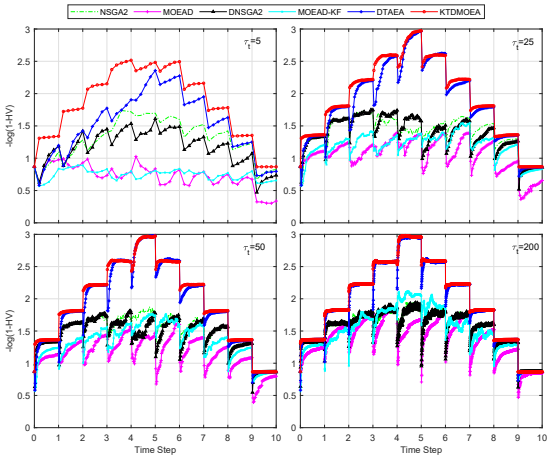

(a) WFG3

(b) WFG4

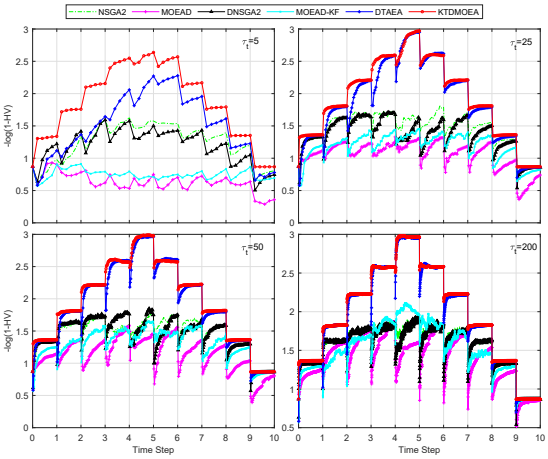
(c) WFG5

(d) WFG6

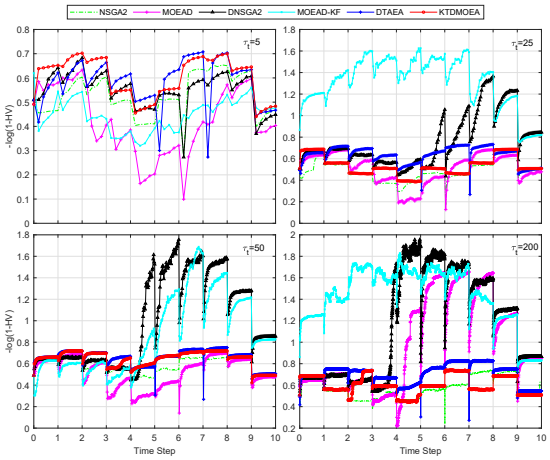
Figure 5.11: HV trajectories across the whole evolution process for the NObj changing sequence of '2-7-2 one by one'.




(a) WFG7

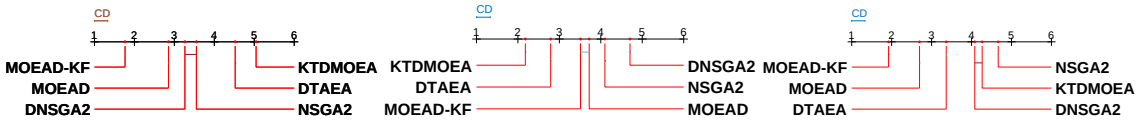



(b) WFG8




(c) WFG9

Figure 5.12: HV trajectories across the whole evolution process for the NObj changing sequence of '2-7-2 one by one'.




(a) MHV (b) MGD (c) MMS

Figure 5.13: Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by 6 algorithms in the changing sequence of firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

Moreover, it is clear from Table 5.8 that KTDMOEA performs significantly worse than MOEAD on F1 and F3 only when the frequency of change is 200. As for the comparison results of the proposed KTDMOEA against the other approaches on WFG1, WFG5, WFG6 and WFG9, it is clear from Table 5.8 that KTDMOEA performs significantly worse than DTAEA when $\tau_t$ is large. In addition, at the last generation of all or half NObjs, KTDMOEA performs significantly worse than DTAEA when $\tau_t$ is large, which can be found from Figures 5.10(e), 5.11(c), 5.11(d) and 5.12(c).

#### 5.4.2.2 NObj Decreasing from 7 to 2 And then Increasing from 2 to 7

Figure 5.13 presents the Nemenyi post-tests results among MHV, MGD and MMS of obtained solutions at the last generation after optimization by 6 algorithms. Friedman test detects significant differences in average values for MHV, MGD and MMS with a p-value of 3.56E-223, 1.58E-129, and 4.98E-183, respectively. Tables 5.11-5.13 show mean and standard deviation values of MHV, MGD and MMS of obtained solutions in the first generation after changes averaged across 10 environmental changes in the sequence of changes as '7-2-7 one by one'.

It can be found from the Friedman test results in Figure 5.13 that KTDMOEA achieves significantly better results than all other algorithms regarding MHV and MGD metrics. As for the MS results, KTDMOEA and DNGSA2 rank the second in the Friedman ranking test, both are outperformed by NSGA2 only. Overall, those statistical results imply that the proposed knowledge transfer is able to help the optimization in obtaining better convergence and at least similar diversity compared to the start-of-the-arts in the changing sequence of firstly decreasing from 7 to 2 and then increasing from 2 to 7, under all frequencies of changes on most problems.

Similar to the other changing sequence of '2-7-2 one by one', KTDMOEA gets worse MHV and MMS values on F4 than MOEAD, MOEAD-KF and DTAEA; KTDMOEA gets significantly worse mean values of MHV and MMS than NSGA2, DNSGA2 and DTAEA on WFG3, which can be observed from Tables 5.11 and 5.13. In addition, it is also clear from Tables 5.11-5.13 KTDMOEA gets worse results than MOEA/D.

Table 5.11: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 0.00E+00(0.00E+00)† | 5.55E-01(3.14E-03)‡ | 0.00E+00(0.00E+00)† | 1.37E-02(4.54E-04)† | 3.66E-01(3.80E-03) | 3.60E-01(1.38E-02) |
| | 25 | 1.28E-01(1.92E-02)† | 9.88E-01(2.40E-05)‡ | 4.98E-02(8.67E-03)† | 0.00E+00(0.00E+00)† | 7.62E-01(4.80E-03)† | 8.14E-01(5.04E-03) |
| | 50 | 4.05E-01(7.48E-03)† | 9.96E-01(2.90E-08)‡ | 3.90E-01(4.57E-03)† | 0.00E+00(0.00E+00)† | 9.82E-01(2.69E-04)† | 9.95E-01(1.06E-05) |
| | 200 | 6.30E-01(4.99E-03)† | 9.96E-01(2.29E-12)‡ | 4.46E-01(2.42E-03)† | 6.54E-02(1.65E-03)† | 9.96E-01(4.44E-09)‡ | 9.96E-01(1.54E-07) |
| F2 | 5 | 8.63E-01(5.29E-04)† | 9.28E-01(2.09E-05)† | 9.13E-01(2.06E-05)† | 2.78E-01(3.77E-03)† | 9.48E-01(1.25E-06)† | 9.54E-01(1.08E-06) |
| | 25 | 9.20E-01(1.02E-04)† | 9.53E-01(6.17E-07)† | 9.28E-01(2.68E-05)† | 2.95E-01(6.94E-04)† | 9.58E-01(3.95E-09)† | 9.58E-01(1.32E-08) |
| | 50 | 9.25E-01(7.13E-05)† | 9.55E-01(7.03E-08)† | 9.23E-01(2.63E-05)† | 2.72E-01(2.38E-04)† | 9.58E-01(1.41E-09)† | 9.58E-01(9.93E-10) |
| | 200 | 9.21E-01(9.92E-05)† | 9.58E-01(1.32E-08)† | 9.16E-01(5.86E-05)† | 2.67E-01(5.88E-05)† | 9.59E-01(3.39E-10)† | 9.59E-01(3.27E-10) |
| F3 | 5 | 0.00E+00(0.00E+00)† | 3.79E-01(2.61E-03)‡ | 0.00E+00(0.00E+00)† | 0.00E+00(0.00E+00)† | 1.72E-01(3.67E-03) | 1.64E-01(1.35E-02) |
| | 25 | 1.28E-02(1.48E-03)† | 9.14E-01(1.41E-04)‡ | 1.85E-02(2.52E-03)† | 0.00E+00(0.00E+00)† | 4.97E-01(8.85E-03) | 4.98E-01(2.03E-02) |
| | 50 | 3.51E-01(3.99E-03)† | 9.51E-01(4.08E-06) | 3.33E-01(5.48E-03)† | 0.00E+00(0.00E+00)† | 7.36E-01(1.87E-02)† | 9.21E-01(4.63E-03) |
| | 200 | 4.32E-01(2.20E-03)† | 9.57E-01(3.23E-08)† | 3.98E-01(1.65E-03)† | 3.71E-02(1.75E-03)† | 9.57E-01(8.00E-07)† | 9.58E-01(6.20E-07) |
| F4 | 5 | 5.96E-01(2.45E-04)† | 9.23E-01(2.53E-05) | 8.71E-01(7.99E-05)† | 9.00E-01(6.83E-05)† | 9.34E-01(1.44E-05)‡ | 9.13E-01(7.25E-04) |
| | 25 | 8.38E-01(4.48E-05)† | 9.50E-01(2.20E-05)‡ | 8.90E-01(5.87E-05)† | 9.46E-01(3.71E-05)‡ | 9.58E-01(3.55E-08)‡ | 8.91E-01(2.32E-03) |
| | 50 | 8.66E-01(2.38E-05)† | 9.55E-01(9.45E-06)‡ | 8.82E-01(1.91E-05)† | 9.51E-01(1.69E-05)‡ | 9.58E-01(3.50E-09)‡ | 9.09E-01(2.16E-04) |
| | 200 | 8.63E-01(2.59E-05)† | 9.58E-01(2.07E-08)‡ | 8.60E-01(1.12E-05)† | 9.54E-01(2.74E-07)‡ | 9.59E-01(2.69E-10)‡ | 9.08E-01(5.07E-04) |
| WFG1 | 5 | 8.58E-01(8.92E-05)† | 7.42E-01(7.79E-05)† | 8.40E-01(9.80E-05)† | 8.07E-01(2.70E-06)† | 8.48E-01(6.14E-05)† | 8.71E-01(3.34E-05) |
| | 25 | 9.29E-01(1.87E-05)† | 8.43E-01(2.45E-05)† | 9.26E-01(1.85E-05)† | 8.10E-01(1.95E-08)† | 9.32E-01(1.83E-05)† | 9.45E-01(9.73E-06) |
| | 50 | 9.52E-01(4.20E-06)† | 8.78E-01(1.40E-05)† | 9.51E-01(4.47E-06)† | 8.10E-01(9.22E-09)† | 9.58E-01(2.70E-06)† | 9.62E-01(1.22E-06) |
| | 200 | 9.68E-01(1.19E-07)† | 9.21E-01(7.92E-06)† | 9.68E-01(5.49E-08)† | 8.10E-01(4.83E-08)† | 9.71E-01(4.92E-08) | 9.71E-01(7.19E-08) |
| WFG2 | 5 | 9.05E-01(1.32E-04)† | 8.32E-01(6.70E-05)† | 9.26E-01(1.68E-05)† | 2.39E-01(9.44E-07)† | 9.34E-01(1.37E-05)† | 9.51E-01(4.80E-06) |
| | 25 | 9.61E-01(6.50E-07)† | 9.23E-01(2.60E-06)† | 9.62E-01(2.73E-07)† | 2.42E-01(7.66E-09)† | 9.66E-01(5.96E-08)† | 9.67E-01(3.26E-08) |
| | 50 | 9.65E-01(6.74E-08)† | 9.35E-01(1.17E-06)† | 9.64E-01(6.21E-08)† | 2.42E-01(1.44E-09)† | 9.68E-01(4.79E-09)† | 9.68E-01(3.60E-09) |
| | 200 | 9.65E-01(2.18E-08)† | 9.48E-01(3.60E-07)† | 9.65E-01(2.86E-08)† | 2.42E-01(9.66E-10)† | 9.68E-01(1.85E-10) | 9.68E-01(2.77E-10) |
| WFG3 | 5 | 6.45E-01(5.06E-05)‡ | 5.58E-01(2.24E-04)† | 6.37E-01(3.03E-05) | 2.46E-01(7.24E-06)† | 6.19E-01(5.36E-05)† | 6.35E-01(4.68E-05) |
| | 25 | 7.00E-01(8.43E-06)‡ | 6.66E-01(1.82E-05)† | 7.07E-01(2.56E-06)‡ | 2.53E-01(2.16E-08)† | 6.79E-01(8.94E-06) | 6.79E-01(2.45E-05) |
| | 50 | 7.09E-01(5.92E-06)‡ | 6.80E-01(4.45E-06)† | 7.13E-01(1.04E-06)‡ | 2.53E-01(8.06E-09)† | 6.89E-01(9.13E-06) | 6.88E-01(1.97E-05) |
| | 200 | 7.16E-01(9.98E-07)‡ | 6.94E-01(1.42E-06)† | 7.16E-01(5.32E-07)‡ | 2.54E-01(1.82E-09)† | 7.00E-01(2.58E-06)‡ | 6.97E-01(4.97E-06) |
| WFG4 | 5 | 6.57E-01(1.12E-04)† | 5.83E-01(2.82E-04)† | 6.62E-01(5.36E-05)† | 3.09E-01(6.43E-05)† | 7.48E-01(4.35E-05)† | 8.08E-01(9.49E-06) |
| | 25 | 7.46E-01(2.78E-05)† | 6.91E-01(5.76E-05)† | 7.25E-01(2.48E-05)† | 3.49E-01(1.89E-05)† | 8.27E-01(3.08E-06)† | 8.30E-01(6.27E-08) |
| | 50 | 7.51E-01(2.01E-05)† | 7.18E-01(1.74E-05)† | 7.32E-01(2.04E-05)† | 3.50E-01(2.05E-05)† | 8.32E-01(6.62E-08)† | 8.32E-01(4.07E-08) |
| | 200 | 7.65E-01(1.67E-05)† | 7.55E-01(9.14E-06)† | 7.54E-01(3.71E-05)† | 3.48E-01(1.66E-06)† | 8.32E-01(1.08E-08)† | 8.32E-01(8.23E-09) |
| WFG5 | 5 | 6.67E-01(3.04E-04)† | 6.07E-01(7.81E-05)† | 6.47E-01(4.89E-05)† | 5.29E-01(5.17E-06)† | 7.77E-01(1.28E-05)† | 7.79E-01(1.04E-05) |
| | 25 | 7.17E-01(3.26E-05)† | 6.93E-01(8.21E-06)† | 7.01E-01(2.17E-05)† | 5.36E-01(2.30E-06)† | 8.02E-01(2.72E-07)† | 8.03E-01(1.60E-06) |
| | 50 | 7.21E-01(1.81E-05)† | 7.11E-01(4.69E-06)† | 7.11E-01(2.58E-05)† | 5.33E-01(1.75E-06)† | 8.04E-01(4.02E-08)† | 8.04E-01(6.46E-08) |
| | 200 | 7.28E-01(2.02E-05)† | 7.39E-01(2.18E-06)† | 7.23E-01(1.69E-05)† | 5.32E-01(1.52E-06)† | 8.04E-01(1.88E-08)† | 8.04E-01(2.64E-08) |
| WFG6 | 5 | 5.79E-01(1.90E-04)† | 4.96E-01(2.17E-04)† | 6.21E-01(1.26E-04)† | 1.46E-01(2.67E-06)† | 6.80E-01(2.14E-04)† | 7.66E-01(1.83E-04) |
| | 25 | 7.14E-01(5.63E-05)† | 6.77E-01(1.25E-05)† | 7.31E-01(1.99E-05)† | 1.53E-01(3.33E-07)† | 8.16E-01(6.02E-06)† | 8.24E-01(2.18E-06) |
| | 50 | 7.43E-01(1.30E-05)† | 7.10E-01(1.30E-05)† | 7.45E-01(1.68E-05)† | 1.54E-01(2.22E-08)† | 8.29E-01(3.79E-07)† | 8.30E-01(2.22E-07) |
| | 200 | 7.61E-01(1.53E-05)† | 7.56E-01(5.90E-06)† | 7.60E-01(1.31E-05)† | 1.55E-01(1.56E-08)† | 8.32E-01(2.49E-08)† | 8.32E-01(5.14E-08) |
| WFG7 | 5 | 7.14E-01(5.28E-05)† | 4.72E-01(2.58E-04)† | 6.81E-01(4.98E-05)† | 5.79E-01(7.21E-05)† | 7.67E-01(1.84E-05)† | 8.01E-01(5.14E-06) |
| | 25 | 7.65E-01(1.25E-05)† | 6.58E-01(6.81E-05)† | 7.58E-01(1.46E-05)† | 7.29E-01(1.26E-05)† | 8.29E-01(1.56E-07)† | 8.30E-01(7.57E-08) |
| | 50 | 7.74E-01(4.37E-06)† | 7.11E-01(2.03E-05)† | 7.71E-01(5.70E-06)† | 7.49E-01(2.89E-05)† | 8.32E-01(3.90E-08)† | 8.32E-01(2.89E-08) |
| | 200 | 7.84E-01(2.02E-06)† | 7.68E-01(4.32E-06)† | 7.83E-01(1.05E-06)† | 7.72E-01(9.54E-06)† | 8.33E-01(1.85E-09)† | 8.33E-01(2.46E-09) |
| WFG8 | 5 | 7.11E-01(4.50E-05)† | 4.73E-01(1.96E-04)† | 6.81E-01(4.83E-05)† | 5.75E-01(1.69E-04)† | 7.70E-01(1.27E-05)† | 8.01E-01(6.09E-06) |
| | 25 | 7.65E-01(1.25E-05)† | 6.58E-01(6.81E-05)† | 7.58E-01(1.46E-05)† | 7.28E-01(2.73E-05)† | 8.29E-01(9.57E-08)† | 8.30E-01(1.80E-07) |
| | 50 | 7.75E-01(3.40E-06)† | 7.12E-01(1.63E-05)† | 7.71E-01(5.73E-06)† | 7.49E-01(2.40E-05)† | 8.32E-01(2.68E-08)† | 8.32E-01(2.65E-08) |
| | 200 | 7.84E-01(1.87E-06)† | 7.68E-01(4.94E-06)† | 7.84E-01(9.69E-07)† | 7.72E-01(7.30E-06)† | 8.33E-01(3.56E-09)† | 8.33E-01(2.17E-09) |
| WFG9 | 5 | 7.13E-01(8.80E-05)‡ | 4.00E-01(1.63E-03)† | 6.52E-01(4.33E-04)‡ | 6.48E-01(1.77E-04)‡ | 6.02E-01(1.33E-02) | 6.06E-01(1.03E-02) |
| | 25 | 7.65E-01(1.25E-05) | 6.58E-01(6.81E-05) | 7.58E-01(1.46E-05) | 7.50E-01(1.03E-05) | 7.11E-01(1.55E-02) | 6.93E-01(1.60E-02) |
| | 50 | 7.64E-01(1.02E-05) | 7.09E-01(7.41E-03) | 7.66E-01(8.58E-06) | 7.57E-01(1.26E-05) | 6.56E-01(1.37E-02) | 6.76E-01(1.62E-02) |
| | 200 | 7.78E-01(7.26E-06)‡ | 7.57E-01(6.08E-03)‡ | 7.81E-01(2.62E-06)‡ | 7.55E-01(9.93E-06)‡ | 6.65E-01(1.49E-02)‡ | 6.42E-01(1.41E-02) |
| +-= | | 44/6/2 | 38/10/4 | 44/5/3 | 45/5/2 | 35/7/10 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.12: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 1.91E+02(8.26E+01)† | 1.17E+01(4.57E+00)‡ | 2.03E+02(8.19E+01)† | 5.79E+02(3.58E+03)† | 3.21E+01(1.06E+01)‡ | 3.73E+01(7.85E+01) |
| | 25 | 9.29E+01(5.06E+01)† | 1.89E+00(3.46E-01)‡ | 1.17E+02(4.25E+02)† | 7.27E+02(3.15E+03)† | 5.98E+00(2.32E+00)† | 3.27E+00(1.57E+00) |
| | 50 | 8.08E+01(8.68E+01)† | 1.78E+00(3.64E-01)† | 9.10E+01(5.14E+01)† | 7.93E+02(9.04E+02)† | 7.28E-01(1.48E-01)† | 2.11E-01(2.76E-02) |
| | 200 | 1.12E+02(1.20E+02)† | 1.78E+00(1.93E+00)† | 1.09E+02(1.28E+02)† | 8.69E+02(8.73E+02)† | 4.69E-02(3.44E-02)† | 1.31E-02(1.36E-02) |
| F2 | 5 | 3.81E-01(3.26E-03)† | 1.72E-01(5.96E-05)† | 3.22E-01(3.10E-04)† | 3.96E+00(2.44E-01)† | 1.03E-01(2.65E-05)† | 7.54E-02(1.12E-05) |
| | 25 | 2.59E-01(4.72E-04)† | 8.75E-02(9.81E-06)† | 2.95E-01(2.47E-04)† | 6.10E+00(7.08E-02)† | 4.44E-02(1.96E-06)† | 4.18E-02(1.38E-06) |
| | 50 | 2.86E-01(2.63E-04)† | 7.14E-02(6.72E-06)† | 3.22E-01(1.84E-04)† | 6.53E+00(9.92E-02)† | 3.58E-02(4.83E-07)† | 3.53E-02(3.95E-07) |
| | 200 | 3.18E-01(3.32E-01)† | 4.63E-02(4.51E-02)† | 3.63E-01(3.63E-01)† | 6.70E+00(6.47E+00)† | 3.25E-02(3.24E-02) | 3.24E-02(3.15E-02) |
| F3 | 5 | 4.07E+02(3.96E+02)† | 2.71E+01(1.74E+01)‡ | 4.32E+02(6.44E+02)† | 7.91E+02(2.79E+03)† | 8.14E+01(8.46E+01)‡ | 9.13E+01(5.38E+02) |
| | 25 | 1.97E+02(1.94E+02)† | 3.92E+00(3.28E+00)‡ | 2.27E+02(6.93E+02)† | 9.33E+02(2.90E+03)† | 1.71E+01(1.82E+01)† | 1.25E+01(2.41E+01) |
| | 50 | 1.60E+02(1.19E+02)† | 4.02E+00(2.05E+00)† | 1.64E+02(1.58E+02)† | 1.03E+03(2.65E+03)† | 3.55E+00(3.33E+00)† | 8.47E-01(3.97E-01) |
| | 200 | 2.52E+02(2.38E+02)† | 2.94E+00(2.57E+00)† | 2.38E+02(2.61E+02)† | 1.11E+03(1.06E+03)† | 1.47E-01(5.48E-02)† | 9.65E-02(3.34E-01) |
| F4 | 5 | 9.24E-01(1.55E-03)† | 1.02E-01(1.23E-04) | 4.98E-01(4.35E-04)† | 1.08E-01(1.48E-03) | 1.67E-01(1.28E-04)† | 9.93E-02(2.42E-04) |
| | 25 | 4.12E-01(2.25E-04)† | 5.93E-02(3.22E-05)† | 4.12E-01(2.56E-04)† | 1.54E-02(1.13E-05)‡ | 4.06E-02(1.86E-06)† | 3.20E-02(1.42E-05) |
| | 50 | 4.22E-01(2.35E-04)† | 5.69E-02(9.84E-06)† | 4.34E-01(1.16E-04)† | 1.28E-02(3.06E-06)‡ | 3.24E-02(3.77E-07)† | 2.77E-02(1.96E-06) |
| | 200 | 4.50E-01(4.46E-01)† | 4.20E-02(4.53E-02)† | 4.66E-01(4.80E-01)† | 1.08E-02(1.04E-02)‡ | 3.31E-02(3.36E-02)† | 2.78E-02(2.64E-02) |
| WFG1 | 5 | 4.82E-01(2.93E-04)† | 6.41E-01(1.47E-03)† | 5.14E-01(3.43E-04)† | 1.25E+00(1.97E-03)† | 4.30E-01(1.49E-04)† | 3.81E-01(9.24E-05) |
| | 25 | 3.23E-01(1.08E-04)† | 3.87E-01(1.82E-04)† | 3.30E-01(1.33E-04)† | 1.24E+00(4.16E-04)† | 2.55E-01(9.73E-05)† | 2.05E-01(2.22E-03) |
| | 50 | 2.89E-01(3.96E-05)† | 3.13E-01(8.66E-05)† | 2.94E-01(5.53E-05)† | 1.26E+00(1.58E-04)† | 2.05E-01(2.58E-05)† | 1.44E-01(4.22E-03) |
| | 200 | 2.80E-01(2.78E-01)† | 2.33E-01(2.28E-01)† | 2.85E-01(2.88E-01)† | 1.28E+00(1.26E+00)† | 1.70E-01(1.59E-01)‡ | 1.73E-01(1.67E-01) |
| WFG2 | 5 | 7.20E-01(1.18E-03)† | 4.69E-01(4.17E-04)‡ | 6.81E-01(3.64E-04)† | 3.12E-01(1.44E-05)‡ | 5.64E-01(5.22E-04)† | 5.11E-01(2.30E-04) |
| | 25 | 6.63E-01(4.82E-04)† | 3.11E-01(2.33E-05)‡ | 6.67E-01(5.33E-04)† | 3.08E-01(2.20E-07)‡ | 4.47E-01(9.41E-05)‡ | 4.61E-01(6.93E-05) |
| | 50 | 7.05E-01(3.83E-04)† | 3.03E-01(3.14E-05)‡ | 7.09E-01(3.98E-04)† | 3.08E-01(3.29E-07)‡ | 4.50E-01(3.53E-05) | 4.52E-01(4.09E-05) |
| | 200 | 7.16E-01(7.39E-01)† | 3.01E-01(2.95E-01)‡ | 7.19E-01(7.06E-01)† | 3.07E-01(3.08E-01)‡ | 4.32E-01(4.35E-01) | 4.27E-01(4.33E-01) |
| WFG3 | 5 | 1.42E+00(9.10E-03) | 2.20E+00(1.17E-03)† | 1.81E+00(3.31E-03)† | 1.30E+00(3.79E-04)‡ | 1.57E+00(8.32E-04)† | 1.39E+00(9.49E-03) |
| | 25 | 1.34E+00(3.45E-02)‡ | 2.26E+00(3.55E-04)† | 1.92E+00(6.87E-04)† | 1.24E+00(1.51E-05)‡ | 1.61E+00(6.16E-04)† | 1.55E+00(7.80E-03) |
| | 50 | 1.65E+00(1.31E-02) | 2.28E+00(4.13E-04)† | 1.96E+00(5.61E-04)† | 1.23E+00(7.80E-06)‡ | 1.68E+00(4.86E-04)† | 1.61E+00(3.70E-03) |
| | 200 | 1.95E+00(1.98E+00)† | 2.29E+00(2.27E+00)† | 2.00E+00(2.03E+00)† | 1.23E+00(1.23E+00)‡ | 1.72E+00(1.70E+00)† | 1.68E+00(1.70E+00) |
| WFG4 | 5 | 4.86E-01(1.22E-04)† | 4.89E-01(3.12E-04)† | 5.80E-01(1.45E-04)† | 4.38E-01(2.49E-04)† | 3.43E-01(6.43E-05)† | 2.92E-01(3.71E-05) |
| | 25 | 4.61E-01(6.27E-05)† | 2.99E-01(2.79E-05)† | 5.03E-01(7.70E-05)† | 3.98E-01(1.33E-04)† | 2.24E-01(1.28E-05)† | 2.19E-01(6.61E-06) |
| | 50 | 4.71E-01(4.04E-05)† | 2.70E-01(1.56E-05)† | 5.01E-01(4.56E-05)† | 4.17E-01(1.63E-04)† | 2.09E-01(5.78E-06) | 2.09E-01(6.39E-06) |
| | 200 | 4.71E-01(4.70E-01)† | 2.34E-01(2.36E-01)† | 4.74E-01(4.70E-01)† | 4.11E-01(4.10E-01)† | 2.19E-01(2.14E-01)† | 2.15E-01(2.17E-01) |
| WFG5 | 5 | 5.13E-01(2.29E-04)† | 5.08E-01(2.58E-04)† | 6.38E-01(1.86E-04)† | 7.22E-02(1.13E-05)‡ | 3.42E-01(8.28E-05)† | 3.38E-01(7.80E-05) |
| | 25 | 4.78E-01(8.03E-05)† | 3.45E-01(4.54E-05)† | 5.28E-01(9.40E-05)† | 5.87E-02(5.34E-06)‡ | 2.46E-01(6.64E-06) | 2.47E-01(8.31E-06) |
| | 50 | 4.82E-01(8.83E-05)† | 3.02E-01(2.17E-05)† | 5.10E-01(7.13E-05)† | 6.13E-02(6.00E-06)‡ | 2.38E-01(4.64E-06) | 2.40E-01(7.22E-06) |
| | 200 | 4.88E-01(4.77E-01)† | 2.59E-01(2.57E-01)† | 4.85E-01(5.04E-01)† | 5.96E-02(6.01E-02)‡ | 2.37E-01(2.39E-01) | 2.41E-01(2.42E-01) |
| WFG6 | 5 | 5.69E-01(4.39E-04)† | 6.32E-01(5.34E-04)† | 7.20E-01(8.31E-04)† | 1.14E-01(1.71E-05)‡ | 5.55E-01(1.42E-03)† | 3.96E-01(9.29E-04) |
| | 25 | 4.76E-01(8.12E-05)† | 3.16E-01(5.33E-05)† | 5.06E-01(7.75E-05)† | 1.14E-01(1.12E-06)‡ | 2.48E-01(2.10E-05)† | 2.33E-01(1.11E-05) |
| | 50 | 4.72E-01(4.44E-05)† | 2.74E-01(2.20E-05)† | 4.85E-01(4.68E-05)† | 1.14E-01(7.95E-07)‡ | 2.19E-01(4.43E-06)† | 2.18E-01(7.16E-06) |
| | 200 | 4.68E-01(4.72E-01)† | 2.32E-01(2.26E-01)† | 4.77E-01(4.69E-01)† | 1.21E-01(1.20E-01)‡ | 2.11E-01(2.15E-01)‡ | 2.16E-01(2.12E-01) |
| WFG7 | 5 | 5.73E-01(8.87E-05)† | 6.32E-01(2.69E-03)† | 6.54E-01(1.90E-04)† | 2.51E-01(1.21E-04)‡ | 3.65E-01(7.46E-05)† | 3.01E-01(4.34E-05) |
| | 25 | 5.19E-01(4.64E-05)† | 3.82E-01(1.84E-04)† | 5.34E-01(5.94E-05)† | 8.91E-02(1.59E-05)‡ | 2.15E-01(4.20E-06)† | 2.12E-01(6.68E-06) |
| | 50 | 5.12E-01(3.76E-05)† | 3.31E-01(7.90E-05)† | 5.21E-01(4.54E-05)† | 6.54E-02(3.41E-06)‡ | 2.04E-01(5.71E-06) | 2.05E-01(4.46E-06) |
| | 200 | 4.98E-01(5.13E-01)† | 2.71E-01(2.64E-01)† | 5.11E-01(5.06E-01)† | 5.51E-02(5.38E-02)‡ | 2.29E-01(2.25E-01)‡ | 2.30E-01(2.30E-01) |
| WFG8 | 5 | 5.76E-01(1.36E-04)† | 6.45E-01(1.30E-03)† | 6.55E-01(1.87E-04)† | 2.51E-01(1.82E-04)‡ | 3.63E-01(7.22E-05)† | 3.02E-01(2.04E-05) |
| | 25 | 5.19E-01(4.64E-05)† | 3.82E-01(1.84E-04)† | 5.34E-01(5.94E-05)† | 8.93E-02(1.19E-05)‡ | 2.14E-01(4.43E-06)† | 2.12E-01(4.51E-06) |
| | 50 | 5.12E-01(3.22E-05)† | 3.32E-01(6.67E-05)† | 5.20E-01(3.42E-05)† | 6.51E-02(3.01E-06)‡ | 2.03E-01(6.16E-06) | 2.04E-01(7.01E-06) |
| | 200 | 5.11E-01(5.03E-01)† | 2.69E-01(2.72E-01)† | 5.07E-01(4.96E-01)† | 5.43E-02(5.35E-02)‡ | 2.29E-01(2.23E-01) | 2.31E-01(2.29E-01) |
| WFG9 | 5 | 6.07E-01(5.82E-04)‡ | 8.16E-01(8.13E-03)† | 7.84E-01(3.16E-03) | 1.50E-01(5.77E-05)‡ | 7.67E-01(5.12E-02)† | 7.11E-01(3.47E-02) |
| | 25 | 5.19E-01(4.64E-05) | 3.82E-01(1.84E-04) | 5.34E-01(5.94E-05) | 8.61E-02(2.53E-06)‡ | 4.50E-01(5.72E-02) | 4.79E-01(5.66E-02) |
| | 50 | 5.32E-01(3.18E-05) | 3.39E-01(2.08E-02) | 5.37E-01(4.43E-05) | 7.65E-02(2.53E-06)‡ | 5.42E-01(4.90E-02) | 5.03E-01(5.85E-02) |
| | 200 | 5.21E-01(5.27E-01)‡ | 2.43E-01(2.40E-01)‡ | 5.20E-01(5.05E-01)‡ | 6.55E-02(6.22E-02)‡ | 6.20E-01(6.88E-01) | 7.49E-01(4.45E-01) |
| +-= | | 45/3/4 | 40/9/3 | 48/1/3 | 20/31/1 | 33/6/13 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.13: Mean and standard deviation values of MMS metric for optimized solutions obtained by all compared algorithms at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one.

| Prob. | $\tau_t$ | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|---|
| F1 | 5 | 1.00E+00(0.00E+00)‡ | 9.61E-01(2.41E-04) | 9.99E-01(2.74E-06)‡ | 9.36E-01(4.39E-04) | 9.99E-01(3.55E-06)‡ | 9.36E-01(9.97E-03) |
| | 25 | 1.00E+00(0.00E+00)‡ | 9.78E-01(2.18E-05)‡ | 1.00E+00(6.75E-06)‡ | 9.72E-01(2.24E-04)‡ | 6.39E-01(2.32E-01)† | 9.02E-01(3.88E-04) |
| | 50 | 1.00E+00(3.66E-07)‡ | 9.94E-01(4.11E-06)‡ | 1.00E+00(2.50E-07)‡ | 9.70E-01(1.03E-04)† | 9.97E-01(5.72E-06)‡ | 9.89E-01(8.92E-04) |
| | 200 | 1.00E+00(1.54E-09)‡ | 1.00E+00(1.28E-09)‡ | 1.00E+00(1.54E-09)‡ | 9.63E-01(1.50E-04)† | 9.99E-01(6.19E-07)‡ | 9.96E-01(3.22E-04) |
| F2 | 5 | 9.65E-01(1.95E-04)† | 1.00E+00(3.52E-08)‡ | 1.00E+00(1.01E-06) | 9.36E-01(1.40E-03)† | 9.99E-01(1.11E-06) | 9.98E-01(6.01E-06) |
| | 25 | 9.99E-01(1.35E-06)† | 1.00E+00(0.00E+00)‡ | 1.00E+00(2.36E-11)‡ | 8.55E-01(9.51E-04)† | 6.45E-01(2.37E-01)† | 1.00E+00(1.07E-07) |
| | 50 | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(5.53E-12) | 7.45E-01(1.27E-03)† | 1.00E+00(5.65E-12)† | 1.00E+00(2.43E-12) |
| | 200 | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(0.00E+00)‡ | 5.34E-01(1.81E-04)† | 1.00E+00(3.47E-12) | 1.00E+00(4.03E-12) |
| F3 | 5 | 1.00E+00(0.00E+00)‡ | 9.94E-01(8.92E-06) | 1.00E+00(0.00E+00)‡ | 8.99E-01(8.45E-04) | 1.00E+00(9.81E-09) | 9.06E-01(1.40E-02) |
| | 25 | 1.00E+00(0.00E+00)‡ | 1.00E+00(3.03E-07)‡ | 1.00E+00(0.00E+00)‡ | 9.13E-01(3.39E-04)‡ | 6.44E-01(2.36E-01) | 8.77E-01(1.63E-03) |
| | 50 | 1.00E+00(8.19E-08)‡ | 1.00E+00(1.25E-11)‡ | 1.00E+00(0.00E+00)‡ | 9.06E-01(2.89E-04)† | 1.00E+00(5.95E-07) | 9.96E-01(3.23E-04) |
| | 200 | 1.00E+00(2.78E-14)‡ | 1.00E+00(0.00E+00)‡ | 1.00E+00(2.77E-11)‡ | 7.65E-01(1.68E-03)† | 1.00E+00(1.31E-07) | 9.97E-01(3.22E-04) |
| F4 | 5 | 9.95E-01(5.29E-05)‡ | 9.58E-01(2.31E-04)‡ | 9.87E-01(1.04E-04)‡ | 9.89E-01(2.48E-04)‡ | 9.95E-01(2.27E-05)‡ | 8.91E-01(1.28E-03) |
| | 25 | 9.76E-01(8.00E-05)‡ | 9.90E-01(1.67E-04)‡ | 9.92E-01(1.11E-04)‡ | 9.84E-01(1.49E-04)‡ | 6.45E-01(2.36E-01)‡ | 8.50E-01(2.31E-03) |
| | 50 | 1.00E+00(7.94E-10)‡ | 9.98E-01(2.91E-05)‡ | 1.00E+00(1.08E-11)‡ | 9.87E-01(3.37E-05)‡ | 1.00E+00(1.77E-06)‡ | 8.38E-01(1.83E-03) |
| | 200 | 1.00E+00(0.00E+00)‡ | 1.00E+00(3.23E-16)‡ | 1.00E+00(0.00E+00)‡ | 9.82E-01(1.40E-05)‡ | 1.00E+00(1.96E-12)‡ | 8.42E-01(2.67E-03) |
| WFG1 | 5 | 8.31E-01(2.95E-04)† | 6.29E-01(8.15E-04)† | 7.79E-01(2.31E-04)† | 4.47E-01(2.50E-04)† | 7.87E-01(5.45E-04)† | 8.51E-01(4.75E-04) |
| | 25 | 8.34E-01(1.35E-04)† | 5.52E-01(3.69E-04)† | 8.20E-01(1.32E-04)† | 4.83E-01(1.10E-04)† | 5.13E-01(1.50E-01)† | 8.83E-01(1.97E-04) |
| | 50 | 8.85E-01(9.55E-05)† | 5.22E-01(1.77E-04)† | 8.84E-01(9.32E-05)† | 4.89E-01(7.80E-05)† | 8.47E-01(1.46E-04)† | 9.15E-01(6.05E-05) |
| | 200 | 9.71E-01(1.22E-05)‡ | 5.20E-01(1.25E-04)† | 9.70E-01(1.14E-05)‡ | 4.91E-01(2.23E-05)† | 9.61E-01(1.53E-05)† | 9.67E-01(1.44E-05) |
| WFG2 | 5 | 8.02E-01(1.90E-04)† | 5.61E-01(8.99E-04)† | 8.59E-01(2.00E-04)† | 1.83E-01(4.75E-06)† | 8.76E-01(1.47E-04)† | 9.45E-01(2.85E-04) |
| | 25 | 9.38E-01(6.02E-05)† | 5.33E-01(2.69E-04)† | 9.63E-01(1.39E-05)† | 1.88E-01(1.35E-07)† | 6.14E-01(2.14E-01)† | 9.72E-01(3.17E-04) |
| | 50 | 9.79E-01(5.35E-06)† | 5.39E-01(2.54E-04)† | 9.82E-01(3.25E-06) | 1.88E-01(1.34E-07)† | 9.80E-01(6.87E-06) | 9.81E-01(4.57E-06) |
| | 200 | 9.95E-01(2.15E-07)‡ | 5.47E-01(2.17E-04)† | 9.95E-01(2.81E-07)‡ | 1.87E-01(5.19E-07)† | 9.76E-01(3.40E-07)‡ | 9.76E-01(1.69E-07) |
| WFG3 | 5 | 8.89E-01(2.95E-04)‡ | 7.50E-01(4.04E-03)† | 8.35E-01(2.70E-04)† | 4.76E-01(2.38E-05)† | 8.20E-01(4.40E-04)† | 8.57E-01(6.63E-04) |
| | 25 | 9.84E-01(1.36E-05)‡ | 9.02E-01(8.45E-04)† | 9.75E-01(2.44E-05)† | 4.78E-01(8.08E-07)† | 6.01E-01(2.05E-01)† | 9.23E-01(9.19E-04) |
| | 50 | 9.94E-01(2.29E-06)‡ | 9.31E-01(4.76E-04)† | 9.90E-01(3.19E-06)‡ | 4.79E-01(1.49E-06)† | 9.54E-01(2.40E-04) | 9.47E-01(6.09E-04) |
| | 200 | 9.99E-01(6.98E-08)‡ | 9.73E-01(5.07E-05)† | 9.98E-01(1.00E-07)‡ | 4.80E-01(3.72E-07)† | 9.87E-01(3.90E-05) | 9.80E-01(2.17E-04) |
| WFG4 | 5 | 9.38E-01(9.66E-05)† | 8.95E-01(4.74E-04)† | 9.77E-01(8.49E-06)† | 8.43E-01(2.15E-05)† | 9.83E-01(2.04E-05)† | 9.95E-01(4.46E-06) |
| | 25 | 9.97E-01(2.18E-07)† | 9.37E-01(7.76E-05)† | 9.95E-01(2.31E-07)† | 8.54E-01(2.58E-06)† | 6.44E-01(2.36E-01)† | 9.99E-01(3.19E-07) |
| | 50 | 9.99E-01(8.26E-08)† | 9.48E-01(4.17E-05)† | 9.97E-01(1.88E-07)† | 8.57E-01(1.13E-06)† | 1.00E+00(2.77E-08)† | 1.00E+00(3.40E-08) |
| | 200 | 1.00E+00(8.16E-09)† | 9.61E-01(2.07E-05)† | 9.99E-01(2.03E-08)† | 8.57E-01(1.24E-07)† | 1.00E+00(1.14E-08)† | 1.00E+00(4.48E-09) |
| WFG5 | 5 | 9.56E-01(1.20E-04)† | 9.21E-01(1.07E-04)† | 9.67E-01(5.03E-06)† | 8.42E-01(8.65E-06)† | 9.85E-01(2.86E-06)† | 9.85E-01(3.81E-06) |
| | 25 | 9.86E-01(3.73E-07)† | 9.39E-01(2.14E-05)† | 9.82E-01(2.94E-06)† | 8.46E-01(4.96E-06)† | 6.39E-01(2.32E-01)† | 9.91E-01(1.05E-06) |
| | 50 | 9.87E-01(1.80E-07)† | 9.45E-01(1.70E-05)† | 9.85E-01(1.90E-06)† | 8.47E-01(3.15E-06)† | 9.91E-01(3.13E-07) | 9.91E-01(1.53E-07) |
| | 200 | 9.88E-01(3.42E-07)† | 9.55E-01(1.83E-05)† | 9.87E-01(1.06E-07)† | 8.51E-01(4.86E-06)† | 9.92E-01(1.43E-08) | 9.92E-01(2.61E-08) |
| WFG6 | 5 | 9.15E-01(7.35E-05)† | 8.31E-01(6.82E-04)† | 9.65E-01(2.05E-05)† | 3.93E-01(5.72E-06)† | 9.68E-01(2.92E-05)† | 9.83E-01(1.86E-05) |
| | 25 | 9.92E-01(5.42E-06)† | 9.26E-01(3.93E-05)† | 9.93E-01(1.10E-06)† | 3.91E-01(1.12E-06)† | 6.43E-01(2.35E-01)† | 9.97E-01(1.53E-06) |
| | 50 | 9.97E-01(3.94E-07)† | 9.40E-01(3.42E-05)† | 9.96E-01(4.57E-07)† | 3.91E-01(1.06E-07)† | 9.99E-01(1.54E-07)† | 9.99E-01(3.97E-07) |
| | 200 | 9.99E-01(1.49E-07)† | 9.56E-01(1.55E-05)† | 9.99E-01(6.64E-08)† | 3.90E-01(3.46E-08)† | 1.00E+00(1.32E-08) | 1.00E+00(1.11E-08) |
| WFG7 | 5 | 9.86E-01(3.33E-06)† | 8.64E-01(1.07E-03)† | 9.78E-01(9.31E-06)† | 9.53E-01(2.61E-05)† | 9.86E-01(5.10E-06)† | 9.92E-01(3.71E-06) |
| | 25 | 9.98E-01(1.58E-07)† | 9.45E-01(4.43E-05)† | 9.97E-01(2.28E-07)† | 9.91E-01(5.30E-06)† | 6.44E-01(2.36E-01)† | 9.99E-01(2.34E-07) |
| | 50 | 9.99E-01(2.29E-08)† | 9.58E-01(2.73E-05)† | 9.99E-01(4.51E-08)† | 9.96E-01(1.96E-06)† | 1.00E+00(1.19E-07) | 1.00E+00(2.39E-07) |
| | 200 | 1.00E+00(4.46E-10)† | 9.74E-01(1.84E-05)† | 1.00E+00(5.95E-10)† | 9.94E-01(6.24E-06)† | 1.00E+00(6.16E-10) | 1.00E+00(1.92E-09) |
| WFG8 | 5 | 9.86E-01(4.49E-06)† | 8.69E-01(3.12E-04)† | 9.78E-01(8.55E-06)† | 9.49E-01(2.04E-05)† | 9.87E-01(5.33E-06)† | 9.92E-01(2.49E-06) |
| | 25 | 9.98E-01(1.58E-07)† | 9.45E-01(4.43E-05)† | 9.97E-01(2.28E-07)† | 9.92E-01(2.69E-06)† | 6.44E-01(2.36E-01)† | 9.99E-01(5.22E-07) |
| | 50 | 9.99E-01(2.55E-08)† | 9.61E-01(1.74E-05)† | 9.99E-01(4.85E-08)† | 9.95E-01(2.74E-06)† | 1.00E+00(6.57E-08)† | 1.00E+00(1.00E-07) |
| | 200 | 1.00E+00(3.17E-10)† | 9.75E-01(1.42E-05)† | 1.00E+00(6.85E-10)† | 9.94E-01(8.15E-06)† | 1.00E+00(1.57E-09)† | 1.00E+00(1.12E-09) |
| WFG9 | 5 | 9.89E-01(5.30E-06)‡ | 8.44E-01(5.45E-04)† | 9.73E-01(6.49E-05)‡ | 9.67E-01(2.45E-05)‡ | 9.35E-01(1.65E-03)† | 9.29E-01(1.19E-03) |
| | 25 | 9.98E-01(1.58E-07) | 9.45E-01(4.43E-05) | 9.97E-01(2.28E-07) | 9.87E-01(4.33E-06) | 6.20E-01(2.20E-01)† | 9.55E-01(1.67E-03) |
| | 50 | 9.98E-01(1.55E-07) | 9.50E-01(9.11E-04) | 9.98E-01(6.71E-08) | 9.88E-01(4.94E-06) | 9.43E-01(1.44E-03) | 9.50E-01(1.69E-03) |
| | 200 | 9.99E-01(6.61E-08)‡ | 9.64E-01(6.39E-04)‡ | 9.99E-01(3.60E-08)‡ | 9.91E-01(6.01E-06)‡ | 9.46E-01(1.57E-03) | 9.38E-01(1.49E-03) |
| +-= | | 28/22/2 | 33/15/4 | 27/20/5 | 40/8/4 | 28/8/16 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

It should be noted that the performance of KTDMOEA against other algorithms on WFG1, WFG5 and WFG6 in the changing sequence of '7-2-7 one by one' is different from that of the other changing sequence '2-7-2 one by one'. Specifically, in this NObj changing sequence '7-2-7 one by one', KTDMOEA significantly performs best at almost all cases, which can be found from Tables 5.11-5.13.

#### 5.4.2.3 Why Does Knowledge Transfer Help Optimization?

It has been presented in Section 5.4.1 that the proposed PS expansion/contraction has indeed enhanced the diversity of knowledge transfer, resulting in better solution quality of obtained solutions than other state-of-the-arts in the first generation after changes. In other words, given the results of better solution quality than other algorithms in the first generation after changes, our proposed KTDMOEA is able to find solutions with good convergence and diversity at at all frequencies of change, even when the frequency of change is very high. This means that our proposed approach is robust to different frequencies of change.

Because the transferred solutions are better distributed in the new environment with a better diversity already, KTDMOEA is able to find better solutions across different frequencies of change. This is also the reason why KTDMOEA is able to quickly respond to the changes in the NObj, since finding good solution under high frequency of change means fast response to changes. There are some problems where KTDMOEA did not perform best when the frequency of changes is large. The specific results and analyses are presented in next paragraph.

#### 5.4.2.4 Why Doesn't Knowledge Transfer sometimes Help Optimization?

It has been presented in this section that KTDMOEA gets worse optimized solutions at the last generation after optimization on some problems. This section aims to analyze the reasons why it performs worse. The reason why the proposed KTDMOEA performs worse on F4 and WFG3 is that the diversity maintenance strategy in KTDMOEA is unable to maintain diversity when the transferred solutions themselves had poor diversity right after the change, as analyzed in Section 5.4.1. The reason behind this is that diversity maintenance strategy in KTDMOEA can only increase the even distribution of solutions via the evenly generated weight vectors, unable to spread the solutions to cover other areas of the PF. It can be concluded that the optimization algorithm in KTDMOEA cannot increase the spread of population if the knowledge transfer cannot provide enough diversity.

The reason why KTDMOEA performs worse than MOEA/D on F1 and F3 when $\tau_t$ is large can be analyzed as follows. It has been found that when $\tau_t$ is large and the

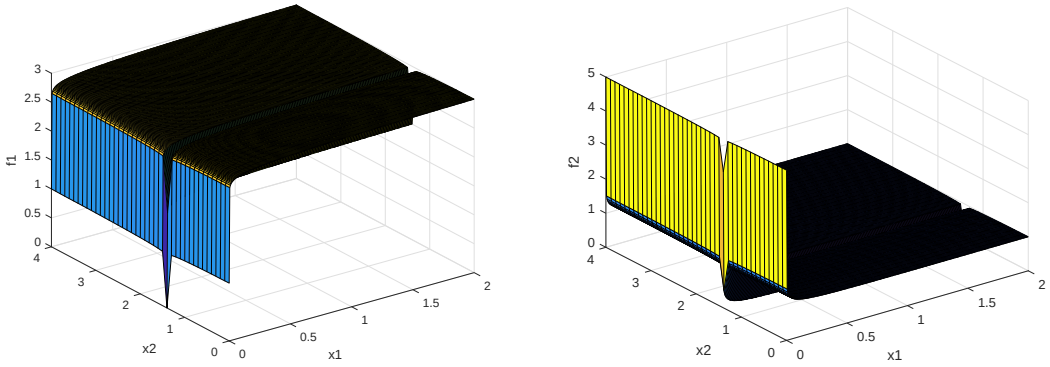



Figure 5.14: Fitness landscape of bi-objective WFG1 problem with two variables ($x1$ and $x2$) where the sample step sizes for x1 and x2 are 0.005 and 0.1, respectively.

NObj is larger than 3 KTDMOEA probably produces dominance resistance solutions (DRS) at several runs, while MOEAD has enough generation to make the population perform well through draw the final plot of found solutions at some objectives. The reason why KTDMOEA probably generates DRSs is that the environmental selection in KTDMOEA is based on nondominated sorting and the diversity maintenance in KTDMOEA cannot delete DRSs, while the diversity maintenance in MOEAD is able to. The reason why KTDMOEA performs worse on F1 and F3 with multimodality rather than other problem with multimodality like WFG4 is that decision variables of F1 and F3 are correlated, which results in lots of DRSs in the fitness landscape. Therefore, KTDMOEA cannot get better results on problems with multimodality and variable correlation when the frequency of change is low (i.e. $\tau_t$ is large).

It should be noted that KTDMOEA performs worse on WFG1, WFG5, WFG6 and WFG9 when $\tau_t$ is large only in the changing sequence '2-7-2 one by one'. The reason is that when $\tau_t$ is large, the proposed KTDMOEA is unable to provide enough diversity such that the population can reach the global area of those problems, as they have a vary narrow valley area (WFG1, WFG5 and WFG9) and several nonseparable variables (WFG6) in the fitness landscape. However, DTAEA with an extra specialized archive for diversity maintenance can achieve this aim when $\tau_t$ is large. The diagram of fitness landscape for WFG1, WFG5 and WFG9 with two variables ($x_1$ and $x_2$) are drawn in Figures 5.14-5.16, respectively. It is clear that each objective of those problems have a vary narrow valley area, which may be difficult for the proposed KTDMOEA to each when $\tau_t$ is large. The reason why KTDMOEA does not perform worse on WFG1, WFG5, WFG6 and WFG9 in the changing sequence '7-2-7 one by one' is that population diversity may be increased inherently when decreasing the NObj, as analyzed in [3]. The introduced diversity together with the diversity introduced via PS expansion/contraction makes KTDMOEA able to perform better than other algorithms than DTAEA on those four problems.

'

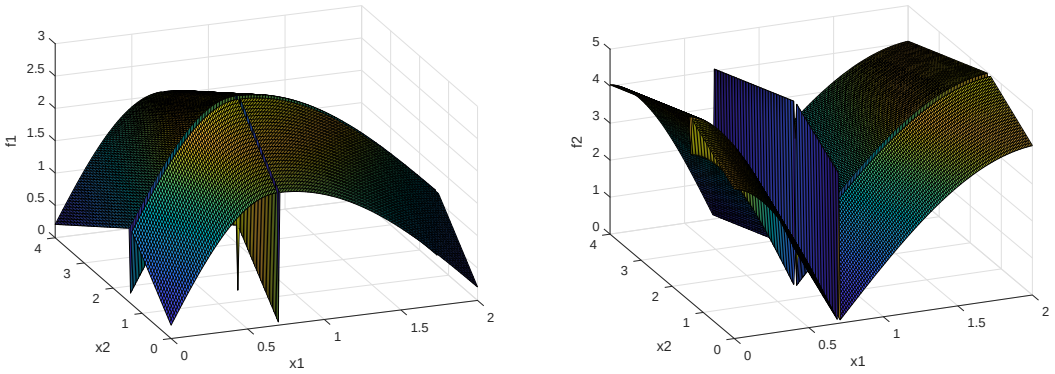



Figure 5.15: Fitness landscape of bi-objective WFG5 problem with two variables ($x1$ and $x2$) where the sample step sizes for x1 and x2 are 0.005 and 0.01, respectively.

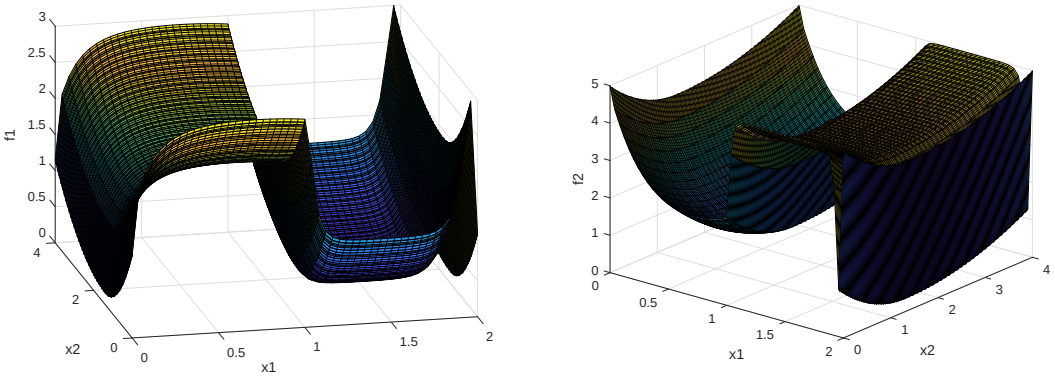



Figure 5.16: Fitness landscape of bi-objective WFG9 problem with two variables ($x1$ and $x2$) where the sample step sizes for x1 and x2 are 0.002 and 0.004, respectively.

### 5.4.3 Further Analysis of Our Knowledge Transfer Methods

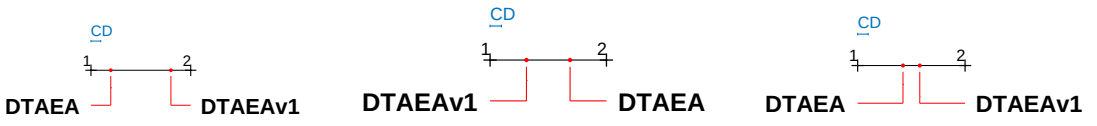
In order to further verify the effectiveness of the proposed PS expansion/contraction method against the state-of-the-art method DTAEA, two pairs of comparisons are designed. First, DTAEA is compared to DTAEAv1, where the CA reconstruction after changes is replaced by the proposed PS expansion/contraction with other components of DTAEA unchanged. Second, KTDMOEA is compared to KTDMOEAv1, where the proposed PS expansion/contraction is replaced by the CA reconstruction of DTAEA with other components of KTDMOEA unchanged.

All experimental settings are set the same as in Section 5.3.3 except for the frequency of change and the NObj changing sequence, which is set as 25 and NObj increasing from 2 to 7 and then decreasing from 7 to 2 one by one, respectively, to save space. For Friedman and Nemenyi tests, the mean metric value of 10 environmental changes that each algorithm gets on one problem with one frequency of change at each independent run is regarded as an observation of the test. Therefore, there are 403 (13 problems and 31 environmental changes) observations for each algorithm in the Friedman and Nemenyi tests.

Figure 5.17 presents the Nemenyi post-tests results among MHV, MGD and MMS of obtained solutions at the last generation after optimization before the next change by DTAEA and DTAEAv1 algorithms. Friedman test detects significant differences in average values for MHV, MGD and MMS with a p-value of 9.97E-34, 5.40E-14, 0.0017, respectively. Similarly, Figure 5.18 presents the Nemenyi post-tests results among MHV, MGD and MMS of obtained solutions at the last generation after optimization before the next change by KTDMOEA and KTDMOEAv1 algorithms. Friedman test detects significant differences in average values for MHV, MGD and MMS with a p-value of 1.50E-57, 4.88E-07, 8.63E-35, respectively.

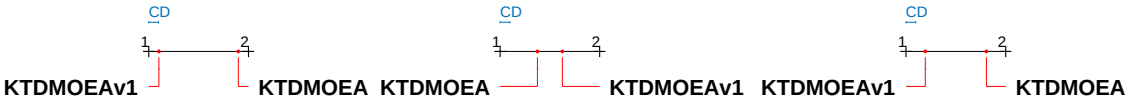
Overall, it can be observed from Figures 5.17 and 5.18 the algorithm with our proposed knowledge transfer strategy significantly outperforms the one without it, i.e, DTAEAv1 outperforms DTAEA and KTDMOEA outperforms KTDMOEAv1. More details can be found from Tables 138-140 of the Suppmentary File. From this result, we can get the conclusion that, the proposed PS expansion/contraction method works better than the knowledge transfer in DTAEA, further confirming the effectiveness of our proposed knowledge transfer method.

The mean and standard deviation values of MHV, MGD and MMS for optimized solutions at the last generation after optimization in the changing sequence of firstly increasing from 2 to 7 and then decreasing from 7 to 2 are presented in Tables 5.14-5.16, respectively. It is clear that DTAEAv1 gets significantly better quality of optimized solution than DTAEA on almost all problems except for WFG2 and WFG9. In addition, it can be also seen from those tables that KTDMOEA significantly outperforms KTDMOEAv1 on all test problems regarding MHV values.




(a) MHV (b) MGD (c) MMS

Figure 5.17: Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by DTAEA and DTAEAv1 in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.




(a) MHV (b) MGD (c) MMS

Figure 5.18: Friedman ranking among MHV, MGD and MMS of optimized solutions at the last generation by KTDMOEA and KTDMOEAv1 in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

Table 5.14: Mean and standard deviation values of MHV metric for solutions obtained by two pairs of comparison under different optimization algorithms at the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | DTAEAv1 | DTAEA | KTDMOEAv1 | KTDMOEA |
|---|---|---|---|---|
| F1 | 9.96E-01(1.46E-07) | 9.75E-01(9.87E-04)† | 9.56E-01(2.47E-03)† | 9.96E-01(9.75E-10) |
| F2 | 9.58E-01(1.61E-09) | 9.58E-01(6.69E-08)† | 9.56E-01(1.17E-06)† | 9.58E-01(1.15E-09) |
| F3 | 9.58E-01(1.05E-06) | 8.86E-01(3.76E-03)† | 8.67E-01(3.97E-03)† | 9.56E-01(1.58E-06) |
| F4 | 9.58E-01(3.66E-09) | 9.58E-01(3.12E-08)† | 7.96E-01(1.67E-04)† | 8.89E-01(4.17E-04) |
| WFG1 | 9.69E-01(4.35E-06) | 9.54E-01(1.21E-05)† | 9.49E-01(1.78E-05)† | 9.70E-01(1.18E-06) |
| WFG2 | 9.56E-01(6.14E-04) | 9.67E-01(2.07E-08)‡ | 9.66E-01(1.37E-06)† | 9.68E-01(2.36E-09) |
| WFG3 | 7.01E-01(7.44E-06) | 6.94E-01(1.06E-05)† | 6.68E-01(5.12E-05)† | 6.89E-01(7.10E-04) |
| WFG4 | 8.30E-01(1.22E-07) | 8.28E-01(5.53E-07)† | 8.25E-01(9.44E-06)† | 8.30E-01(2.10E-07) |
| WFG5 | 7.86E-01(6.81E-07) | 7.85E-01(9.78E-07)† | 7.55E-01(5.55E-05)† | 7.86E-01(1.48E-06) |
| WFG6 | 7.47E-01(1.68E-03) | 7.56E-01(1.77E-03) | 7.39E-01(4.54E-04)† | 7.61E-01(4.48E-04) |
| WFG7 | 8.32E-01(1.40E-08) | 8.31E-01(1.01E-07)† | 8.31E-01(1.35E-07)† | 8.32E-01(3.03E-04) |
| WFG8 | 8.32E-01(2.61E-08) | 8.31E-01(6.00E-08)† | 8.31E-01(1.20E-07)† | 8.32E-01(2.09E-04) |
| WFG9 | 4.86E-01(6.97E-05) | 4.92E-01(6.23E-05)‡ | 4.47E-01(1.50E-03)† | 4.52E-01(1.31E-03) |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.15: Mean and standard deviation values of MGD metric for solutions obtained by two pairs of comparison under different optimization algorithms in the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | DTAEAv1 | DTAEA | KTDMOEAv1 | KTDMOEA |
|---|---|---|---|---|
| F1 | 1.72E-01(1.29E-01) | 1.04E+00(6.01E-01)† | 8.01E-01(3.75E-01)† | 4.34E-02(2.29E-03) |
| F2 | 3.50E-02(8.58E-07) | 3.89E-02(6.02E-07)† | 3.84E-02(4.63E-07)† | 3.38E-02(2.22E-07) |
| F3 | 2.16E-01(9.26E-02) | 4.19E+00(8.22E+00)† | 2.80E+00(3.61E+00)† | 3.00E-01(1.37E-01) |
| F4 | 3.41E-02(6.87E-07) | 3.37E-02(5.80E-07) | 1.47E-02(8.53E-07)‡ | 2.31E-02(3.55E-06) |
| WFG1 | 1.81E-01(2.48E-05) | 2.24E-01(2.83E-05)† | 2.20E-01(5.01E-05)† | 1.81E-01(2.00E-05) |
| WFG2 | 4.36E-01(5.57E-03) | 4.46E-01(5.24E-05)† | 4.43E-01(8.43E-05)‡ | 4.54E-01(3.96E-05) |
| WFG3 | 1.58E+00(9.48E-04) | 1.58E+00(9.13E-04) | 5.30E-01(6.90E-02)‡ | 1.48E+00(5.12E-03) |
| WFG4 | 2.25E-01(9.04E-06) | 2.24E-01(1.04E-05)‡ | 2.21E-01(1.02E-05)‡ | 2.24E-01(9.45E-06) |
| WFG5 | 2.76E-01(8.84E-06) | 2.81E-01(1.06E-05)† | 2.86E-01(2.17E-05)† | 2.74E-01(1.06E-05) |
| WFG6 | 3.44E-01(6.23E-03) | 3.42E-01(7.66E-03) | 3.32E-01(1.65E-03) | 3.12E-01(1.36E-03) |
| WFG7 | 2.05E-01(5.35E-06) | 2.09E-01(3.98E-06)† | 2.07E-01(5.14E-06)† | 2.05E-01(1.06E-05) |
| WFG8 | 2.06E-01(5.54E-06) | 2.09E-01(6.16E-06)† | 2.08E-01(7.66E-06)† | 2.05E-01(5.90E-06) |
| WFG9 | 9.25E-01(4.19E-04) | 9.23E-01(3.52E-04) | 8.94E-01(2.89E-03) | 8.70E-01(2.34E-03) |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Table 5.16: Mean and standard deviation values of MMS metric for solutions obtained by two pairs of comparison under different optimization algorithms in the last generation after optimization on all DMOPs with a changing NObj, when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | DTAEAv1 | DTAEA | KTDMOEAv1 | KTDMOEA |
|---|---|---|---|---|
| F1 | 9.99E-01(3.74E-06) | 9.84E-01(4.23E-05)† | 9.56E-01(8.72E-05)† | 9.96E-01(1.99E-05) |
| F2 | 1.00E+00(2.37E-07) | 1.00E+00(3.61E-12)‡ | 9.98E-01(2.31E-06)† | 1.00E+00(5.12E-08) |
| F3 | 9.99E-01(3.21E-07) | 9.98E-01(3.35E-06)† | 9.86E-01(1.56E-05)† | 9.98E-01(2.87E-06) |
| F4 | 1.00E+00(2.02E-06) | 9.99E-01(6.21E-06)† | 7.05E-01(2.24E-04)† | 7.17E-01(8.45E-04) |
| WFG1 | 9.60E-01(1.09E-04) | 7.88E-01(3.69E-04)† | 7.86E-01(1.93E-04)† | 9.55E-01(3.01E-05) |
| WFG2 | 9.81E-01(6.19E-04) | 9.58E-01(1.84E-05)† | 9.31E-01(8.89E-05)† | 9.84E-01(7.96E-06) |
| WFG3 | 9.97E-01(5.29E-05) | 9.90E-01(6.51E-05)† | 9.95E-01(6.71E-06)‡ | 9.74E-01(1.39E-03) |
| WFG4 | 1.00E+00(6.69E-08) | 1.00E+00(6.68E-08) | 9.98E-01(8.19E-06)† | 1.00E+00(3.18E-08) |
| WFG5 | 9.86E-01(1.06E-06) | 9.87E-01(7.13E-07) | 9.74E-01(2.18E-05)† | 9.86E-01(6.81E-07) |
| WFG6 | 9.75E-01(1.50E-04) | 9.78E-01(1.79E-04) | 9.74E-01(6.21E-05)‡ | 9.74E-01(5.22E-04) |
| WFG7 | 1.00E+00(2.74E-07) | 1.00E+00(1.77E-08) | 1.00E+00(2.63E-08)‡ | 9.90E-01(9.03E-04) |
| WFG8 | 9.99E-01(3.95E-07) | 1.00E+00(2.03E-08) | 1.00E+00(2.64E-07)‡ | 9.94E-01(6.24E-04) |
| WFG9 | 8.86E-01(1.21E-05) | 8.87E-01(1.49E-05) | 8.27E-01(3.28E-03)‡ | 7.92E-01(3.56E-03) |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

All these results demonstrate the effectiveness of our proposed knowledge transfer approach.

### 5.4.4 Performance Comparison on Other Changes in the NObj

In the previous experiments, the NObj only increases or decreases one by one. This section aims to verify the performance of the proposed algorithm in the scenario where the NObj increases or decreases by more than one. Two different changing sequences where the NObj increases or decreases by one or two each time are designed as follows:

$$m(t) = \begin{cases} 2, & t{=}0 \\ 3, & t{=}1 \\ m(t-1)+2, & t \in [2{,}3] \\ m(t-1)-2, & t \in [4{,}5] \\ m(t-1)-1, & t{=}6, \end{cases} \tag{5.9}$$

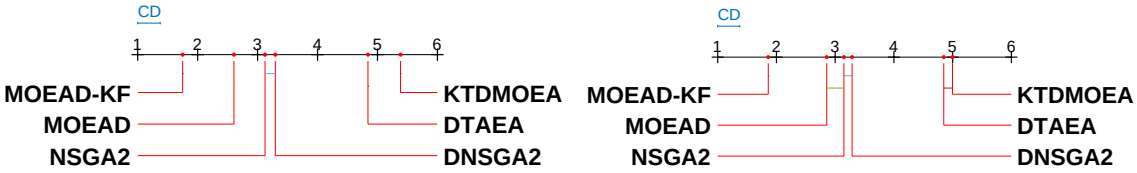
where $t$ is the identifier of the change. In this sequence of change, NObj firstly increases from 2 to 3. Then there are four changes with the first two changes increasing the NObj by two and then two changes decreasing the NObj by two. Lastly, the NObj decreases from 3 to 2.

$$m(t) = \begin{cases} 7, & t{=}0 \\ m(t-1)-2, & t \in [1{,}2] \\ m(t-1)-1, & t{=}3 \\ m(t-1)+1, & t{=}4 \\ m(t-1)+2, & t \in [5{,}6], \end{cases} \tag{5.10}$$

where $t$ is the identifier of the change. In this sequence of change, the initial NObj is set as 7. Then, there are two changes where the NObj decreases by two. Later on, the NObj decreases from 3 to 2 and then increases from 2 to 3. In the last two changes, the NObj increases by two at each change.

All experimental settings are set the same as in Section 5.3.3 except for the frequency of change and the metric, which is set as 25 and MHV, respectively, to save space. For Friedman and Nemenyi tests, the MHV values that all algorithms get on one problem with one frequency of change at one independent run of 31 runs is regarded as an observation of the test. Therefore, there are 403 (13 problems and 1 frequency of changes and 31 independent runs) observed data.

Figure 5.19 (a) and (b) presents the Nemenyi post-tests results among MHV values of optimized solutions at the last generation by all compared algorithms in two changing sequences '2-3-5-7-5-3-2' and '7-5-3-2-3-5-7', respectively. Friedman




(a) For sequence '2-3-5-7-5-3-2'. (b) For sequence '7-5-3-2-3-5-7'.

Figure 5.19: Friedman ranking among MHV of optimized solutions at the last generation by all algorithms in the changing sequences '2-3-5-7-5-3-2' and '7-5-3-2-3-5-7', respectively.

detects significant differences in average value for MHV with a p-value of 9.6123e-232 and 9.6955e-181, respectively for these two changing sequences. Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the last generation after optimization on all DMOPs with a changing NObj, where changes of the NObj following the pattern in equations (5.9) and (5.10) are presented in Tables 5.17 and 5.18, respectively.

Overall, it can be observed from Figure 5.19 in the changing sequence of '2-3-5-7-5-3-2', our proposed KTDMOEA performs the best among all compared algorithms. In another changing sequence of '7-5-3-2-3-5-7', both KTDMOEA and DTAEA performs the best. More details can be found in Tables 5.17 and 5.18.

Table 5.17: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the last generation after optimization on all DMOPs with a changing NObj, where changes of the NObj following the pattern in equation 5.9.

| Prob. | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|
| F1 | 7.65E-01(7.35E-03)† | 9.73E-01(3.06E-04)† | 6.11E-01(8.45E-03)† | 1.52E-01(1.49E-03)† | 9.43E-01(3.84E-03)† | 9.93E-01(5.47E-07) |
| F2 | 9.23E-01(1.15E-05)† | 9.34E-01(2.59E-06)† | 9.26E-01(2.61E-06)† | 4.93E-01(2.98E-03)† | 9.40E-01(1.14E-08)† | 9.40E-01(1.79E-06) |
| F3 | 6.49E-01(2.55E-03)† | 8.50E-01(6.62E-04)† | 6.14E-01(2.72E-03)† | 2.77E-02(5.05E-04)† | 8.25E-01(1.06E-02)† | 9.41E-01(4.28E-06) |
| F4 | 9.14E-01(4.64E-05)‡ | 9.08E-01(2.66E-04)‡ | 9.06E-01(1.28E-04)‡ | 9.14E-01(2.33E-05)‡ | 9.40E-01(3.18E-08)‡ | 8.61E-01(5.04E-04) |
| WFG1 | 9.34E-01(1.86E-05)† | 8.67E-01(4.61E-05)† | 9.32E-01(1.60E-05)† | 1.83E-01(5.84E-08)† | 9.36E-01(1.18E-05)† | 9.52E-01(4.33E-06) |
| WFG2 | 9.42E-01(7.58E-07)† | 9.03E-01(1.25E-05)† | 9.43E-01(2.22E-07)† | 2.48E-01(7.26E-09)† | 9.47E-01(1.57E-07)† | 9.48E-01(2.62E-07) |
| WFG3 | 7.02E-01(1.02E-05)† | 6.90E-01(1.75E-05)† | 7.14E-01(1.73E-06)† | 2.60E-01(2.64E-08)† | 6.96E-01(2.43E-05)† | 7.49E-01(3.86E-05) |
| WFG4 | 7.23E-01(3.14E-05)† | 6.79E-01(3.07E-05)† | 7.02E-01(2.81E-05)† | 3.51E-01(2.07E-05)† | 7.87E-01(6.30E-06)† | 8.06E-01(2.86E-05) |
| WFG5 | 6.75E-01(1.20E-04)† | 6.79E-01(1.40E-05)† | 6.78E-01(1.99E-05)† | 5.34E-01(2.02E-05)† | 7.50E-01(1.77E-06)† | 7.57E-01(7.44E-05) |
| WFG6 | 6.52E-01(1.55E-04)† | 6.62E-01(4.61E-05)† | 6.85E-01(2.06E-04)† | 1.57E-01(1.53E-05)† | 7.21E-01(6.23E-04)† | 7.39E-01(5.06E-04) |
| WFG7 | 7.41E-01(1.40E-05)† | 6.41E-01(4.04E-05)† | 7.38E-01(1.30E-05)† | 7.00E-01(2.79E-05)† | 7.97E-01(1.75E-07)† | 8.18E-01(2.12E-08) |
| WFG8 | 7.41E-01(1.37E-05)† | 6.45E-01(4.61E-05)† | 7.38E-01(1.80E-05)† | 6.99E-01(3.96E-05)† | 7.97E-01(4.64E-07)† | 8.18E-01(1.96E-08) |
| WFG9 | 4.08E-01(1.04E-03)† | 4.07E-01(7.42E-05)† | 4.64E-01(2.07E-03)† | 6.70E-01(7.90E-03)‡ | 4.75E-01(1.09E-04)‡ | 4.65E-01(3.18E-04) |
| +-= | 11/2/0 | 11/2/0 | 12/1/0 | 11/2/0 | 12/1/0 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

The mean and standard deviation values of MHV for optimized solutions at the last generation after optimization in the those two changing sequences are exhibited in Tables 5.17 and 5.18. It is clear from Tables 5.17 that in the changing sequence of

following the pattern in Equation (5.9) that KTDMOEA significantly outperforms than others on all test problems except for F4 and WFG9. The reason is similar to that analyzed previously. For F4 with an extremely strong bias problem, the proposed PS expansion is unable to expand the PS and the diversity maintenance strategy in KTDMOEA cannot provide much diversity to spread the population. As for WFG9, KTDMOEA gets significant worse results than MOEAD-KF and DTAEA. The reason why KTDMOEA performs worse than MOEAD-KF is that the DE operator in MOEAD-KF is able to reach the global optima in the optimization process while KTDMOEA cannot achieve this via SBX operator.

Table 5.18: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms in the last generation right after optimization on all DMOPs with a changing NObj, where changes of the NObj follow the pattern in equation 5.10.

| Prob. | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|
| F1 | 3.58E-02(3.40E-03)† | 8.05E-01(4.69E-03)‡ | 7.47E-03(7.20E-04)† | 1.52E-01(1.49E-03)† | 5.38E-01(5.29E-03)† | 5.97E-01(3.56E-03) |
| F2 | 9.05E-01(1.31E-04)† | 9.35E-01(1.90E-06)† | 9.22E-01(7.18E-06)† | 4.93E-01(2.98E-03)† | 9.39E-01(9.43E-09) | 9.39E-01(1.22E-06) |
| F3 | 4.92E-03(6.90E-04)† | 6.60E-01(2.45E-03)‡ | 2.24E-04(1.54E-06)† | 2.77E-02(5.05E-04)† | 2.38E-01(4.43E-03)† | 3.54E-01(1.26E-02) |
| F4 | 8.20E-01(8.67E-05)† | 9.29E-01(7.68E-05)‡ | 8.96E-01(1.59E-04)‡ | 9.14E-01(2.33E-05)‡ | 9.39E-01(6.13E-08)‡ | 8.68E-01(8.29E-04) |
| WFG1 | 8.98E-01(2.76E-05)† | 8.11E-01(3.82E-05)† | 8.97E-01(2.97E-05)† | 1.83E-01(5.84E-08)† | 9.01E-01(1.53E-05)† | 9.16E-01(1.27E-05) |
| WFG2 | 9.39E-01(1.78E-06)† | 8.98E-01(1.00E-05)† | 9.41E-01(6.65E-07)† | 2.48E-01(7.26E-09)† | 9.44E-01(4.40E-06)† | 9.46E-01(2.40E-07) |
| WFG3 | 6.98E-01(7.44E-06)‡ | 6.74E-01(1.31E-05)† | 7.08E-01(4.29E-06)‡ | 2.60E-01(2.64E-08)† | 6.76E-01(2.30E-05)† | 6.81E-01(2.52E-05) |
| WFG4 | 7.16E-01(5.10E-05)† | 6.73E-01(2.27E-05)† | 7.06E-01(1.92E-05)† | 3.51E-01(2.07E-05)† | 7.85E-01(6.21E-06)† | 7.87E-01(3.12E-05) |
| WFG5 | 6.93E-01(3.43E-05)† | 6.74E-01(3.44E-05)† | 6.76E-01(2.79E-05)† | 5.34E-01(2.02E-05)† | 7.69E-01(2.82E-06)† | 7.70E-01(1.98E-06) |
| WFG6 | 6.71E-01(8.37E-05)† | 6.54E-01(2.37E-05)† | 7.00E-01(4.13E-05)† | 1.57E-01(1.53E-05)† | 7.76E-01(1.44E-05) | 7.75E-01(4.24E-05) |
| WFG7 | 7.37E-01(2.25E-05)† | 6.16E-01(1.01E-04)† | 7.34E-01(1.65E-05)† | 7.00E-01(2.79E-05)† | 7.93E-01(6.58E-07)† | 7.95E-01(3.38E-07) |
| WFG8 | 7.37E-01(1.09E-05)† | 6.11E-01(8.64E-05)† | 7.35E-01(1.31E-05)† | 6.99E-01(3.96E-05)† | 7.93E-01(1.03E-06)† | 7.96E-01(4.46E-07) |
| WFG9 | 7.29E-01(1.71E-05)‡ | 4.84E-01(3.90E-03)† | 7.28E-01(2.10E-05)‡ | 6.70E-01(7.90E-03) | 7.24E-01(1.16E-02) | 6.24E-01(1.16E-02) |
| +-= | 11/2/0 | 9/3/0 | 9/3/0 | 11/1/1 | 8/1/3 | — |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between KTDMOEA and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of KTDMOEA, respectively. The best mean value is highlighted with gray background.

Besides, it can be concluded from Table 5.18 that on the changing sequence as shown in Equation (5.10), KTDMOEA significantly performs the best among all compared algorithms. Specifically, KTDMOEA performs best on all problems except for F1-F4, WFG3 and WFG9. KTDMOEA significantly performs worse than MOEAD on F1 and F3. The reason is the same as that analyzed in Section 5.4.2.1, which is that KTDMOEA is unable to delete nondominated solutions far away from the PF on problems with multimodality and variable correlation. The proposed KTDMOEA is unable to provide enough diversity On F4 and WFG3 which needs more diversity to spread the population, and the reason is the same as previously analyzed in Section 5.4.2.1. As for WFG9 which is the most complex problem with several problem features in the fitness landscape, KTDMOEA performs significantly worse than NSGA2 and DNSGA2. The reason is that for WFG9, an algorithm is required to converge fast. However, when decreasing the NObj from 7 to 2, the introduced diversity by PS contraction is so much that it puts an obstacle to KTDMOEA's convergence, making KTDMOEA has worse convergence than NSGA2 and DNSGA2. In addition, PS contraction cannot increase population convergence, especially on complex problems like WFG9.

### 5.4.5 Impact of Algorithm Parameters

In the process of PS expansion, there is a parameter $\theta$ to set which is the number of solutions to generate along each expansion direction. In this section, different values of this parameter will be set to verify whether different parameter settings affect the performance of KTDMOEA.

All experimental settings are set the same as in Section 5.3.3 except for the frequency of change and the metric, which is set as 25 and MHV, respectively, to save space. The changing sequence is that the NObj firstly increases from 2 to 7 and then decreases from 7 to 2 one by one. There are three KTDMOEAs (denoted as KTDMOEA-1, KTDMOEA-2 and KTDMOEA-4), which has the value of $\theta$ as 1, 2 and 4, respectively. In order to verify whether different parameter settings affect the performance of KTDMOEA, the Friedman and Nemenyi tests on 5 state-of-the-arts and 3 KTDMOEAs are conducted to indicate the significant differences among them. The MHV values that all algorithms get on one problem with one frequency of change at one independent run of 31 runs is regarded as an observation of the test. Therefore, there are 403 (13 problems and 1 frequency of changes and 31 independent runs) observed data.

Figure 5.20 presents the Nemenyi post-tests results among MHV values of optimized solutions at the last generation by 8 algorithms in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one. Friedman detects significant differences in average value for MHV with a p-value of 1.7925e-281.

It is clear that the three KTDMOEAs get the best MHV values among all algorithms. It can be found from Figure 5.20 that there is no significant difference among the three KTDMOEAs with different setting of the parameter $\theta$. These results have verified that the performance of the proposed PS expansion/contraction is not sensitive to the setting of the parameter $\theta$. The performance of the proposed KTDMOEA against the existing algorithms is not affected by the setting of different parameter values.

Table 5.19 presents results of mean and deviation values for MHV of optimized solutions in the changing sequence of firstly increasing from 2 to 7 and then decreasing it from 7 to 2 by all 8 algorithms (5 state-of-the-arts and three KTDMOEAs).

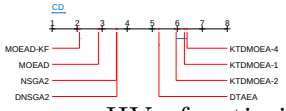



Figure 5.20: Friedman ranking among HV of optimized solutions at the last generation by 6 state-of-the-arts and 3 KTDMOEAs with different values of parameters theta (1, 2 and 4) in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

Table 5.19: Mean and standard deviation values of MHV metric for optimized solutions obtained by 5 state-of-the-arts and three KTDMOEAs with different values of parameters ($\theta = 1$, 2 and 4) at the last generation of all changes on all DMOPs with a changing NObj, when firstly decreasing the NObj from 7 to 2 and then increasing it from 2 to 7, both one by one

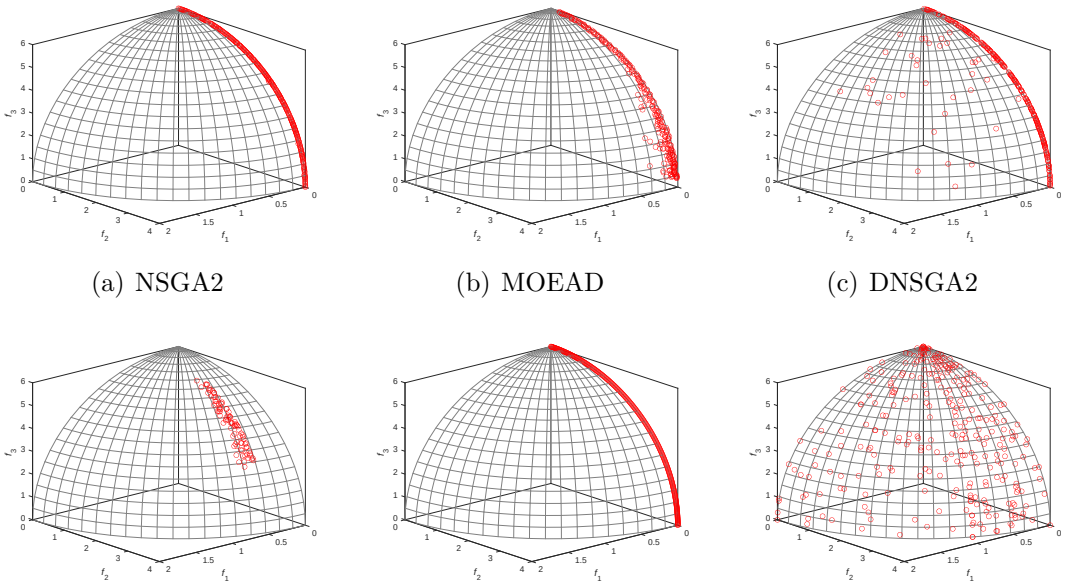
| Problems | NSGA2 | MOEAD | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA-1 | KTDMOEA-2 | KTDMOEA-4 |
|---|---|---|---|---|---|---|---|---|
| DTLZ1 | 6.08E-01(2.07E-03) | 9.88E-01(5.08E-05) | 5.87E-01(9.39E-04) | 8.64E-02(3.34E-04) | 9.75E-01(9.87E-04) | 9.96E-01(6.82E-08) | 9.96E-01(9.75E-10) | 9.96E-01(2.36E-08) |
| DTLZ2 | 9.38E-01(3.44E-06) | 9.53E-01(6.22E-07) | 9.33E-01(1.67E-05) | 2.80E-01(5.38E-04) | 9.58E-01(6.69E-08) | 9.58E-01(9.02E-09) | 9.58E-01(1.15E-09) | 9.58E-01(9.33E-10) |
| DTLZ3 | 5.71E-01(6.49E-04) | 9.27E-01(6.77E-05) | 5.54E-01(5.72E-04) | 2.08E-02(1.82E-04) | 8.86E-01(3.76E-03) | 9.58E-01(2.73E-06) | 9.56E-01(1.58E-06) | 9.58E-01(1.41E-06) |
| DTLZ4 | 9.10E-01(1.07E-04) | 9.36E-01(1.02E-04) | 8.95E-01(7.57E-05) | 9.39E-01(1.24E-05) | 9.58E-01(3.12E-08) | 8.82E-01(7.89E-04) | 8.89E-01(4.17E-04) | 8.82E-01(7.78E-04) |
| WFG1 | 9.51E-01(1.14E-05) | 8.86E-01(2.85E-05) | 9.47E-01(1.14E-05) | 8.10E-01(1.74E-08) | 9.54E-01(1.21E-05) | 9.70E-01(8.45E-07) | 9.70E-01(1.18E-06) | 9.69E-01(3.87E-06) |
| WFG2 | 9.63E-01(5.44E-07) | 9.28E-01(5.60E-06) | 9.63E-01(1.76E-07) | 2.42E-01(2.11E-09) | 9.67E-01(2.07E-08) | 9.68E-01(6.87E-08) | 9.68E-01(2.36E-09) | 9.68E-01(2.20E-08) |
| WFG3 | 7.03E-01(1.49E-05) | 6.84E-01(1.10E-05) | 7.11E-01(1.15E-06) | 2.53E-01(1.65E-08) | 6.94E-01(1.06E-05) | 6.84E-01(8.75E-04) | 6.89E-01(7.10E-04) | 7.02E-01(2.65E-06) |
| WFG4 | 7.45E-01(2.25E-05) | 7.00E-01(2.49E-05) | 7.21E-01(3.57E-05) | 3.47E-01(2.85E-05) | 8.28E-01(5.53E-07) | 8.31E-01(8.12E-08) | 8.30E-01(2.10E-07) | 8.30E-01(1.00E-07) |
| WFG5 | 7.05E-01(3.01E-05) | 7.00E-01(1.23E-05) | 7.00E-01(2.56E-05) | 5.35E-01(3.14E-06) | 7.85E-01(9.78E-07) | 7.87E-01(9.39E-07) | 7.86E-01(1.48E-06) | 7.87E-01(1.33E-06) |
| WFG6 | 6.90E-01(1.63E-04) | 6.85E-01(1.95E-05) | 7.11E-01(1.71E-04) | 1.46E-01(1.65E-05) | 7.56E-01(1.77E-03) | 7.61E-01(3.95E-04) | 7.61E-01(4.48E-04) | 7.60E-01(5.37E-04) |
| WFG7 | 7.66E-01(1.56E-05) | 6.82E-01(2.69E-05) | 7.59E-01(7.21E-06) | 7.23E-01(3.10E-05) | 8.31E-01(1.01E-07) | 8.32E-01(3.44E-08) | 8.27E-01(3.03E-04) | 8.32E-01(1.35E-08) |
| WFG8 | 7.66E-01(1.37E-05) | 6.83E-01(3.27E-05) | 7.60E-01(1.09E-05) | 7.21E-01(3.49E-05) | 8.31E-01(6.00E-08) | 8.32E-01(1.75E-08) | 8.28E-01(2.09E-04) | 8.32E-01(2.10E-08) |
| WFG9 | 4.24E-01(7.02E-04) | 4.10E-01(8.64E-05) | 5.47E-01(4.91E-03) | 7.01E-01(6.71E-03) | 4.92E-01(6.23E-05) | 4.46E-01(1.32E-03) | 4.34E-01(1.31E-03) | 4.62E-01(7.82E-04) |

In this table, three algorithms with best three MHV values are highlighted. It is clear from Table 5.19 that three KTDMOEAs with different $\theta$s gets the top three MHV values on all problems exclude F4, WFG3 and WFG9, which is consistent to the result in Section 5.4.2.1.

## 5.5 Summary

It has been investigated in this Chapter that existing work (DTAEA) cannot handle well DMOPs with a changing NObj and more complex PF shapes (convex, discontinuous and mixed shape of convex and concave) and fitness landscapes (non-separability and deceptiveness). The main reason is the lack of sufficient population diversity right after dynamic changes. To solve this problem, two research questions were studied, first how to transfer knowledge so as to enhance diversity and second how the knowledge transfer helps optimization. In order to answer both research questions, inspired by the characteristic of DMOPs with a changing NObj, a dynamics handling strategy–PS expansion/contraction was proposed and a new algorithm, KTDMOEA, was designed to make use of this strategy. Experimental studies have demonstrated the effectiveness of the proposed knowledge transfer in enhancing the diversity right after changes and assisting the optimization under different number of objective changing sequences.

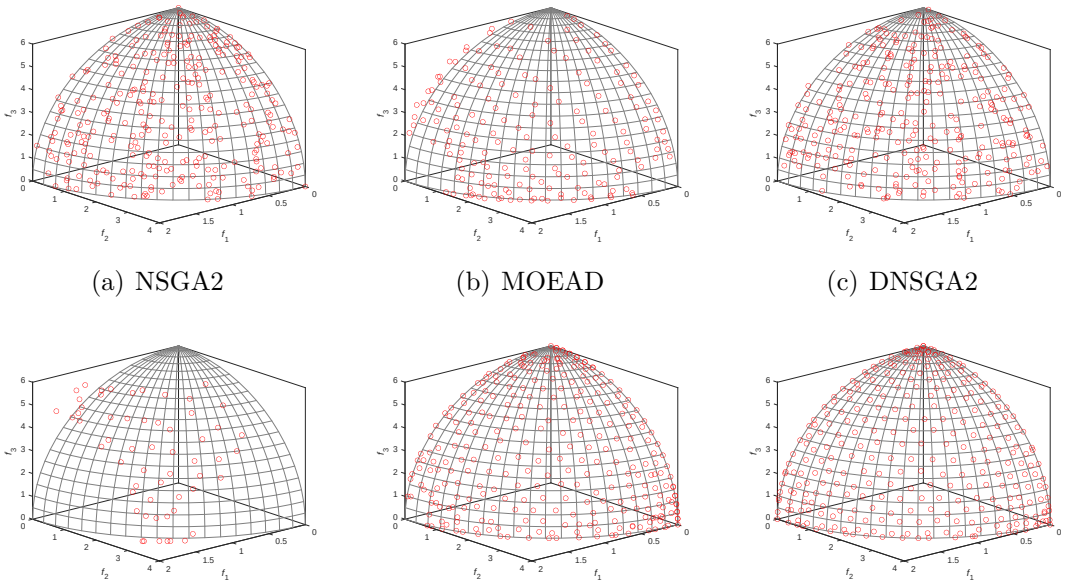
In comparison with the state-of-the-art in solving DMOPs with a changing NObj, our KTDMOEA achieved the best performance according to MHV, MGD and MMS metrics across a number of test functions. We also argue that it is important to use both DTLZ and WFG functions to build dynamic benchmarks because they provide rather different problem characteristics. KTDMOEA performed well on different problems under different parameter settings and frequencies of change.



(a) NSGA2 (b) MOEAD (c) DNSGA2

(d) MOEAD/KF (e) DTAEA (f) KTDMOEA

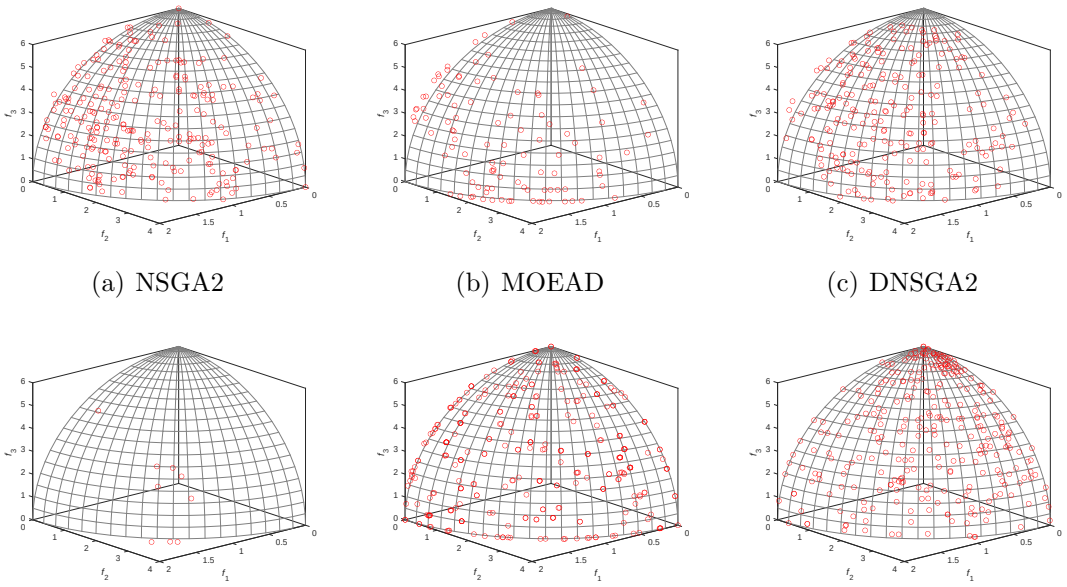
Figure 5.21: Distribution of populations obtained by all compared algorithm on WFG4 *at the first generation right after changes* when increasing the NObj from 2 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1).



(a) NSGA2 (b) MOEAD (c) DNSGA2

(d) MOEAD/KF (e) DTAEA (f) KTDMOEA

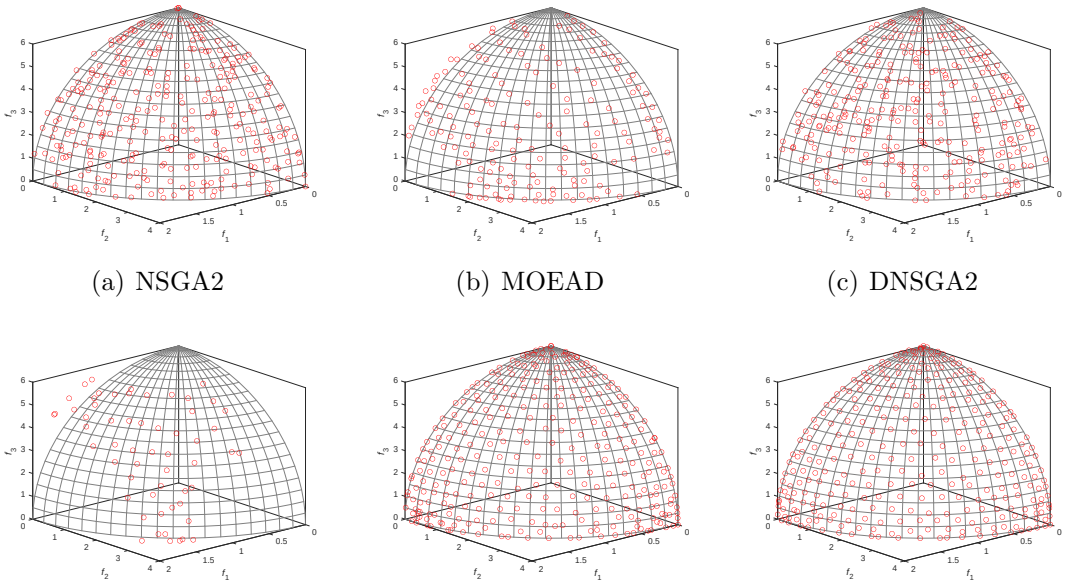
Figure 5.22: Distribution of populations obtained by all compared algorithm on WFG4 *at the last generation after optimization* when increasing the NObj from 2 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1).



(a) NSGA2 (b) MOEAD (c) DNSGA2

(d) MOEAD/KF (e) DTAEA (f) KTDMOEA

Figure 5.23: Distribution of populations obtained by all compared algorithm on WFG4 *at the first generation right after changes* when decreasing the NObj from 4 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1).



(a) NSGA2 (b) MOEAD (c) DNSGA2

(d) MOEAD/KF (e) DTAEA (f) KTDMOEA

Figure 5.24: Distribution of populations obtained by all compared algorithm on WFG4 *at the last generation after optimization* when decreasing the NObj from 4 to 3 and $\tau_t = 25$ in the changing sequence of equation (6.1).

As expected, no algorithm would be the best on all possible problems. According to the details in the experimental results, our proposed knowledge transfer strategy performed worse than DTAEA on problem with strong bias right after increasing the NObj and the problem with several extremely complex problem features (deceptiveness, bias, multimodality, etc) together. Therefore, there might be other methods beyond the heuristic, such as learning methods that is able to overcome the limitations of the existing heuristic method, which will be studied in the next Chapter.

# Chapter 6

# Learning to Expand and Contract PSs in DMO with a Changing NObj [1]

As one of the most recent transfer approaches for solving DMOPs with a changing NObj, dynamic two archive evolutionary algorithm (DTAEA) [3] was computationally revealed in Section 5.1 of Chapter 5 to lose diversity when solving problems with complex features in PF shape (convex, discontinues and mixed shape of convex and concave) and fitness landscape (nonseparability and deceptiveness). In order to address this limitation, knowledge transfer dynamic multi-objective evolutionary algorithm (KTDMOEA) was proposed in Section 5.2 of Chapter 5 to enhance diversity via heuristic rules for PS expansion or contraction, after considering the assumption that changing NObj results in PS expansion and contraction. The heuristic rules are the methods of heuristically searching for expansion and contraction directions as defined in Algorithm 5.2 of Section 5.2.1.1 and Algorithm 5.3 of Section 5.2.1.2. Computational studies demonstrated that KTDMOEA performed better than DTAEA on most problems.

However, we will specifically show that knowledge transfer via PS expansion in KTDMOEA is unable to increase diversity on problems with extremely strong bias directly after increasing the NObj in Section 6.1 of this Chapter, since the heuristic used by KTDMOEA for setting the PS expansion direction fails to work for this kind of problems. In addition, we will also show that, when decreasing the NObj, solutions transferred by KTDMOEA have poor convergence on problems with multi-modality and variable correlation in Section 6.1 of this Chapter. The reasons are that the multi-modality and variable correlation cause solutions previously converged in the old environment to be far away from the true PF after decreasing the NObj, and solutions transferred along heuristically found contraction directions in PS contraction of KTDMOEA only aim at improving population diversity without any improvement on convergence.

[1] This chapter is written based on content from the submitted paper as [*4] in Section 1.4.

According to our experience of using PS expansion and contraction in KTDMOEA to solve DMOPs with a changing NObj, it is essential to obtain expansion and contraction directions first directly after changes when PSs are expanded and contracted. However, it is almost impossible to obtain correct expansion and contraction directions directly without doing any exploration on the new problem directly after changes, since no prior knowledge is available before the exploration about the new problem. Therefore, given the limitation of heuristic rules in finding PS expansion and contraction directions of KTDMOEA, we could learn candidate expansion and contraction directions using data mining approaches.

Therefore, in this chapter, we aim to answer **Research question 4 (RQ4)** presented in Section 1.2: "How to learn the PS expansion and contraction directions when solving DMOPs with a changing number of objectives, so as to expand and contract PS right after changes and further help the optimization itself?" To answer this research question, the following three sub-research questions will be answered:

1. How to learn a set of potentially good candidate directions for PS expansion and contraction, respectively?
2. How to select correct expansion and contraction directions from their candidates, so as to expand and contract PS for enhancing knowledge transfer directly after changes?
3. How does learned PS expansion and contraction help the optimization process after the changes?

It is known that Principal Component Analysis (PCA) is able to learn the principal components of a dataset, representing key characteristics of this dataset. In addition, PS of a problem with $m(t)$ NObj is a (m(t)-1)-d manifold. Then, PCA could be used to learn the principal components of the manifold that represents the data structure of the PS. Lastly, the promising expansion directions are those directions perpendicular to the principal components when increasing the NObj. The reason why they are promising is that there are some directions perpendicular to the lower dimension manifold to make solutions to expand to the higher dimension manifold. These principal components are the promising contraction directions when decreasing the NObj. The reason why they are promising is that the contraction directions are within the manifold.

Therefore, in order to answer these sub-research questions, we firstly propose to conduct Principal Component Analysis (PCA) on PS of the previous problem to get eigenvectors. Then, when increasing the NObj, we learn candidate expansion directions from those directions perpendicular to these eigenvectors, since these directions might be the best ones that make the manifold of the old PS to expand; when decreasing the NObj, we learn candidate contraction directions from these eigenvectors. Lastly, we select the correct expansion and contraction directions by

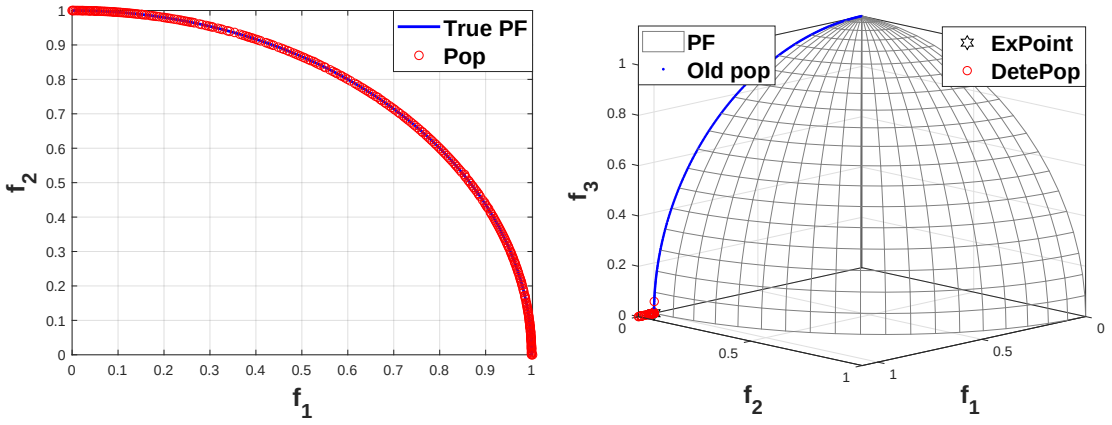
determining which candidate expansion helps diversity and which contraction helps convergence, so as to enhance the diversity and convergence of expanded and contracted PS, respectively. Note the convergence and diversity of solutions would be difficult to be accurately measured. Here, we use the domination relationship between solutions to roughly measure the convergence and diversity of generated solutions along the directions. Experimental studies based on 13 DMOPs with a changing NObj, modified from 4 DTLZ [60] and 9 WFG [100] problems, demonstrate the effectiveness of our proposed approach in terms of solution quality directly after changes and after optimization of different generations.

Note that the knowledge transfer method used in LEC is a solution-based transfer optimization algorithm, which transfers the solutions obtained in the previous environment to the new environment to make them have good solution quality in the new environment. The transfer optimization method finds expansion and contraction directions based on PCA and then transfers the optimal solutions obtained in the previous environment along these directions. The distinction of LEC and existing transfer methods is that LEC makes use of the idea of transfer by PCA, rather than using a specific transfer learning algorithm (e.g., transfer component analysis, manifold transfer learning and TrAdaboost) from machine learning like the existing transfer algorithms [25–27, 81].

The organization of this Chapter is described as follows. Section 6.1 investigates the limitations of the state-of-the-art transfer algorithm (KTDMOEA) proposed in Chapter 5 for solving DMOPs with a changing NObj. It shows that KTDMOEA lacks sufficient diversity on the problem with extremely strong bias right after increasing the NObj and loses convergence on problems with multimodality and variable correlation right after decreasing the NObj. The proposed knowledge transfer-based algorithm is elaborated in Section 6.2. Section 6.3 describes the specific experimental setup. The experimental results are presented in detail in Section 6.4. Finally, Section 6.5 summarizes this chapter.

## 6.1 A Further Computational Investigation of KTDMOEA

Even though KTDMOEA has been computationally demonstrated in Section 5.4 of Chapter 5 to be effective on DMOPs with a changing NObj, it has limitations on some problems. It has been analyzed in Section 5.4.1.4 that KTDMOEA lacks diversity on the problem with extremely strong bias right after increasing the NObj. We will conduct a very detailed investigation to reveal this limitation in this Section. Moreover, we also find that our proposed KTDMOEA loses convergence on problems with multi-modality and variable correlation right after decreasing the NObj, which will also be elaborated in this section.



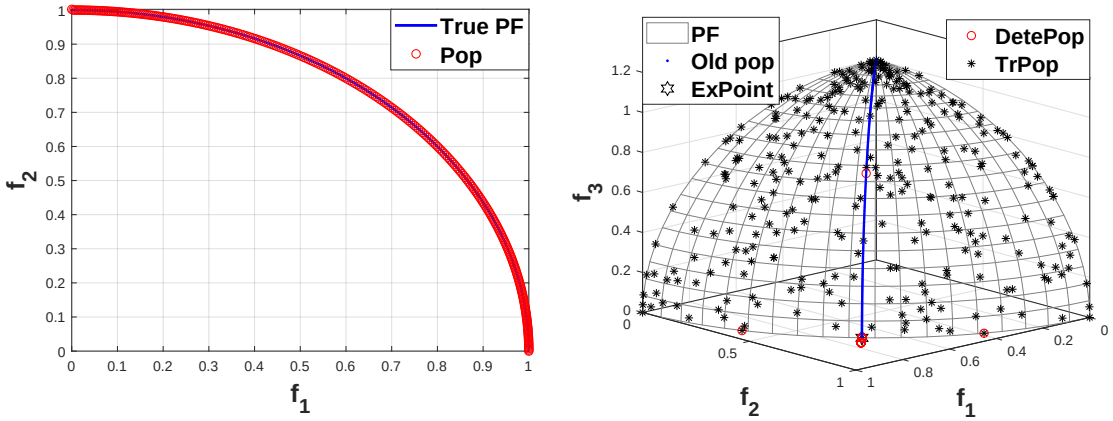

(a) Approximation on bi-F4 (b) After increasing one NObj

Figure 6.1: (a) Distribution of approximated population given 1000 generations on bi-F4; (b) Distribution of the old population ('Old pop'), i.e. reevaluating 'pop' obtained in (a) on the tri-F4, the extreme point ('ExPoint'), the detective population ('DetePop'), all used to find the expansion directions. Transferred solutions by KTDMOEA on F4 are still on the curve of 'Old pop', thus not being plotted to avoid solutions duplication, and failing to spread through the PF.

In order to analyze the limitations of KTDMOEA, F4 with extremely strong bias and F3 with multi-modality and variable correlation, from the DTLZ suite [3], are selected as examples to conduct an experimental investigation of how the quality of solutions transferred by KTDMOEA is after increasing and decreasing NObj, respectively. At the same time, another problem F2 which has the same PF shape as F3-F4 and no problem feature in the fitness landscape was also selected from the DTLZ suite [3] as a base comparison.

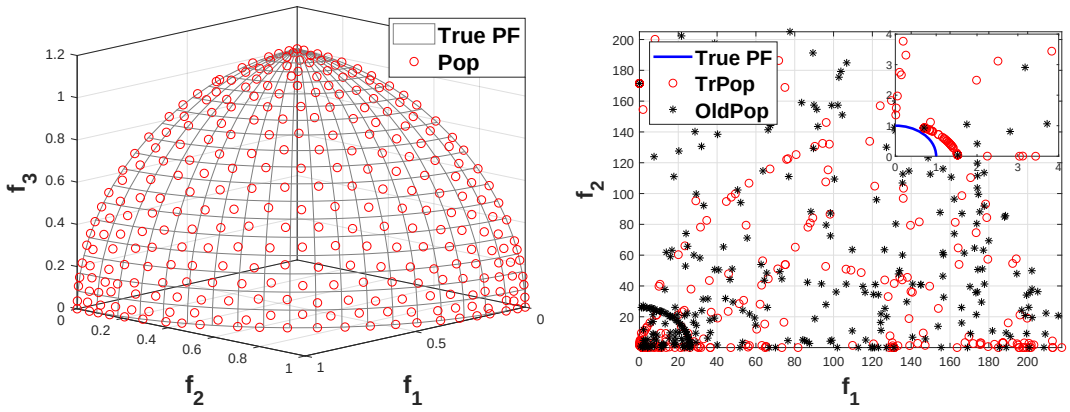
Specifically, when increasing NObj, problems F2 and F4 are set as bi-objective problem and then given 1000 generations by KTDMOEA to make population approximate true PF well before increasing the NObj from 2 to 3. Similarly, when decreasing the NObj, problems F2 and F3 are set as a tri-objective problem and then given 1000 generations by KTDMOEA to make population approximate true PF well before decreasing the NObj from 3 to 2. In these two examples, other parameters are the same as in KTDMOEA, as shown in Section 5.3 of Chapter 5.

Figures 6.1 and 6.2 show the distribution of approximated population given 1000 generations on bi-objective problems and solutions transferred by PS expansion of KTDMOEA at the first generation right after increasing NObj from 2 to 3 on F4 and F2, respectively. As shown in Figures 6.1(b) and 6.2(b), the extreme point ('ExPoint') is firstly selected from the old population ('Old pop'). A detective population ('DetePop') is then randomly generated around the extreme point in the solution space. Solutions dominated by the old population and solutions located in the same subspace as any solution of the old population will be deleted from the detective population. The extreme point and the remaining solutions in the detective population are connected to form the expansion directions by regarding



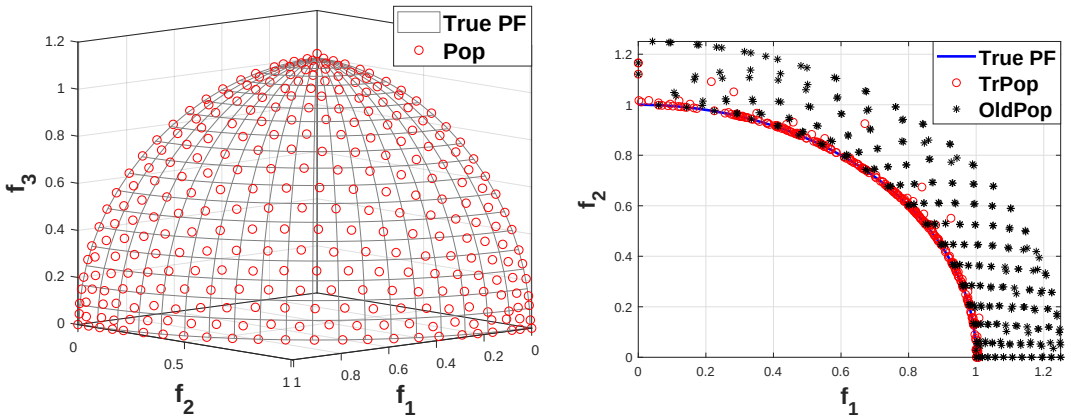

(a) Approximation on bi-F2 (b) After increasing one NObj

Figure 6.2: (a) Distribution of approximated population given 1000 generations on bi-F2; (b) Distribution of the old population ('Old pop'), i.e. reevaluating 'pop' obtained in (a) on the tri-F2, the extreme point ('ExPoint'), the detective population ('DetePop'), all used to find the expansion directions in the objective space, and transferred population('TrPop') via PS expansion in KTDMOEA.




(a) Approximation on tri-F3 (b) After decreasing one NObj

Figure 6.3: Distribution of approximated population given 1000 generations on tri-F3 and solutions transferred by PS contraction of KTDMOEA at the first generation right after decreasing NObj from 3 to 2 on F3.




(a) Approximation on tri-F2 (b) After decreasing one NObj

Figure 6.4: Distribution of approximated population given 1000 generations on tri-F2 and solutions transferred by PS contraction of KTDMOEA at the first generation right after decreasing NObj from 3 to 2 on F2.

the extreme point as the start of these directions.

However, as we can see in Figure 6.1(b), when observed in the objective space, there will be no solution in the detective population after removing solutions dominated by the old population and located in the same subspace as any solution of the old population, resulting in no expansion directions to be found. As a comparison, there are still two solutions in the detective population for F2 without the feature of strong bias, forming expansion directions together with the extreme point, which can be seen from Figure 6.2(b).

The reason is that the feature of strong bias in the fitness landscape of F4 hinders solutions with diversity to be generated in the detective population. More specifically, for most possible values of each variable ($x_i$) within the range [0,1], the intermediate function (i.e. $y = x_i^{100}$) containing the feature of strong bias which forms the objective function of F4 maps them to the value of nearly zero. Therefore, most of randomly generated solutions around the extreme point would have the same objective values to that of the extreme point, thus resulting in most of solutions duplicated in the extreme point, as show in 6.1(b).

Similarly, Figures 6.3 and 6.4 show the distribution of approximated population given 1000 generations on tri-objective problems and solutions transferred by PS contraction of KTDMOEA at the first generation right after decreasing NObj from 3 to 2 on F3 and F2, respectively. It can be seen from Figure 6.3(b) that the multimodality and variable correlation of F3 lead most of already converged solutions in the old environment (as shown in Figure 6.3(a)) to be far away from the true PF after decreasing the NObj, compared to the solutions distribution on F2 without these problem features shown in Figure 6.4(b).

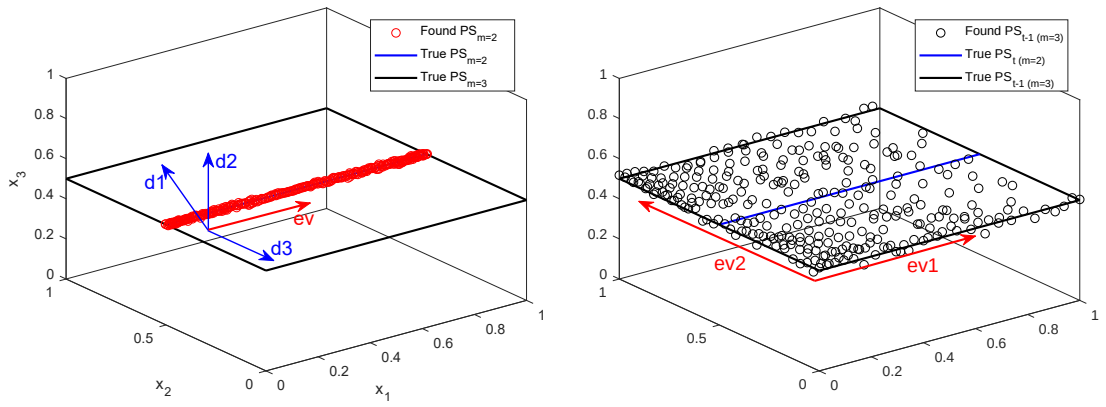
The heuristic rule in PS contraction of KTDMOEA only aims at improving pop-

ulation diversity via generating spread solutions along the inverse direction from one extreme point and its nearest solution, and even solutions between any two solutions from the nondominated solution set. This can be found from Figures 6.3(b)) and Figure 6.4(b)) that transferred solutions in $TrPop$ generated around the extreme point of black stars near the $f_2$ axis side improves spread of $OldPop$, and transferred solutions between any two solutions from the nondominated solution set of $OldPop$ indeed increase the even distribution of $OldPop$. Therefore, the heuristic rule in PS contraction tends to improve diversity and maintain convergence, which is unable to increase population convergence when decreased NObj degenerates population convergence on problems with the problem feature of multi-modality and variable correlation.

## 6.2 Learning to Expand and Contract PSs (LEC)

Given the limitations of heuristic rules in KTDMOEA when expanding and contracting PS regarding unable to determine expansion directions for problems with extremely strong bias and improve population convergence for problems with variable correlation and multi-modality, it is essential to learn how to expand and contract the PS. Therefore, in this section, we present our proposed knowledge transfer-based strategy, i.e. learning to expand and contract PSs (LEC). This strategy contains three procedures that need to be carried out sequentially, i.e. (1) learning candidate expansion or contraction directions, (2) selection of the most promising directions among these candidate expansion or contraction directions and (3) PS expansion or contraction. Specifically, Section 6.2.1 describes how to learn the candidate expansion and contraction directions under the assistance of PCA. Section 6.2.2 presents the methodology of selecting the most promising expansion and contraction directions according to which candidate expansion direction improves diversity and which contraction direction improves convergence. The process of how to expand and contract PS after obtaining the most promising expansion and contraction directions will be elaborated in Section 6.2.3.

Obviously, Section 6.2.1 aims to answer the first research question. Sections 6.2.2 and 6.2.3 answer the second research question. Note that our proposed LEC is conducted directly after changes; after LEC is conducted when the environment remains unchanged, the population can be further optimized by any existing algorithm. In our proposed LEC, we use the same optimization algorithm as that of KTDMOEA to evolve the population.




(a) Learning expansion directions (b) Learning contraction directions

Figure 6.5: Method of learning the candidate expansion and contraction directions for increasing and decreasing NObj, respectively: (a), **ev** is the eigenvector of the found $PS_{m=2}$ and **d**1 to **d**3 are three directions that are perpendicular to $PS_{m=2}$, (i.e. **ev** * $\mathbf{d}_i = 0$, where $i = 1, 2, 3$); **d**1 to **d**3 are the three candidate expansion directions (b), **ev1** and **ev2** are two eigenvectors of $PS_{m=3}$, which are also the candidate contraction directions.

### 6.2.1 Learning Candidate Expansion and Contraction Directions via Principal Component Analysis

Considering the limitation of heuristic rules in finding expansion and contraction directions of KTDMOEA, we propose to firstly learn candidate expansion and contraction directions under the assistance of PCA, as given in Section 6.2.1.1 and Section 6.2.1.2 in detail, respectively.

#### 6.2.1.1 Learning Candidate Expansion Directions

This section aims at learning candidate expansion directions from PS before the change under the assistance of PCA.

The idea of learning the candidate expansion directions is illustrated in Figure 6.5(a). Note this figure is just drawn to illustrate the process of learning the expansion directions, the specific solutions in real problems may be different. As shown in the figure, suppose the red circle points are the found $PS_{m=2}$ before the change; **ev** is the eigenvector obtained by conducting principal component analysis on $PS_{m=2}$. Then, $d1$ to $d3$ are three of many candidate expansion directions, which are perpendicular to the $PS_{m=2}$.

It is a well-known theory in geometry that two intersecting lines in space determine a plane. Assume $PS_{m=3}$ manifold is a plane, then $PS_{m=2}$ manifold could be

**Algorithm 6.1: Learning Candidate Expansion Directions**

**Input:** PS at time step $t$ before the change ($\mathbf{PS}_t$); re-evaluated solutions of $\mathbf{PS}_t$ on the new problem after the change (i.e. time step $t+1$) **OldP**; population size $N$; the NObj for the old problem before the change $m(t)$; dimension of the problem $n$;

**Output:** Set of candidate expansion directions $\mathbf{D}_p$ or NULL.

1 Conduct principal component analyses on $\mathbf{PS}_t$ to get $m(t)$-1 eigenvectors as $\mathbf{ev}_i$, where $i = 1, ..., m(t) - 1$;

2 Suppose $\mathbf{D}_p = (\mathbf{D}_p^1, ..., \mathbf{D}_p^j, ..., \mathbf{D}_p^N)$ is a set of $N$ vectors, which are perpendicular to these eigenvectors (i.e. $\mathbf{ev}_i$), where each vector in $\mathbf{D}_p$ is initialized with null values, i.e. $\mathbf{D}_p^j = (d_j^1, ..., d_j^n) = (null, ..., null)$ where $d_j^1$ to $d_j^n$ are the variables we want to obtain from steps 3 to 8;

3 Form the system of homogeneous linear equations: $\mathbf{D}_p * \mathbf{ev}_i = 0$ with $m(t) - 1$ number of linear equations, where $i = 1, ..., m(t) - 1$. Note there are $n$ variables while only $m(t) - 1$ equations, other variables (with the number of $n - m(t) + 1$) are therefore needed to be sampled;

4 Sample $N$ vectors as **Vec** with the dimension of $n - m(t) + 1$ using Latin hypercube sampling with all elements of each vector within [0,1] ;

5 **for** *$j$ = 1 to $N$* **do**

6 Assign $\mathbf{Vec}_j$ to the first $n - m(t) + 1$ elements of $\mathbf{D}_p^j$, i.e. $d_j^k = \mathbf{Vec}_j^k$, where $k = 1, ..., n - m(t) + 1$ ;

7 Calculate the value of $d_j^{n-m(t)+2}$ to $d_j^n$ via putting $d_j^1$ to $d_j^{n-m(t)+1}$ to the system of homogeneous linear equations constructed in line 3;

8 Convert $\mathbf{D}_p^j$ to a unit vector using $\mathbf{D}_p^j = \frac{(d_j^1,...,d_j^n)}{\|(d_j^1,...,d_j^n)\|}$ ;

**Return** $\mathbf{D}_p$.

a segment line in this plane, since $PS_{m=2}$ is a subset of $PS_{m=3}$ [3], where the $PS$ manifold is the location of Pareto-optimal solutions in the solution space. According to the theory of two intersecting lines determining a plane, there are infinite lines in the PS manifold of tri-objective problem that have intersections with the eigenvector of the $PS_{m=2}$, since $PS_{m=3}$ manifold is the expansion of $PS_{m=2}$ manifold along certain directions. However, there is only one unit vector and its inverse vector on $PS_{m=3}$ that are perpendicular to the eigenvector of the $PS_{m=2}$. Therefore, only the directions perpendicular to the $PS_{m=2}$ are selected as the candidate expansion directions, so as to decrease the number of candidate expansion directions.

LEC will create a set of size $N$ containing evenly sampled perpendicular vectors to the $PS$ of the previous environment as the candidate expansion directions. The framework of learning candidate expansion directions is exhibited in Algorithm 6.1. Given the Pareto optimal solution set at the time step t ($\mathbf{PS}_t$) before the change, the first step of learning the candidate expansion directions is conducting principal component analysis on $\mathbf{PS}_t$, to get the eigenvectors ($\mathbf{ev}s$), as shown in line 1 of Algorithm 6.1. The number of the eigenvectors is $m(t)-1$, as the $PS$ of a continuous

multiobjective optimization problem with $m(t)-1$ is a piecewise continuous $m(t)-1$-D manifold [3] [106]. In line 2, suppose $\mathbf{D}_p = (\mathbf{D}_p^1, ..., \mathbf{D}_p^j, ..., \mathbf{D}_p^N)$ is a set of $N$ vectors, which are perpendicular to these eigenvectors (i.e. $\mathbf{ev}_i$), where each vector in $\mathbf{D}_p$ is initialized with null values, i.e. $\mathbf{D}_p^j = (d_j^1, ..., d_j^n) = (null, ..., null)$, where $d_j^1$ to $d_j^n$ are the variables we want to obtain from steps 3 to 8.

Afterwards, a system of homogeneous linear equations is constructed in line 3 by the fact that the product of two orthogonal vectors (i.e. **ev** and candidate expansion directions $\mathbf{D}_p^j$) is zero. Note the number of homogeneous linear equations is $m(t)-1$, since there are only $m(t)-1$ eigenvectors (i.e. $\mathbf{ev}_i$s, $i = 1, ..., m(t)-1$). Now, there are only $m(t)-1$ homogeneous linear equations while $n$ variables (i.e. $d_i^1$ to $d_i^n$) to get. It is known that there are infinite solutions for this set of linear equations. To support the creation of directions that evenly cover the space, we create a set **Vec** containing $N$ evenly sampled vectors with dimension $n-m(t)+1$ within a unit hypercube. To make solutions of the $m(t)-1$ homogeneous linear equations with $n$ variables finite, the values of $n-m(t)+1$ variables must be assigned beforehand, which are from **Vec**.

The for loop from lines 5 to 8 calculates values of $d_j^1$ to $d_j^n$ for $\mathbf{D}_j$ to construct $N$ vectors ($\mathbf{D}_p^1$ to $\mathbf{D}_p^N$) and converts them to unit vectors as candidate expansion directions. Specifically, line 6 puts the values of the $j$-th sampled vector ($\mathbf{Vec}_j$) into the first $n-m(t)+1$ positions of the $j$-th vector of $\mathbf{D}_p$ (i.e. $\mathbf{D}_p^j$). Then, line 7 calculates values of $d_j^{n-m(t)+2}$ to $d_j^n$ after putting $d_j^1$ to $d_j^{n-m(t)+1}$ in the system of $m(t)-1$ homogeneous linear equations. Now, all values of $d_j^1$ to $d_j^n$ in $\mathbf{D}_p^j$ are obtained. Then line 8 converts $\mathbf{D}_p^j$ to a unit vector. Once all $N$ vectors in $D_p$ are created, they are returned as the candidate expansion directions in Line 9.

#### 6.2.1.2 Learning Candidate Contraction Directions

This section aims at learning candidate contraction directions from the PS that was available right before the change based on PCA.

**Algorithm 6.2: Learning Candidate Contraction Directions**

**Input:** PS at time step $t$ before the change ($\mathbf{PS}_t$)
**Output:** Set of the learnt candidate contraction directions $\mathbf{C}_{con}$ or NULL.
1 Conduct principal component analyses on $\mathbf{PS}_t$ to get $m(t)$-1 eigenvectors as $\mathbf{ev}_i$, where $i = 1, ..., m(t)-1$;
2 Put these eigenvectors, i.e. $\mathbf{ev}_1$, ..., $\mathbf{ev}_{m(t)-1}$ in $\mathbf{C}_{con}$;
**Return** $\mathbf{C}_{con}$.

The idea of learning candidate contraction directions in the decision space is illustrated in Figure 6.5(b). Note this figure is just drawn to demonstrate the process of learning PS contraction directions, the specific cases in real problems may be

different. As shown in Figure 6.5(b), black points are the found $PS_{m=3}$ before the change; **ev1** and **ev2** are two eigenvectors of $PS_{m=3}$, which are regarded as the candidate contraction directions.

The method for learning candidate contraction directions is described in Algorithm 6.2. Given a set of Pareto optimal solutions at the time step t ($\mathbf{PS}_t$) before the change, the first step of learning the candidate contraction directions is conducting PCA on $\mathbf{PS}_t$, to get the eigenvectors (**ev**$s$), as shown in line 1 of Algorithm 6.2. Then, in line 2, regard these eigenvectors as the candidate contraction directions and put them in $\mathbf{C}_{con}$ to return.

**Algorithm 6.3: Most Promising Expansion Directions Selection**

**Input:** Candidate expansion directions $\mathbf{D}_p$; re-evaluated solutions of $\mathbf{PS}_t$ on the new problem after the change (i.e. time step $t+1$) **OldP**; population size $N$;

**Output:** Set of the most promising expansion directions $\mathbf{D}_{exp}$ or NULL.

1 Randomly sample a solution **x** from **OldP**;
2 **for** *i = 1 to N* **do**
3  Generate a solution **y** based on **x** and $\mathbf{D}_p^i$ using equation (5.4);
4  Evaluate **y** and compare it to all solutions in **OldP**;
5  **if** ***y*** *is dominated by any one solution in* ***OldP*** **then**
  continue;
  **else**
  Put $\mathbf{D}_p^i$ into $\mathbf{D}_{exp}$;

**Return** $\mathbf{D}_{exp}$.

### 6.2.2 Most Promising Expansion and Contraction Directions Selection

After learning the candidate expansion and contraction directions, the next step is to select the most promising among them. The most promising expansion directions are defined as those candidate expansion directions along which generated solutions are non-dominated. The most promising contraction directions are defined as those candidate contraction directions along which generated solutions are dominating.

#### 6.2.2.1 Selection of Expansion Directions to Improve Diversity

The selection criteria of the most promising expansion directions is to see which candidate direction does not result in dominated solutions by any solution in reevaluated $PS$ of the previous environment on the new problem. Since increasing the

NObj may lead to the expansion of PF [3], it is intuitive that population diversity increases after the expansion of PF. Therefore, solutions generated along the most promising expansion directions will improve population diversity.

The process of selecting most promising expansion directions from their candidates is presented in Algorithm 6.3. Given the set of candidate expansion directions $\mathbf{D}_p$ and re-evaluated solutions of $\mathbf{PS}_t$ on the new problem after the change (i.e. time step $t+1$) **OldP**, a solution $\mathbf{x}$ is firstly randomly sampled from **OldP** in line 1. For each candidate direction in $\mathbf{D}_p$, generate a solution $\mathbf{y}$ based on $\mathbf{x}$ and $\mathbf{D}_p^i$ using equation 5.4. Then, evaluate $\mathbf{y}$ and compare it to all solutions in **OldP**; if $\mathbf{y}$ is not dominated by any solution in **OldP**, then add this candidate direction to the set of expansion directions $\mathbf{D}_{exp}$.

#### 6.2.2.2 Selection of Contraction Directions to Improve Convergence

The selection criteria of the most promising contraction directions is to see which candidate direction results in solutions with better convergence. Since decreasing the NObj may lead to the contraction of PF/PS [3], solutions used to be optimal may not still be optimal after the change. Therefore, solutions moving from positions of past optimal solutions to the positions of new optimal solutions along the most promising contraction direction will improve population convergence to help them reach the new (i.e. contracted) PS/PF.

**Algorithm 6.4: Most Promising Contraction Directions Selection**

**Input:** Candidate contraction directions $\mathbf{C}_{con}$; re-evaluated solutions of $\mathbf{PS}_t$ on the new problem after the change (i.e. time step $t+1$) **OldP**; population size $N$; the NObj for the old problem before the change $m(t)$

**Output:** Set of the most promising contraction directions $\mathbf{D}_{con}$ or NULL.

1 Randomly sample a solution $\mathbf{x}$ from **OldP**;
2 **for** $i = 1$ *to* $m(t)-1$ **do**
3 Generate $\lfloor N/(m(t)-1) \rfloor$ solutions $\mathbf{y}_j$ $(j = 1, ..., \lfloor N/(m(t)-1) \rfloor)$, based on $\mathbf{x}$ and $\mathbf{C}_{con}^i$, using equation (5.4);
4 Put these generated solutions $(\mathbf{y}_j)$ into $\mathbf{Y}$ ;
5 Evaluate these generated solutions and compare them to $\mathbf{x}$;
6 **if** $\exists \boldsymbol{y}_j \in \boldsymbol{Y}$ *dominates* $\boldsymbol{x}$ **then**
Put $\mathbf{C}_{con}^i$ into $\mathbf{D}_{con}$.

**Return** $\mathbf{D}_{con}$.

The process of selecting the most promising expansion directions from their candidates is presented in Algorithm 6.4. Given the set of candidate expansion directions $\mathbf{C}_{exp}$ and re-evaluated solutions of $\mathbf{PS}_t$ on the new problem after the change (i.e. time step $t+1$) **OldP**, a solution $\mathbf{x}$ is firstly randomly sampled from **OldP** in

line 1 of Algorithm 6.4. Since PS of a continuous $m(t)$- objective MOP is an ($m(t)$ - 1)-dimensional piecewise continuous manifold in the decision space, there are ($m(t)$ - 1) candidate expansion directions in $\mathbf{C}_{exp}$. In order not to consume too much function evaluations, we make total generated solutions to find the most promising contraction directions not exceed the population size $N$. Therefore, for each candidate contraction direction in $\mathbf{C}_{con}$, generate $\lfloor N/(m(t)-1) \rfloor$ solutions $\mathbf{y}_j$ based on $\mathbf{x}$ and $\mathbf{C}^i_{con}$ using equation (5.4). Later on, in line 4, these generated solutions (i.e. $\mathbf{y}_j$) are put into the set $\mathbf{Y}$. Evaluate these generated solutions and compare them to $\mathbf{x}$; if there exist one solution $\mathbf{y}_j \in \mathbf{Y}$ dominating $\mathbf{x}$, then put $\mathbf{ev}_i$ into $\mathbf{D}_{con}$.

### 6.2.3 PS Expansion and Contraction

After determining the most promising expansion and contraction directions, the next step is to expand and contract PSs for enhancing diversity and convergence when increasing and decreasing NObj, respectively.

#### 6.2.3.1 Diversity Enhancing PS Expansion

The process of expanding PS to enhance diversity after determining the most promising expansion directions is the same as that of KTDMOEA, as shown in Section 5.2.1.1 of Chapter 5. The main idea is to evenly select some solutions from the $PS$ before the change as the base solutions, and then generate a new population based on these base solutions and the learnt expansion directions.

After determining the expansion directions $\mathbf{D}_{exp}$, generate $\theta$ (a parameter set by the user) solutions based on each evenly selected solution from **OldP** (number of selected solution can refer to Section 5.2.1.1 of Chapter 5) along each direction from $\mathbf{D}_{exp}$, following equation (5.4). If $\mathbf{P}_{exp}$ is not full, just evenly select solutions from $\mathbf{PS}_t$ in the objective space.

Note that in our proposed LEC, the learning-based PS expansion is conducted right after increasing the NObj. The expanded PS $\mathbf{P}_{exp}$ is regarded as the initial population for the optimization of the new problem.

#### 6.2.3.2 Convergence Enhancing PS Contraction

The process after determining the most promising contraction directions is to contract the PS along these learnt contraction directions for enhancing convergence. The main idea is to randomly select some solutions from the $PS$ before the change

as the base solutions, and then generate a new population based on these base solutions and the learnt contraction directions.

Randomly select $\lambda$ solutions from $PS_t$ and put them in the set $P_{base}$. The size of the learnt contraction directions $\mathbf{D}_{con}$ is $N_{con}$. Then, to make the number of generated solutions from each selected solution along each contraction direction equal and generated solutions not exceed the population size $N$, generate $N_g = \lfloor \frac{N}{N_{con} * \lambda} \rfloor$ solutions based on each solution in $\mathbf{P}_{base}$ along each contraction direction in $\mathbf{D}_{con}$, following equation (5.4) in Section 5.2.1.1, and put them into $\mathbf{P}_g$. Afterwards, in line 4, we combine evaluated $\mathbf{P}_g$ and $\mathbf{PS}_t$, and then select solutions with the number of $N$ as the contracted PS $\mathbf{P}_{con}$ using nondominated sorting and decomposition-based density estimation from Section 5.2.2 of Chapter 5.

Note that in our proposed LEC, the learning-based PS contraction is conducted right after decreasing the NObj. The contracted PS $\mathbf{P}_{con}$ is regarded as the initial population for the optimization of the new problem.

### 6.2.4 Time Complexity Analysis

The time complexity of LEC is discussed in this subsection for one generation. When increasing or decreasing NObj, the PS expansion or contraction is evoked. In the case of increasing NObj, the main complexity of PS expansion is cost in principal component analyses (line 1 of Algorithm 1) and solving systems of homogeneous linear equations with $m(t) - 1$ variables (line 7 of for loop between 5 and 8 of Algorithm 1) for learning candidate expansion directions. The former consumes $O(m(t)^2)$ and the latter costs $O(Nm(t)^2)$, where $m(t)$ is the number of objectives at time step $t$. As for the case of decreasing the NObj, the principal component analyses in line 1 of Algorithm 2 cost $O(m(t)^2)$; while the nondominated sorting and density estimation (Section **??**)) consume $O(N^2)$ and $O(N|W(t)|)$, respectively, where $W(t)$ is the number of weight vectors at time step $t$. As LEC utilizes the CA update mechanism from DTAEA to update the population, the population update in LEC costs $O(N^2)$ comparisons. In summary, assuming that $N > W(t)$ and $N > m(t)$, complexity of LEC in one generation is $\max(O(Nm(t)^2), O(N^2))$.

## 6.3 Experimental Setup

In this section, experimental studies are designed to verify whether our proposed knowledge transfer approach (i.e. LEC) answers the second research question mentioned at the beginning of this Chapter. Analyses will be carried out to reveal whether existing DMOEAs for DMOPs are able to deal with a changing NObj de-

spite not being designed to do so. These are important baselines.

### 6.3.1 Benchmark Problems

Two suites of multi-objective optimization test problems DTLZ [60] and WFG [100] are modified to be DMOPs with a changing NObj. Four DMOPs with a changing NObj from DTLZ1-DTLZ4 are renamed as F1-F4 following [3]. These two suites of benchmark functions are utilized to verify that the proposed algorithm is able to deal with problems with different problem features. Detailed descriptions of problems features can be found in Table 5.1 of Section 5.3.

The sequence of changes for these benchmark problems is: the NObj firstly increases from 2 to 7 one by one and then decreases from 7 to 2 one by one following [3];

$$m(t) = \begin{cases} 2, & t{=}0 \\ 3, & t{=}1 \\ m(t-1)+1, & t \in [2{,}5] \\ m(t-1)-1, & t \in [6{,}10], \end{cases} \tag{6.1}$$

where $t$ is the identifier of the change. It has 10 environmental changes.

### 6.3.2 Compared Algorithms

In our experimental studies, four algorithms are selected for the comparison, so as to verify the performance of our proposal against the state-of-the-arts. Two popular and state-of-the-art DMOEAs including DNSGA-II [7] and MOEA/D-KF [43] specifically designed for DMOPs with changing shapes and/or positions of PS and/or PF are selected as the baselines. These two algorithms are selected as the baselines due to their popularity and good performance on solving DMOPs with changing shapes and/or positions of PS and/or PF. In order to verify whether our proposed knowledge transfer approach (LEC) answers the research questions mentioned at the beginning of this chapter, as two of recently developed algorithms targeted for handling changes in the NObj, DTAEA [3] and KTDMOEA proposed in Chapter 5 are also chosen to be compared. Note that the optimization algorithm used within our LEC is the same as the one used in KTDMOEA due to its competitive results obtained in our previous paper on these problem instances. Section 5.3.2 presents the detailed descriptions of these algorithms.

### 6.3.3 Parameter Settings

The parameters of these compared algorithms are set as follows:

- Population size: $popsize = 300$, the same as that of DTAEA and KTDMOEA, $\theta$ in KTDMOEA is set as 2. The impact of $\theta$ on KTDMOEA's performance was analyzed in Section 5.4.5 of Chapter 5;
- Number of solutions $\lambda$ selected from the $\mathbf{PS}_t$ in learning to contract PS is set as $popsize/20$ according to experience. The impact of $\lambda$ on LEC's performance will be analyzed Section 6.4.4;
- Several different frequencies of change ($\tau_t$) are investigated to understand their effect on the algorithm: $\tau_t$ is set as 50 and 100 and 200; note a small value of $\tau_t$ means high frequency of change. The setting of $\tau_t$ here is larger than that of DMOPs with changing position/shape of PSs/PFs, since most algorithms may require more generations to perform well on problems with more than 4 objectives [62];
- When a change occurs, we may need a solution to be adopted as quickly as possible. So, one may wish to adopt the transferred solutions right after the change happens. However, while this solution is being adopted in practice, it is possible to continue running the optimization for a number of generations (i.e. $\tau_t$s), so that the population can converge before the next change happens, enabling good knowledge transfer for the next change. Therefore, we could output the optimized solutions before the end of running the optimization to see which algorithm is able to quickly output optimized solutions with better quality. Following this line, we set different generations of outputting optimized solution (denoted as *gap*) for different $\tau_t$s: 5, 25 and 50 for $\tau_t = 50$; 5, 25, 50 and 100 for $\tau_t = 100$; 5, 25, 50, 100 and 200 for $\tau_t = 200$;
- All algorithms run 31 times independently, also the same as in DTAEA's work [3];
- 1000 generations are given to each algorithm before the first change so that the population before the change can converge;
- The crossover probability is $p_c = 1.0$ and its distribution index is $\eta_c = 20$. The mutation probability was $p_m = 1/\text{n}$ (where n denotes the number of decision variables) and its distribution $\eta_m = 20$. These parameters are chosen because of their good performance on solving continuous problems, which have been analyzed in [103] and [60].
- The neighbourhood size and the number $n_r$ of solutions allowed to replace in MOEA/D were set to 20 and 2, respectively, which is the same as in the original paper [57].

### 6.3.4 Performance Metrics

- Mean Hypervolume (MHV) [3] is the mean Hypervolume (HV) [104] of obtained solution sets by an algorithm at a set of discrete time steps. HV comprehensively measures the convergence and diversity of solution sets; the larger the better.
- Mean Generational Distance (MGD) [38] [15] is the mean Generational Distance (GD) [105] of obtained solution sets by an algorithm at a set of discrete time steps. GD evaluates the convergence of obtained solution sets; the smaller the better.

Note that these two metrics are used to measure the solution quality of a solution set found for a given environmental condition. They can be also used to measure the overall performance of an algorithm by averaging the metric values of all obtained solutions under multiple environmental changes.

## 6.4 Experimental Results and Analyses

In order to achieve the objectives of the experiment, i.e. answering the research questions and verifying whether the existing DMOEAs for solving DMOPs with fixed NObj are able to tackle DMOPs with a changing NObj, experimental results of all compared algorithms are presented in this section. Furthermore, further analyses regarding performance comparison of different NObj changing sequence and the impact of algorithm parameters are also given in Section 6.4.3 and 6.4.4 of this section.

The metrics MHV and MGD are used to measure the quality of the found solutions at the first generation after the change and in the last generation before the next change by five algorithms. To check if there is significant difference between the proposed LEC and other algorithms across all problem instances in general, Friedman and Nemenyi statistical tests [90] with a significance level 0.05 are adopted across all benchmark problems regarding the MHV and MGD of five compared algorithms. The larger the values of MHV, the better the algorithm. Therefore, the larger the Friedman ranking, the better the algorithm. Similarly, the smaller the Friedman ranking of MGD, the better the algorithm.

Additionally, the Wilcoxon Rank Sum test at the 5% significance level is utilized to check if there is significant difference between the proposed LEC and other algorithms on each individual problem of each frequency of change.

For Friedman and Nemenyi statistical tests, the mean metric (MHV and MGD)

value of 10 environmental changes that each algorithm gets on one problem at each independent run is regarded as an observation of the test, for all frequencies of change. Therefore, there are 403 (13 problems, 31 independent runs) observations for each algorithm in the Friedman and Nemenyi tests. ForWilcoxon Rank Sum test, it is implemented on each benchmark problem regarding each metric of five compared algorithms at each parameter setting. Therefore, there are 31 observations obtained from 31 independent runs for each algorithm on each problem with one parameter setting.

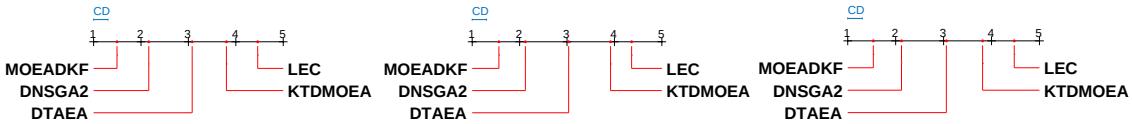
### 6.4.1 Initial Effectiveness of Learnt PS Expansion and Contraction

In order to verify (1) whether the proposed LEC is able to increase solution quality regarding convergence and diversity right after the change compared to KTDMOEA and DTAEA, and (2) whether LEC obtains better solution quality over other baselines after the change, the quality of the solutions obtained by all algorithms in the first generation after changes is compared.

#### 6.4.1.1 Overall Statistic and Specific Metric Results

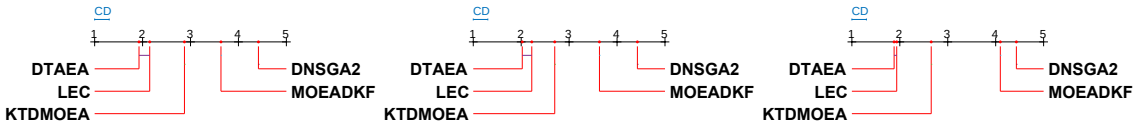
Figures 6.6 and 6.7 present the Nemenyi post-tests results among MHV and MGD of obtained solutions at the first generation after changes by 5 algorithms, when $\tau_t$ = 50, 100 and 200, respectively. Table 6.1 shows the number of cases when LEC obtains significant better quality of solutions than (+), worse than (-) and equal to (=) corresponding algorithms at the first generation measured by metrics MHV and MGD averaged across all changes ('overall'), the changes of increasing NObj from 2 to 7 ('InNObj') and decreasing NObj from 7 to 2 ('DeNObj'), respectively.

Overall, it can be seen from Figure 6.6 that when comparing all algorithms, LEC significantly outperforms all others at all frequencies of changes. In addition, it can be observed from Figure 6.7 that LEC is the equal best, together with DTAEA, regarding GD at all frequencies of changes. This implies that the proposed LEC indeed obtains the best quality of transferred solutions right after changes among all compared algorithms, under all frequencies of changes, This observation is also confirmed by the 'Overall' row of Table 6.1.




(a) $\tau_t = 50$ (b) $\tau_t = 100$ (c) $\tau_t = 200$

Figure 6.6: Friedman ranking among MHV of obtained solutions at the first generation by 5 algorithms when $\tau_t$ = 50, 100 and 200, respectively.




(a) $\tau_t = 50$ (b) $\tau_t = 100$ (c) $\tau_t = 200$

Figure 6.7: Friedman ranking among GD of obtained solutions at the first generation by 5 algorithms when $\tau_t = 50$, 100 and 200, respectively.

Table 6.1: The number of cases when LEC obtains significant better quality of solutions than (+), worse than (-) and equal to (=) corresponding algorithms at the first generation measured by metrics MHV and MGD for all changes ('overall'), the changes of increasing NObj from 2 to 7 ('InNObj') and decreasing NObj from 7 to 2 ('DeNObj'), respectively.

| +/-/= | Metrics | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|
| Overall | MHV | 34/2/3 | 36/3/0 | 36/2/1 | 31/8/0 |
| | MGD | 39/0/0 | 32/7/0 | 18/21/0 | 24/8/7 |
| InNObj | MHV | 35/4/0 | 36/3/0 | 35/1/3 | 5/0/34 |
| | MGD | 36/0/3 | 27/12/0 | 11/21/7 | 3/3/32 |
| DeNObj | MHV | 34/3/2 | 36/2/1 | 28/8/3 | 31/8/0 |
| | MGD | 37/2/0 | 31/8/0 | 26/13/0 | 27/9/3 |

It can be seen from the 'InNObj' row of Table 6.1 that LEC gets significantly better MHV than and equal MGD to KTDMOEA when increasing NObj from 2 to 7. This implies that LEC indeed increases the diversity of KTDMOEA when increasing NObj right after changes. In addition, it can be found from the 'DeNObj' row of Table 6.1 that LEC gets better MGD than all other algorithms when decreasing NObj from 7 to 2, which implies that LEC indeed increases the convergence when decreasing NObj right after changes. The comparison results of LEC and KTDMOEA regarding MGD on cases of 'InNObj' and 'DeNObj' explain the results that LEC gets overall better GD than KTDMOEA from Figure 6.7.

Results of mean and standard deviation values for MHV and MGD of obtained solutions by all compared algorithms at the first generation right after changes across 10 environmental changes when the NObj increase from 2 to 7 and then decrease from 7 to 2 are presented in Tables 6.2 and 6.3, respectively. It is clear from Table 6.2 that LEC performs the best on all test problems except for WFG9. Regarding WFG9, it can be observed from those two tables that LEC gets worse MHV value due to worse MGD values, which can be observed from Table 6.3. The results of the comparison of LEC and KTDMOEA on F4 demonstrate that our proposed LEC strategy indeed improves KTDMOEA when solving problems with extremely strong bias like F4.

Table 6.2: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj

| Prob. | $\tau_t$ | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 50 | 4.95E-01(2.54E-07)† | 1.11E-02(2.60E-04)† | 7.27E-01(3.68E-03)† | 7.51E-01(5.90E-03)† | 8.64E-01(1.25E-03) |
| | 100 | 4.27E-01(4.29E-05)† | 1.96E-02(6.42E-04)† | 7.75E-01(2.94E-03)† | 8.01E-01(1.75E-03)† | 8.67E-01(6.92E-04) |
| | 200 | 2.97E-01(2.55E-06)† | 7.29E-03(1.75E-04)† | 8.12E-01(2.45E-03)† | 8.11E-01(1.69E-03)† | 8.76E-01(2.19E-04) |
| F2 | 50 | 8.68E-01(2.10E-07)† | 6.11E-01(1.12E-03)† | 9.29E-01(1.84E-06)† | 9.54E-01(4.49E-06)‡ | 9.51E-01(1.56E-06) |
| | 100 | 8.72E-01(2.51E-06)† | 7.09E-01(5.76E-04)† | 9.29E-01(2.10E-06)† | 9.54E-01(1.86E-06)‡ | 9.52E-01(1.33E-06) |
| | 200 | 8.74E-01(2.88E-06)† | 7.34E-01(1.20E-05)† | 9.29E-01(1.11E-06)† | 9.55E-01(3.33E-06)‡ | 9.52E-01(8.85E-07) |
| F3 | 50 | 4.57E-01(2.86E-04)† | 0.00E+00(0.00E+00)† | 4.93E-01(1.26E-03)† | 5.72E-01(8.81E-04)† | 6.32E-01(2.95E-03) |
| | 100 | 3.74E-01(1.97E-04)† | 0.00E+00(0.00E+00)† | 5.03E-01(1.37E-03)† | 5.93E-01(1.89E-03)† | 6.61E-01(2.04E-03) |
| | 200 | 2.77E-01(1.28E-04)† | 0.00E+00(0.00E+00)† | 5.37E-01(1.23E-03)† | 6.14E-01(1.45E-03)† | 6.18E-01(1.64E-03) |
| F4 | 50 | 8.21E-01(5.69E-06)† | 9.02E-01(3.64E-05)† | 9.28E-01(5.34E-06)† | 8.68E-01(8.44E-04)† | 9.34E-01(2.25E-04) |
| | 100 | 8.14E-01(1.36E-05)† | 9.09E-01(3.69E-05)† | 9.28E-01(4.54E-06)† | 8.89E-01(5.96E-04)† | 9.31E-01(7.10E-04) |
| | 200 | 8.13E-01(1.79E-05)† | 9.11E-01(1.55E-05)† | 9.29E-01(3.65E-06) | 8.63E-01(1.59E-03)† | 9.31E-01(1.47E-04) |
| WFG1 | 50 | 9.17E-01(2.17E-06)† | 3.05E-01(9.83E-06)† | 9.10E-01(3.60E-06)† | 9.33E-01(5.59E-06)† | 9.38E-01(1.53E-05) |
| | 100 | 9.17E-01(6.78E-06)† | 3.08E-01(6.09E-06)† | 9.18E-01(8.92E-07)† | 9.36E-01(3.15E-06)† | 9.38E-01(3.19E-06) |
| | 200 | 9.14E-01(7.93E-07)† | 3.09E-01(1.71E-06)† | 9.20E-01(5.90E-07)† | 9.36E-01(2.65E-06)† | 9.37E-01(1.21E-06) |
| WFG2 | 50 | 8.97E-01(1.89E-06)† | 2.36E-01(5.57E-07)† | 9.25E-01(1.90E-05)† | 9.44E-01(1.10E-05)† | 9.48E-01(6.00E-06) |
| | 100 | 8.99E-01(1.86E-06)† | 2.36E-01(4.40E-07)† | 9.00E-01(3.08E-04)† | 9.41E-01(1.74E-05)† | 9.44E-01(7.23E-06) |
| | 200 | 8.81E-01(1.17E-05)† | 2.35E-01(1.13E-07)† | 9.25E-01(1.61E-05)† | 9.42E-01(9.36E-06)† | 9.44E-01(5.66E-06) |
| WFG3 | 50 | 6.57E-01(3.46E-07)† | 2.51E-01(6.50E-07)† | 6.39E-01(3.00E-05)† | 6.52E-01(4.70E-05)† | 6.63E-01(5.31E-05) |
| | 100 | 6.51E-01(1.88E-06) | 2.51E-01(3.40E-07)† | 6.26E-01(9.41E-05)† | 6.50E-01(3.29E-05)† | 6.51E-01(5.93E-05) |
| | 200 | 6.51E-01(4.60E-07) | 2.51E-01(3.50E-07)† | 5.90E-01(9.05E-05)† | 6.36E-01(8.59E-05)† | 6.50E-01(3.26E-05) |
| WFG4 | 50 | 6.41E-01(6.58E-07)† | 2.97E-01(1.26E-05)† | 7.51E-01(7.51E-07)† | 8.12E-01(1.03E-06)† | 8.15E-01(1.25E-06) |
| | 100 | 6.44E-01(3.78E-07)† | 2.97E-01(5.95E-06)† | 7.52E-01(3.00E-07)† | 8.13E-01(7.52E-07)† | 8.16E-01(1.21E-06) |
| | 200 | 6.52E-01(1.15E-06)† | 2.98E-01(3.26E-06)† | 7.51E-01(7.62E-07)† | 8.13E-01(4.34E-07)† | 8.16E-01(8.64E-07) |
| WFG5 | 50 | 6.19E-01(4.99E-06)† | 5.13E-01(6.30E-06)† | 7.14E-01(1.61E-06)† | 7.71E-01(9.11E-06)‡ | 7.68E-01(4.28E-06) |
| | 100 | 6.20E-01(7.43E-06)† | 5.12E-01(8.52E-06)† | 7.17E-01(8.88E-07)† | 7.75E-01(4.45E-06)‡ | 7.74E-01(1.99E-06) |
| | 200 | 6.30E-01(1.02E-06)† | 5.12E-01(6.84E-06)† | 7.17E-01(1.20E-06)† | 7.78E-01(2.77E-06)‡ | 7.75E-01(2.62E-06) |
| WFG6 | 50 | 5.97E-01(3.16E-06)† | 1.41E-01(6.45E-06)† | 6.84E-01(1.16E-04)† | 7.26E-01(3.65E-04)‡ | 7.22E-01(2.34E-04) |
| | 100 | 6.02E-01(6.08E-06)† | 1.41E-01(5.23E-06)† | 6.40E-01(1.18E-03)† | 7.39E-01(2.95E-04)‡ | 6.80E-01(1.01E-03) |
| | 200 | 6.20E-01(2.80E-05)† | 1.43E-01(4.64E-06)† | 6.81E-01(4.30E-05)† | 7.41E-01(1.56E-04)† | 7.44E-01(9.12E-05) |
| WFG7 | 50 | 6.25E-01(1.61E-06)† | 6.29E-01(3.42E-05)† | 7.31E-01(1.65E-06)† | 8.03E-01(1.62E-06)† | 8.05E-01(1.64E-06) |
| | 100 | 6.27E-01(1.89E-06)† | 6.49E-01(3.79E-05)† | 7.31E-01(2.23E-06)† | 8.03E-01(1.18E-06)† | 8.07E-01(2.06E-06) |
| | 200 | 6.30E-01(5.30E-06)† | 6.51E-01(2.66E-05)† | 7.32E-01(1.61E-06)† | 8.04E-01(1.42E-06)† | 8.07E-01(1.18E-06) |
| WFG8 | 50 | 6.28E-01(1.28E-06)† | 6.29E-01(1.96E-05)† | 7.30E-01(1.74E-06)† | 8.03E-01(1.35E-06)† | 8.05E-01(2.52E-06) |
| | 100 | 6.27E-01(2.87E-07)† | 6.50E-01(2.95E-05)† | 7.31E-01(1.73E-06)† | 8.03E-01(1.52E-06)† | 8.07E-01(1.90E-06) |
| | 200 | 6.28E-01(2.48E-06)† | 6.50E-01(2.20E-05)† | 7.31E-01(1.91E-06)† | 8.04E-01(1.23E-06)† | 8.07E-01(1.08E-06) |
| WFG9 | 50 | 4.91E-01(2.30E-08)‡ | 5.87E-01(1.36E-02)‡ | 5.41E-01(1.66E-03)‡ | 4.47E-01(4.39E-04)† | 4.54E-01(2.03E-04) |
| | 100 | 5.03E-01(5.02E-05)‡ | 6.16E-01(1.03E-02)‡ | 5.12E-01(1.81E-03)‡ | 4.51E-01(2.55E-04)† | 4.58E-01(2.23E-04) |
| | 200 | 5.41E-01(1.26E-04) | 5.94E-01(1.06E-02)‡ | 4.96E-01(1.32E-03)† | 4.60E-01(2.21E-04)† | 5.30E-01(2.03E-03) |
| +-= | | 34/2/3 | 36/3/0 | 36/2/1 | 31/8/0 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

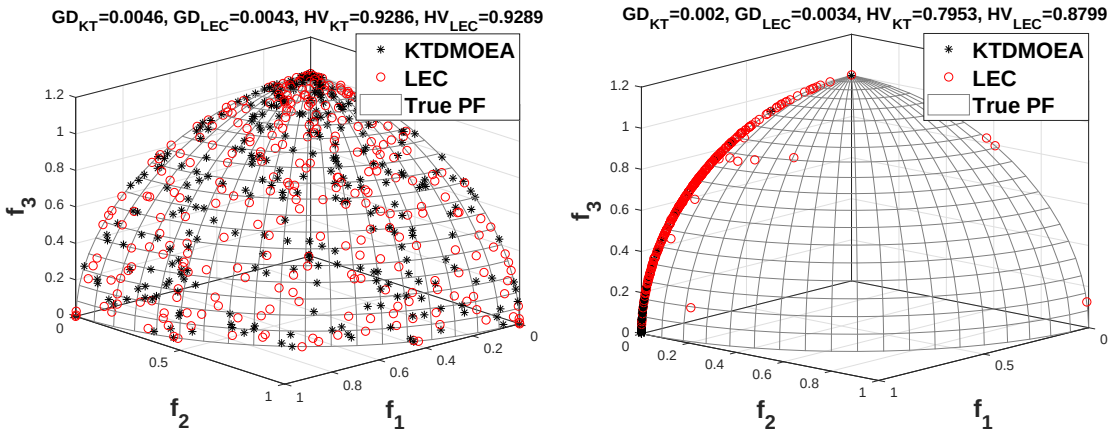
**6.4.1.1.1 When Increasing NObj** Mean and standard deviation values of HV and GD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of increasing NObj from 2 to 7 are presented in Tables 6.4 and 6.5.

Figure 6.8 presents transferred solutions by KTDMOEA and LEC on two problem F2 and F4 right after increasing the NObj from 2 to 3. The MGD and MHV values of both approaches are also put in the title of each subfigure. It is clear that KTDMOEA and LEC gets equally good solutions on F2 while LEC gets better solutions than KTDMOEA on F4. This result demonstrates that our proposed strategy of learning to expand PS indeed overcomes the limitation of KTDMOEA in solving problems with strong bias in the fitness landscape.

Note that transferred solutions by LEC are still not so well spread throughout the true PF of F4. The reason is that even though LEC is able to determine the most promising expansion directions, the strong bias on F4 hinders most transferred solution generated along the directions spread on the true PF, since only a small portion of $x$ value within the range of [0,1] would make $y = x^{100}$ in F4 cover most value of [0,1] with all other value of $x$ make $y$ equal to 0.

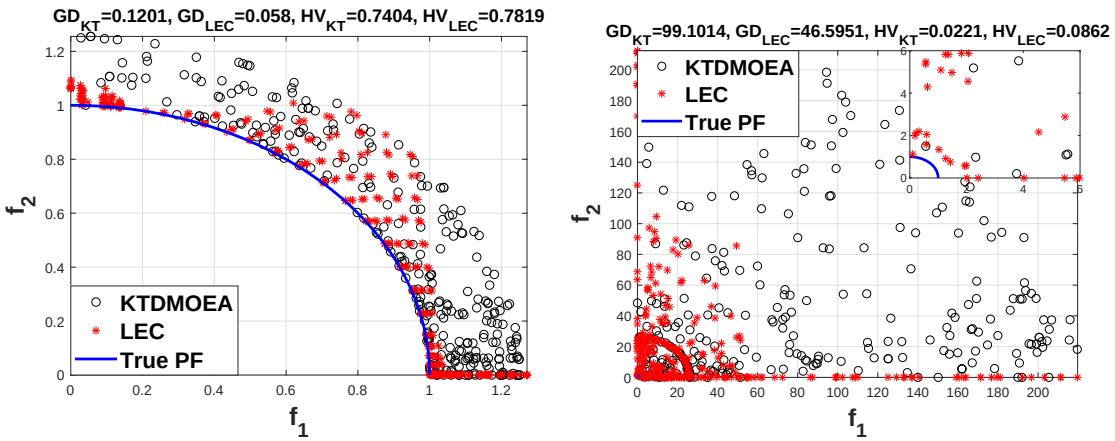
For problems with either strong bias or no bias in the fitness landscape, as long as solutions are converged before the change, LEC is able to learn the correct expansion directions. As described in Section 6.1, for the problem with strong bias like F4, KTDMOEA is unable to find the expansion directions. However, in our proposed LEC when learning the expansion direction, the direction is learnt from the candidate directions that are perpendicular to the previous PS. Therefore, as long as solutions are converged before the change, as shown in Figure 6.5(a), the eigenvectors can be obtained via PCA for learning the candidate expansion directions. After learning these candidate directions, the most promising expansion directions can be naturally selected according to which direction helps diversity.

**6.4.1.1.2 When Decreasing NObj** Similarly, Mean and standard deviation values of MHV and MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of decreasing NObj from 7 to 2 are presented in Tables 6.6 and 6.7.




(a) F2 (b) F4

Figure 6.8: Transferred solutions by KTDMOEA and LEC on two problem F2 and F4 directly after increasing the NObj from 2 to 3. The MGD and MHV values of both approaches are also put in the title of each subfigure.




(a) F2 (b) F3

Figure 6.9: Transferred solutions by KTDMOEA and LEC on two problem F2 and F3 directly after decreasing the NObj from 3 to 2. The MGD and MHV values of both approaches are also put in the title of each subfigure.

Table 6.3: Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj.

| Prob. | $\tau_t$ | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 50 | 1.23E+02(3.79E+01)† | 1.19E+03(8.20E+02)† | 1.83E+01(2.15E+00)† | 2.15E+01(6.32E+00)† | 9.65E+00(6.40E-01) |
| | 100 | 1.36E+02(1.95E+00)† | 1.23E+03(4.53E+02)† | 1.81E+01(1.54E+00)† | 1.72E+01(3.53E+00)† | 8.71E+00(6.85E-01) |
| | 200 | 1.55E+02(4.03E-01)† | 1.25E+03(3.28E+02)† | 1.80E+01(1.55E+00)† | 1.49E+01(2.83E+00)† | 8.49E+00(7.48E-01) |
| F2 | 50 | 4.00E-01(1.43E-05)† | 7.85E+02(1.47E+02)† | 7.89E-02(5.27E-06)‡ | 7.21E-02(4.70E-06)‡ | 8.49E-02(1.86E-05) |
| | 100 | 4.04E-01(2.58E-05)† | 7.74E+02(6.21E+02)† | 7.53E-02(4.01E-06)‡ | 7.18E-02(5.79E-06)‡ | 8.61E-02(1.69E-05) |
| | 200 | 4.13E-01(1.19E-05)† | 7.82E+02(2.36E+02)† | 7.36E-02(2.99E-06)‡ | 7.17E-02(4.41E-06)‡ | 8.91E-02(7.89E-06) |
| F3 | 50 | 2.65E+02(1.54E+02)† | 1.29E+03(9.72E+02)† | 4.56E+01(8.33E+00)† | 5.82E+01(5.95E+01)† | 2.37E+01(6.42E+00) |
| | 100 | 3.14E+02(1.21E+02)† | 1.31E+03(2.58E+02)† | 4.35E+01(6.80E+00)† | 4.28E+01(1.71E+01)† | 2.18E+01(6.36E+00) |
| | 200 | 3.67E+02(1.20E+01)† | 1.30E+03(7.95E+02)† | 4.27E+01(7.90E+00)† | 3.96E+01(1.45E+01)† | 2.05E+01(1.83E+00) |
| F4 | 50 | 5.19E-01(4.13E-06)† | 2.63E-01(3.36E-03)† | 1.07E-01(4.59E-05)† | 7.77E-02(1.52E-04)‡ | 9.29E-02(3.87E-05) |
| | 100 | 5.44E-01(2.98E-06)† | 2.17E-01(1.61E-03)† | 1.01E-01(2.05E-05)† | 7.62E-02(1.43E-04)‡ | 8.90E-02(3.42E-05) |
| | 200 | 5.50E-01(7.71E-06)† | 2.14E-01(2.52E-03)† | 9.89E-02(2.27E-05)† | 9.09E-02(2.05E-04)‡ | 9.50E-02(2.79E-05) |
| WFG1 | 50 | 5.66E-01(4.37E-06)† | 9.59E+02(1.07E+03)† | 2.62E-01(1.53E-05)† | 2.45E-01(1.95E-05)† | 2.41E-01(2.03E-05) |
| | 100 | 5.71E-01(6.90E-06)† | 9.76E+02(7.97E+02)† | 2.40E-01(6.55E-06)† | 2.32E-01(1.50E-05)† | 2.31E-01(1.19E-05) |
| | 200 | 5.65E-01(4.88E-06)† | 1.00E+03(2.76E+02)† | 2.31E-01(9.14E-06)† | 2.27E-01(1.47E-05)† | 2.25E-01(8.28E-06) |
| WFG2 | 50 | 1.00E+00(2.18E-04)† | 1.40E+00(7.04E-05)† | 5.20E-01(9.73E-05)‡ | 6.74E-01(2.82E-04)† | 6.62E-01(4.82E-04) |
| | 100 | 1.04E+00(9.64E-06)† | 1.42E+00(4.75E-05)† | 5.39E-01(4.79E-04)‡ | 6.67E-01(2.24E-04) | 6.70E-01(8.39E-04) |
| | 200 | 1.05E+00(1.85E-05)† | 1.43E+00(6.78E-05)† | 4.99E-01(8.61E-05)‡ | 6.50E-01(2.72E-04)† | 6.09E-01(4.02E-04) |
| WFG3 | 50 | 1.61E+00(2.06E-05)† | 1.54E+00(5.39E-05)† | 1.59E+00(1.45E-03)† | 1.57E+00(1.18E-02)† | 1.44E+00(4.92E-03) |
| | 100 | 1.67E+00(5.03E-04)† | 1.54E+00(5.10E-05)† | 1.64E+00(3.14E-03)† | 1.61E+00(5.67E-03)† | 1.53E+00(4.97E-03) |
| | 200 | 1.67E+00(2.21E-04)† | 1.54E+00(6.16E-05)† | 1.61E+00(1.02E-03)† | 1.41E+00(3.67E-03)‡ | 1.54E+00(7.26E-03) |
| WFG4 | 50 | 5.73E-01(3.26E-06)† | 1.02E+00(7.07E-05)† | 2.48E-01(1.95E-05)‡ | 2.73E-01(1.48E-05)† | 2.72E-01(2.07E-05) |
| | 100 | 5.67E-01(1.01E-05)† | 9.92E-01(3.91E-05)† | 2.38E-01(9.58E-06)‡ | 2.72E-01(1.76E-05)† | 2.69E-01(3.37E-05) |
| | 200 | 5.48E-01(1.80E-06)† | 9.82E-01(1.71E-05)† | 2.26E-01(1.26E-05)‡ | 2.70E-01(1.44E-05)† | 2.68E-01(1.95E-05) |
| WFG5 | 50 | 6.43E-01(5.25E-06)† | 2.93E-01(4.68E-05)† | 2.86E-01(8.23E-06)‡ | 3.19E-01(3.84E-05) | 3.19E-01(2.54E-05) |
| | 100 | 6.28E-01(1.30E-06)† | 3.02E-01(3.76E-05)† | 2.81E-01(4.26E-06)‡ | 3.12E-01(2.33E-05) | 3.13E-01(1.92E-05) |
| | 200 | 6.18E-01(1.17E-05)† | 3.06E-01(1.43E-05)† | 2.78E-01(8.38E-06)‡ | 3.14E-01(1.72E-05) | 3.17E-01(2.09E-05) |
| WFG6 | 50 | 7.26E-01(3.82E-06)† | 9.69E-01(8.19E-04)† | 3.75E-01(4.33E-04)‡ | 4.48E-01(8.77E-04) | 4.51E-01(4.49E-04) |
| | 100 | 7.22E-01(1.71E-05)† | 9.69E-01(6.30E-04)† | 5.24E-01(1.16E-02)‡ | 4.26E-01(6.46E-04)‡ | 5.50E-01(5.15E-03) |
| | 200 | 7.14E-01(4.33E-06)† | 9.53E-01(5.37E-04)† | 3.75E-01(1.73E-04)‡ | 4.16E-01(3.89E-04)† | 4.02E-01(1.28E-04) |
| WFG7 | 50 | 7.24E-01(5.80E-06)† | 2.50E-01(3.09E-05)‡ | 2.77E-01(8.65E-06)‡ | 2.89E-01(2.24E-05)† | 2.82E-01(1.53E-05) |
| | 100 | 7.20E-01(1.79E-06)† | 2.57E-01(3.18E-05)‡ | 2.68E-01(9.11E-06)‡ | 2.75E-01(1.82E-05)† | 2.71E-01(1.58E-05) |
| | 200 | 7.26E-01(1.26E-05)† | 2.88E-01(4.89E-05)† | 2.57E-01(3.67E-06)† | 2.59E-01(1.60E-05)† | 2.57E-01(1.49E-05) |
| WFG8 | 50 | 7.22E-01(1.82E-06)† | 2.48E-01(4.27E-05)‡ | 2.77E-01(6.88E-06)‡ | 2.89E-01(2.24E-05)† | 2.85E-01(1.68E-05) |
| | 100 | 7.30E-01(9.88E-06)† | 2.59E-01(4.52E-05)‡ | 2.67E-01(4.72E-06)‡ | 2.76E-01(2.44E-05)† | 2.71E-01(1.88E-05) |
| | 200 | 7.14E-01(5.59E-06)† | 2.89E-01(4.00E-05)† | 2.58E-01(4.34E-06)† | 2.60E-01(1.32E-05)† | 2.57E-01(1.58E-05) |
| WFG9 | 50 | 9.67E-01(3.33E-07)† | 5.85E-01(5.95E-02)‡ | 7.64E-01(5.92E-03)‡ | 9.00E-01(1.00E-03) | 9.10E-01(5.37E-04) |
| | 100 | 9.59E-01(1.28E-04)† | 5.30E-01(5.01E-02)‡ | 8.36E-01(8.39E-03)‡ | 9.02E-01(7.33E-04) | 9.05E-01(6.80E-04) |
| | 200 | 8.95E-01(2.97E-04)† | 5.84E-01(4.94E-02)‡ | 8.35E-01(3.92E-03)† | 8.99E-01(4.58E-04)† | 7.81E-01(6.16E-03) |
| +-= | | 39/0/0 | 32/7/0 | 18/21/0 | 24/8/7 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.4: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 50 | 9.91E-01(1.02E-06)† | 2.21E-02(1.04E-03)† | 9.93E-01(5.51E-09)† | 9.99E-01(6.53E-10) | 9.99E-01(6.09E-10) |
| | 100 | 8.53E-01(1.71E-04)† | 3.93E-02(2.57E-03)† | 9.93E-01(3.17E-08)† | 9.99E-01(4.42E-09) | 9.99E-01(1.13E-09) |
| | 200 | 5.93E-01(6.62E-09)† | 1.46E-02(6.98E-04)† | 9.93E-01(2.75E-09)† | 9.99E-01(1.91E-07) | 9.99E-01(6.57E-10) |
| F2 | 50 | 9.21E-01(1.50E-06)† | 7.59E-01(3.89E-04)† | 9.26E-01(1.34E-08)† | 9.75E-01(8.30E-07) | 9.75E-01(1.95E-06) |
| | 100 | 9.17E-01(1.99E-06)† | 7.62E-01(5.16E-05)† | 9.26E-01(1.34E-08)† | 9.75E-01(7.45E-07) | 9.75E-01(5.86E-07) |
| | 200 | 9.17E-01(3.05E-07)† | 7.57E-01(1.06E-05)† | 9.26E-01(4.34E-09)† | 9.75E-01(6.55E-07) | 9.75E-01(7.53E-07) |
| F3 | 50 | 8.97E-01(3.55E-05)† | 0.00E+00(0.00E+00)† | 9.25E-01(1.52E-06)† | 9.76E-01(4.03E-07) | 9.77E-01(3.15E-07) |
| | 100 | 7.49E-01(7.87E-04)† | 0.00E+00(0.00E+00)† | 9.24E-01(2.79E-06)† | 9.76E-01(5.67E-06) | 9.76E-01(3.25E-05) |
| | 200 | 5.27E-01(7.39E-04)† | 0.00E+00(0.00E+00)† | 9.25E-01(1.60E-06)† | 9.75E-01(4.23E-05) | 9.77E-01(2.07E-07) |
| F4 | 50 | 9.22E-01(7.23E-07)† | 9.15E-01(9.93E-05)† | 9.39E-01(2.84E-09)† | 8.69E-01(1.65E-03)† | 9.56E-01(3.18E-04) |
| | 100 | 9.05E-01(2.63E-06)† | 9.25E-01(1.12E-04)† | 9.39E-01(3.75E-10)† | 8.93E-01(1.01E-03)† | 9.53E-01(6.46E-04) |
| | 200 | 9.00E-01(2.60E-06)† | 9.30E-01(2.45E-05)† | 9.39E-01(2.00E-10)† | 8.73E-01(1.99E-03)† | 9.43E-01(5.83E-04) |
| WFG1 | 50 | 9.55E-01(4.10E-06)† | 3.00E-01(4.14E-06)† | 9.58E-01(3.57E-06)† | 9.84E-01(6.78E-06) | 9.84E-01(3.19E-06) |
| | 100 | 9.53E-01(6.00E-07)† | 3.00E-01(4.06E-06)† | 9.60E-01(1.94E-06)† | 9.84E-01(1.57E-06) | 9.84E-01(3.53E-06) |
| | 200 | 9.56E-01(2.47E-07)† | 3.00E-01(2.42E-06)† | 9.60E-01(3.21E-07)† | 9.84E-01(3.66E-06) | 9.84E-01(1.17E-06) |
| WFG2 | 50 | 9.57E-01(9.84E-08)† | 2.22E-01(1.76E-08)† | 9.59E-01(4.94E-09)† | 9.91E-01(2.06E-08) | 9.91E-01(1.28E-08) |
| | 100 | 9.57E-01(4.78E-07)† | 2.22E-01(2.30E-09)† | 9.24E-01(1.02E-03)† | 9.91E-01(3.02E-08) | 9.91E-01(3.49E-08) |
| | 200 | 9.57E-01(4.14E-07)† | 2.22E-01(2.92E-09)† | 9.59E-01(6.05E-09)† | 9.91E-01(1.98E-08)† | 9.91E-01(1.11E-08) |
| WFG3 | 50 | 7.02E-01(1.29E-06)‡ | 2.35E-01(1.14E-08)† | 7.01E-01(2.86E-06)‡ | 6.94E-01(2.18E-05) | 6.95E-01(1.28E-05) |
| | 100 | 7.04E-01(8.54E-07)‡ | 2.36E-01(5.61E-09)† | 6.72E-01(3.22E-04)† | 6.97E-01(1.82E-05) | 6.96E-01(1.18E-05) |
| | 200 | 7.03E-01(6.60E-07)‡ | 2.36E-01(3.62E-09)† | 7.00E-01(2.99E-06) | 6.97E-01(7.73E-06)† | 6.98E-01(2.09E-05) |
| WFG4 | 50 | 6.46E-01(5.51E-07)† | 2.80E-01(2.09E-05)† | 7.28E-01(7.28E-08)† | 8.49E-01(1.86E-06) | 8.49E-01(1.14E-06) |
| | 100 | 6.55E-01(1.12E-07)† | 2.74E-01(8.15E-06)† | 7.28E-01(3.88E-08)† | 8.50E-01(1.31E-06) | 8.50E-01(1.34E-06) |
| | 200 | 6.58E-01(4.69E-07)† | 2.71E-01(2.76E-06)† | 7.28E-01(3.28E-08)† | 8.50E-01(1.35E-06) | 8.50E-01(1.32E-06) |
| WFG5 | 50 | 6.31E-01(4.73E-06)† | 4.91E-01(1.29E-05)† | 6.80E-01(1.99E-06)† | 7.94E-01(1.11E-05) | 7.93E-01(1.48E-05) |
| | 100 | 6.32E-01(8.72E-06)† | 4.92E-01(1.45E-05)† | 6.83E-01(2.85E-07)† | 7.96E-01(9.48E-06) | 7.97E-01(6.12E-06) |
| | 200 | 6.41E-01(9.37E-07)† | 4.91E-01(1.58E-05)† | 6.84E-01(3.28E-07)† | 7.99E-01(7.54E-06) | 7.98E-01(8.06E-06) |
| WFG6 | 50 | 6.08E-01(3.91E-06)† | 1.26E-01(2.40E-06)† | 6.58E-01(4.44E-04)† | 7.73E-01(8.60E-04) | 7.81E-01(5.32E-04) |
| | 100 | 6.04E-01(3.66E-06)† | 1.26E-01(1.06E-06)† | 5.67E-01(4.50E-03)† | 7.83E-01(8.67E-04) | 7.81E-01(6.51E-04) |
| | 200 | 6.22E-01(9.42E-06)† | 1.27E-01(6.91E-06)† | 6.54E-01(1.67E-04)† | 7.77E-01(2.95E-04) | 7.78E-01(2.99E-04) |
| WFG7 | 50 | 6.49E-01(8.58E-06)† | 6.16E-01(3.38E-05)† | 7.14E-01(1.92E-07)† | 8.53E-01(1.24E-06) | 8.53E-01(1.29E-06) |
| | 100 | 6.47E-01(7.81E-07)† | 6.33E-01(2.70E-05)† | 7.13E-01(4.70E-08)† | 8.53E-01(1.41E-06) | 8.53E-01(1.27E-06) |
| | 200 | 6.49E-01(4.00E-06)† | 6.37E-01(1.42E-05)† | 7.14E-01(7.71E-09)† | 8.54E-01(1.15E-06) | 8.53E-01(8.67E-07) |
| WFG8 | 50 | 6.50E-01(1.70E-06)† | 6.16E-01(3.09E-05)† | 7.14E-01(7.81E-08)† | 8.53E-01(1.29E-06) | 8.53E-01(2.33E-06) |
| | 100 | 6.47E-01(6.69E-07)† | 6.34E-01(3.39E-05)† | 7.13E-01(5.50E-08)† | 8.53E-01(1.14E-06) | 8.53E-01(1.36E-06) |
| | 200 | 6.48E-01(1.67E-06)† | 6.36E-01(1.16E-05)† | 7.14E-01(5.77E-09)† | 8.54E-01(9.05E-07) | 8.54E-01(1.07E-06) |
| WFG9 | 50 | 3.96E-01(3.86E-06)† | 5.63E-01(2.49E-02)‡ | 4.34E-01(2.29E-04) | 4.32E-01(8.00E-04) | 4.36E-01(5.68E-04) |
| | 100 | 4.07E-01(2.18E-04)† | 5.88E-01(2.68E-02)‡ | 4.05E-01(1.24E-03)† | 4.37E-01(8.28E-04) | 4.37E-01(7.05E-04) |
| | 200 | 4.79E-01(3.20E-04)‡ | 5.53E-01(2.60E-02)‡ | 4.38E-01(3.38E-04) | 4.39E-01(6.80E-04) | 4.42E-01(5.90E-04) |
| +-= | | 35/4/0 | 36/3/0 | 35/1/3 | 5/0/34 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.5: Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 50 | 4.62E+01(6.03E-01)† | 1.05E+03(1.47E+03)† | 3.29E-02(6.90E-04)† | 1.71E-02(7.46E-05) | 1.94E-02(2.71E-04) |
| | 100 | 5.73E+01(5.75E+00)† | 1.09E+03(1.26E+03)† | 5.09E-02(2.37E-03)† | 1.01E-01(1.21E-01) | 2.42E-02(1.57E-03) |
| | 200 | 9.38E+01(6.65E-01)† | 1.12E+03(1.57E+03)† | 3.89E-02(7.24E-04)† | 7.64E-02(9.86E-02) | 2.37E-02(1.78E-03) |
| F2 | 50 | 2.42E-01(7.14E-05)† | 9.39E+02(7.42E+02)† | 4.67E-02(5.47E-07)‡ | 5.13E-02(1.26E-06) | 5.11E-02(1.99E-06) |
| | 100 | 2.53E-01(1.93E-05)† | 9.16E+02(2.92E+03)† | 4.43E-02(2.24E-07)‡ | 5.29E-02(1.20E-06) | 5.29E-02(6.37E-07) |
| | 200 | 2.67E-01(1.98E-05)† | 9.32E+02(1.05E+03)† | 4.29E-02(2.80E-07)‡ | 5.28E-02(6.54E-07) | 5.28E-02(1.18E-06) |
| F3 | 50 | 1.29E+02(2.53E+00)† | 9.67E+02(2.28E+03)† | 1.16E-01(1.43E-02)† | 1.44E-01(3.50E-02) | 7.75E-02(5.68E-03) |
| | 100 | 1.46E+02(2.36E+02)† | 9.81E+02(6.32E+02)† | 1.28E-01(8.61E-03)† | 4.37E-01(1.61E+00)† | 1.01E-01(4.39E-02) |
| | 200 | 2.17E+02(2.98E+01)† | 9.71E+02(1.84E+03)† | 9.98E-02(1.25E-02) | 7.30E-01(1.08E+01) | 8.42E-02(9.10E-03) |
| F4 | 50 | 2.84E-01(1.52E-05)† | 3.74E-02(7.02E-05)‡ | 3.52E-02(1.68E-07)‡ | 2.37E-02(7.08E-06)‡ | 4.01E-02(7.48E-06) |
| | 100 | 3.31E-01(1.68E-05)† | 3.36E-02(8.22E-06)‡ | 3.29E-02(2.45E-07)‡ | 2.55E-02(7.64E-06)‡ | 3.94E-02(2.15E-05) |
| | 200 | 3.55E-01(2.62E-05)† | 3.26E-02(1.61E-06)‡ | 3.22E-02(1.59E-07)‡ | 2.44E-02(1.17E-05)‡ | 3.85E-02(1.51E-05) |
| WFG1 | 50 | 5.70E-01(6.63E-06)† | 9.82E+02(3.63E+03)† | 2.59E-01(2.36E-05)† | 2.48E-01(3.89E-05) | 2.50E-01(4.29E-05) |
| | 100 | 5.92E-01(1.24E-05)† | 1.03E+03(1.13E+03)† | 2.50E-01(1.87E-05)† | 2.46E-01(5.42E-05) | 2.48E-01(5.04E-05) |
| | 200 | 5.84E-01(2.07E-06)† | 1.07E+03(1.10E+03)† | 2.42E-01(1.98E-05) | 2.45E-01(5.24E-05) | 2.43E-01(2.01E-05) |
| WFG2 | 50 | 1.01E+00(1.10E-04)† | 1.46E+00(2.12E-05)† | 6.63E-01(1.16E-04)‡ | 8.57E-01(8.87E-04) | 8.56E-01(7.60E-04) |
| | 100 | 1.05E+00(6.74E-06)† | 1.46E+00(2.34E-05)† | 7.10E-01(1.71E-03)‡ | 8.47E-01(7.98E-04) | 8.54E-01(1.05E-03) |
| | 200 | 1.04E+00(3.92E-05)† | 1.47E+00(2.36E-05)† | 6.34E-01(2.25E-05)‡ | 8.23E-01(8.55E-04) | 8.22E-01(9.36E-04) |
| WFG3 | 50 | 1.49E+00(8.49E-05) | 1.65E+00(1.51E-05)† | 1.47E+00(3.79E-03) | 1.48E+00(3.11E-02) | 1.50E+00(1.47E-02) |
| | 100 | 1.57E+00(1.45E-03) | 1.65E+00(1.42E-05)† | 1.56E+00(1.08E-02) | 1.52E+00(1.47E-02) | 1.53E+00(1.78E-02) |
| | 200 | 1.53E+00(9.34E-05) | 1.65E+00(1.13E-05)† | 1.48E+00(1.57E-03) | 1.50E+00(1.41E-02) | 1.53E+00(2.03E-02) |
| WFG4 | 50 | 6.02E-01(2.03E-06)† | 1.16E+00(1.17E-04)† | 2.82E-01(2.06E-05)‡ | 3.45E-01(6.26E-05) | 3.43E-01(7.68E-05) |
| | 100 | 5.89E-01(7.25E-07)† | 1.13E+00(6.00E-05)† | 2.68E-01(1.29E-05)‡ | 3.47E-01(6.95E-05) | 3.42E-01(1.17E-04) |
| | 200 | 5.82E-01(1.08E-06)† | 1.12E+00(3.44E-05)† | 2.49E-01(1.15E-05)‡ | 3.46E-01(5.97E-05) | 3.44E-01(6.57E-05) |
| WFG5 | 50 | 6.75E-01(1.43E-06)† | 3.64E-01(1.71E-04)† | 3.26E-01(2.52E-05)‡ | 3.78E-01(8.01E-05) | 3.78E-01(7.61E-05) |
| | 100 | 6.72E-01(7.82E-06)† | 3.77E-01(1.22E-04)† | 3.20E-01(1.59E-05)‡ | 3.80E-01(8.00E-05) | 3.82E-01(5.63E-05) |
| | 200 | 6.57E-01(5.93E-06)† | 3.81E-01(5.68E-05)† | 3.16E-01(1.54E-05)‡ | 3.90E-01(5.49E-05) | 3.88E-01(5.08E-05) |
| WFG6 | 50 | 7.34E-01(7.05E-06)† | 1.10E+00(4.58E-04)† | 3.79E-01(1.64E-03)‡ | 4.62E-01(1.91E-03) | 4.52E-01(1.01E-03) |
| | 100 | 7.63E-01(2.38E-05)† | 1.10E+00(2.77E-04)† | 6.83E-01(4.66E-02)† | 4.52E-01(1.86E-03) | 4.51E-01(1.44E-03) |
| | 200 | 7.45E-01(1.72E-05)† | 1.09E+00(1.15E-03)† | 3.84E-01(6.85E-04)‡ | 4.55E-01(5.93E-04) | 4.51E-01(5.22E-04) |
| WFG7 | 50 | 7.39E-01(1.69E-05)† | 1.91E-01(3.07E-05)‡ | 2.87E-01(2.38E-05)‡ | 3.20E-01(6.31E-05)† | 3.16E-01(5.42E-05) |
| | 100 | 7.38E-01(7.46E-06)† | 2.00E-01(4.64E-05)‡ | 2.72E-01(1.85E-05)† | 3.01E-01(5.58E-05) | 3.03E-01(4.81E-05) |
| | 200 | 7.55E-01(4.15E-06)† | 2.41E-01(1.04E-04)‡ | 2.54E-01(4.65E-06)‡ | 2.76E-01(4.85E-05) | 2.76E-01(4.17E-05) |
| WFG8 | 50 | 7.40E-01(2.20E-06)† | 1.90E-01(2.83E-05)‡ | 2.88E-01(1.60E-05)‡ | 3.20E-01(6.33E-05) | 3.22E-01(5.58E-05) |
| | 100 | 7.60E-01(2.73E-05)† | 2.01E-01(5.95E-05)‡ | 2.73E-01(1.46E-05)‡ | 3.01E-01(8.40E-05) | 3.03E-01(6.93E-05) |
| | 200 | 7.43E-01(7.01E-07)† | 2.40E-01(8.41E-05)‡ | 2.54E-01(7.81E-06)† | 2.79E-01(6.82E-05)† | 2.77E-01(5.08E-05) |
| WFG9 | 50 | 1.19E+00(9.77E-06)† | 6.67E-01(1.21E-01)‡ | 1.03E+00(1.71E-03) | 1.02E+00(3.41E-03) | 1.04E+00(2.30E-03) |
| | 100 | 1.18E+00(3.38E-04)† | 6.25E-01(1.38E-01)‡ | 1.12E+00(1.28E-02)† | 1.03E+00(2.53E-03) | 1.03E+00(2.69E-03) |
| | 200 | 1.08E+00(1.10E-03)† | 7.16E-01(1.29E-01)‡ | 1.01E+00(3.97E-04) | 1.02E+00(1.69E-03) | 1.02E+00(2.13E-03) |
| +-= | | 36/0/3 | 27/12/0 | 11/21/7 | 3/3/32 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.6: Mean and standard deviation values of MHV metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing number of objectives, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 50 | 0.00E+00(0.00E+00)† | 0.00E+00(0.00E+00)† | 4.61E-01(1.47E-02)† | 5.04E-01(2.36E-02)† | 7.28E-01(4.99E-03) |
| | 100 | 2.83E-06(2.48E-10)† | 0.00E+00(0.00E+00)† | 5.57E-01(1.18E-02)† | 6.03E-01(7.00E-03)† | 7.35E-01(2.76E-03) |
| | 200 | 1.72E-03(1.05E-05)† | 0.00E+00(0.00E+00)† | 6.30E-01(9.80E-03)† | 6.23E-01(6.77E-03)† | 7.54E-01(8.77E-04) |
| F2 | 50 | 8.16E-01(4.42E-06)† | 4.63E-01(3.04E-03)† | 9.31E-01(7.59E-06)‡ | 9.33E-01(1.71E-05)‡ | 9.28E-01(3.22E-06) |
| | 100 | 8.28E-01(4.89E-06)† | 6.56E-01(2.15E-03)† | 9.31E-01(8.39E-06)‡ | 9.33E-01(7.14E-06)‡ | 9.29E-01(3.95E-06) |
| | 200 | 8.32E-01(1.05E-05)† | 7.12E-01(3.34E-05)† | 9.32E-01(4.50E-06)‡ | 9.34E-01(1.24E-05)‡ | 9.29E-01(3.24E-06) |
| F3 | 50 | 1.71E-02(1.06E-03)† | 0.00E+00(0.00E+00)† | 6.20E-02(5.00E-03)† | 1.68E-01(3.56E-03)† | 2.87E-01(1.18E-02) |
| | 100 | 0.00E+00(0.00E+00)† | 0.00E+00(0.00E+00)† | 8.15E-02(5.53E-03)† | 2.10E-01(7.65E-03)† | 3.47E-01(8.32E-03) |
| | 200 | 2.77E-02(4.97E-05)† | 0.00E+00(0.00E+00)† | 1.48E-01(4.90E-03)† | 2.53E-01(5.59E-03)† | 2.60E-01(6.53E-03) |
| F4 | 50 | 7.20E-01(2.67E-05)† | 8.89E-01(1.28E-04)† | 9.17E-01(2.14E-05) | 8.68E-01(2.16E-03)† | 9.12E-01(2.46E-04) |
| | 100 | 7.22E-01(6.11E-05)† | 8.92E-01(6.89E-05)† | 9.17E-01(1.82E-05) | 8.85E-01(1.25E-03)† | 9.08E-01(8.63E-04) |
| | 200 | 7.27E-01(6.04E-05)† | 8.92E-01(4.13E-05)† | 9.19E-01(1.46E-05) | 8.52E-01(2.90E-03)† | 9.18E-01(8.91E-06) |
| WFG1 | 50 | 8.80E-01(1.85E-06)† | 3.11E-01(3.08E-05)† | 8.61E-01(8.50E-06)† | 8.82E-01(1.88E-05)† | 8.92E-01(5.22E-05) |
| | 100 | 8.80E-01(2.25E-05)† | 3.15E-01(1.21E-05)† | 8.76E-01(3.12E-06)† | 8.87E-01(8.65E-06)† | 8.93E-01(9.74E-06) |
| | 200 | 8.73E-01(2.50E-06)† | 3.17E-01(5.89E-06)† | 8.80E-01(1.66E-06)† | 8.88E-01(6.42E-06)† | 8.90E-01(2.48E-06) |
| WFG2 | 50 | 8.38E-01(9.31E-06)† | 2.49E-01(2.22E-06)† | 8.92E-01(7.57E-05)† | 8.98E-01(4.36E-05)† | 9.06E-01(2.43E-05) |
| | 100 | 8.42E-01(1.15E-05)† | 2.49E-01(1.77E-06)† | 8.76E-01(5.31E-04)† | 8.92E-01(6.95E-05)† | 8.98E-01(2.91E-05) |
| | 200 | 8.04E-01(4.38E-05)† | 2.49E-01(4.54E-07)† | 8.90E-01(6.46E-05)† | 8.94E-01(3.74E-05)† | 8.98E-01(2.24E-05) |
| WFG3 | 50 | 6.13E-01(4.42E-07)† | 2.66E-01(2.53E-06)† | 5.77E-01(1.15E-04)† | 6.10E-01(1.71E-04)† | 6.32E-01(1.89E-04) |
| | 100 | 5.97E-01(8.91E-06)† | 2.66E-01(1.30E-06)† | 5.80E-01(1.09E-04)† | 6.03E-01(1.08E-04)† | 6.06E-01(2.26E-04) |
| | 200 | 6.00E-01(1.60E-06) | 2.66E-01(1.38E-06)† | 4.81E-01(3.52E-04)† | 5.76E-01(2.96E-04)† | 6.01E-01(1.22E-04) |
| WFG4 | 50 | 6.36E-01(8.86E-07)† | 3.14E-01(1.66E-05)† | 7.75E-01(2.95E-06)† | 7.75E-01(2.03E-06)† | 7.80E-01(3.20E-06) |
| | 100 | 6.32E-01(2.22E-06)† | 3.20E-01(1.26E-05)† | 7.75E-01(1.21E-06)† | 7.76E-01(1.35E-06)† | 7.81E-01(3.10E-06) |
| | 200 | 6.45E-01(2.21E-06)† | 3.25E-01(8.54E-06)† | 7.74E-01(2.90E-06)† | 7.76E-01(1.41E-06)† | 7.82E-01(1.92E-06) |
| WFG5 | 50 | 6.07E-01(5.56E-06)† | 5.34E-01(7.35E-06)† | 7.49E-01(4.67E-06)‡ | 7.48E-01(1.49E-05)‡ | 7.44E-01(4.12E-06) |
| | 100 | 6.08E-01(1.15E-05)† | 5.32E-01(1.00E-05)† | 7.50E-01(3.31E-06)† | 7.55E-01(1.00E-05)‡ | 7.52E-01(2.56E-06) |
| | 200 | 6.19E-01(1.11E-06)† | 5.33E-01(4.71E-06)† | 7.50E-01(4.43E-06)† | 7.58E-01(4.41E-06)‡ | 7.53E-01(1.71E-06) |
| WFG6 | 50 | 5.87E-01(4.47E-06)† | 1.56E-01(1.70E-05)† | 7.10E-01(2.58E-05)‡ | 6.79E-01(2.39E-04)‡ | 6.62E-01(2.41E-04) |
| | 100 | 6.01E-01(1.51E-05)‡ | 1.57E-01(1.63E-05)† | 7.13E-01(3.03E-05)‡ | 6.96E-01(2.88E-04)‡ | 5.79E-01(3.63E-03) |
| | 200 | 6.17E-01(7.96E-05)† | 1.59E-01(8.18E-06)† | 7.08E-01(2.89E-05)† | 7.06E-01(3.01E-04)† | 7.10E-01(4.01E-05) |
| WFG7 | 50 | 6.00E-01(1.05E-05)† | 6.42E-01(6.78E-05)† | 7.48E-01(6.13E-06)† | 7.53E-01(5.63E-06)† | 7.58E-01(4.86E-06) |
| | 100 | 6.08E-01(3.53E-06)† | 6.65E-01(1.05E-04)† | 7.48E-01(9.11E-06)† | 7.53E-01(3.73E-06)† | 7.60E-01(7.16E-06) |
| | 200 | 6.10E-01(1.84E-05)† | 6.65E-01(6.80E-05)† | 7.49E-01(6.54E-06)† | 7.55E-01(3.04E-06)† | 7.61E-01(3.97E-06) |
| WFG8 | 50 | 6.07E-01(1.33E-06)† | 6.42E-01(5.98E-05)† | 7.47E-01(6.58E-06)† | 7.54E-01(3.89E-06)† | 7.58E-01(6.34E-06) |
| | 100 | 6.06E-01(4.73E-07)† | 6.66E-01(5.86E-05)† | 7.49E-01(7.36E-06)† | 7.52E-01(3.50E-06)† | 7.60E-01(4.83E-06) |
| | 200 | 6.08E-01(5.73E-06)† | 6.65E-01(6.33E-05)† | 7.48E-01(7.53E-06)† | 7.55E-01(3.48E-06)† | 7.61E-01(2.92E-06) |
| WFG9 | 50 | 5.86E-01(4.57E-06)‡ | 6.10E-01(9.65E-03)‡ | 6.49E-01(7.11E-03)‡ | 4.63E-01(6.60E-04)† | 4.72E-01(1.94E-04) |
| | 100 | 5.99E-01(2.29E-06)‡ | 6.44E-01(3.46E-03)‡ | 6.19E-01(6.25E-03)‡ | 4.66E-01(2.95E-04)† | 4.78E-01(1.60E-04) |
| | 200 | 6.03E-01(2.45E-05) | 6.35E-01(4.81E-03) | 5.53E-01(5.09E-03)† | 4.82E-01(6.00E-05)† | 6.17E-01(8.06E-03) |
| +-= | | 34/3/2 | 36/2/1 | 28/8/3 | 31/8/0 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

#### 6.4.1.2 How and Why Does LEC Get Better Solution Quality Directly after Changes?

In this section, examples are presented to testify how LEC gets overall better solutions on most problems than KTDMOEA. Besides, the reasons behind are then elaborated. Since learning the expansion and contraction directions is one of the essential and key steps to expand and contract PSs, specific process of learning the directions will be particularly detailed.

Figure 6.9 presents transferred solutions by KTDMOEA and LEC on two problem F1 and F3 right after decreasing the NObj from 3 to 2. The MGD and MHV values of both approaches are also put in the title of each subfigure. It is clear that LEC gets better solutions than KTDMOEA on both F1 and F3 regarding both convergence and diversity. This result demonstrates that our proposed learning to contract PS indeed overcomes the limitation of KTDMOEA in solving problems with multimodality and variable correlation and improves population convergence.

When decreasing the NObj, the manifold of PS after the change is a subset of that before the change [3]. Therefore, there are certain directions along which solutions move from the manifold of PS after the change to that before the change. In this case, these directions locate in or across the PS manifold before the change. Considering these facts, our proposed method of learning contraction directions using PCA exactly captures these potential contraction directions, which are $ev1$ and $ev2$, as shown in Figure 6.5(b). Then, based on which candidate direction results in solutions with better convergence, the most promising contraction directions can be chosen from the candidate directions. Therefore, the contracted PS in LEC can have better convergence than that of KTDMOEA which does not have any mechanism improving convergence.

#### 6.4.1.3 Why Doesn't LEC sometimes Get Good Solution Quality Directly after Changes?

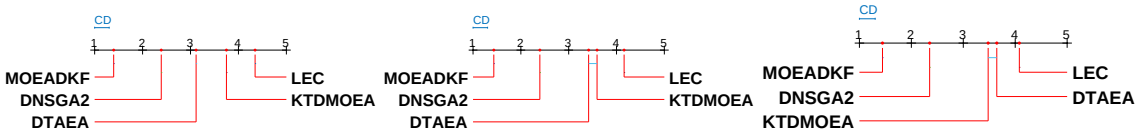
It can be found from Table 6.2 that LEC gets significantly worse MHV results than KTDMOEA on F2, WFG5 and WFG6. By looking at Tables 6.4 and 6.6, we can find that when increasing the NObj from 2 to 7, LEC performs equally to KTDMOEA regarding MHV while LEC gets significantly worse MHV results than KTDMOEA when decreasing the NObj from 7 to 2 on these problems. The worse MHV values of LEC is due to worse MGD values, which can be found in Table 6.7. The reason why LEC gets worse MGD on these problems might be that the learnt PS contraction push solutions already close to PF to be faraway.

It can be found from Tables 6.2-6.7 that LEC performs worse than MOEAD-KF for KTDMOEA on WFG9 regarding MHV and MGD under all frequencies of

Table 6.7: Mean and standard deviation values of MGD metric for solutions obtained by all compared algorithms at the first generation after changes on all DMOPs with a changing NObj, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | $\tau_t$ | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 50 | 1.99E+02(1.51E+02)† | 1.33E+03(7.53E+02)† | 3.65E+01(8.66E+00)† | 4.29E+01(2.52E+01)† | 1.93E+01(2.56E+00) |
| | 100 | 2.15E+02(6.68E-01)† | 1.37E+03(3.48E+02)† | 3.62E+01(6.22E+00)† | 3.42E+01(1.41E+01)† | 1.74E+01(2.77E+00) |
| | 200 | 2.16E+02(3.58E+00)† | 1.38E+03(3.78E+02)† | 3.60E+01(6.20E+00)† | 2.96E+01(1.09E+01)† | 1.70E+01(2.97E+00) |
| F2 | 50 | 5.58E-01(1.55E-06)† | 6.31E+02(3.45E+01)† | 1.11E-01(1.87E-05)‡ | 9.28E-02(1.52E-05)‡ | 1.19E-01(7.84E-05) |
| | 100 | 5.56E-01(7.02E-05)† | 6.32E+02(1.19E+02)† | 1.06E-01(1.58E-05)‡ | 9.07E-02(2.32E-05)‡ | 1.19E-01(6.64E-05) |
| | 200 | 5.60E-01(4.08E-05)† | 6.32E+02(1.07E+01)† | 1.04E-01(1.07E-05)‡ | 9.05E-02(1.68E-05)‡ | 1.25E-01(3.25E-05) |
| F3 | 50 | 4.01E+02(6.21E+02)† | 1.61E+03(5.37E+02)† | 9.11E+01(3.30E+01)† | 1.16E+02(2.38E+02)† | 4.73E+01(2.59E+01) |
| | 100 | 4.83E+02(5.51E+01)† | 1.64E+03(4.64E+02)† | 8.70E+01(2.72E+01)† | 8.51E+01(6.66E+01)† | 4.35E+01(2.46E+01) |
| | 200 | 5.16E+02(6.67E+00)† | 1.63E+03(1.17E+03)† | 8.54E+01(3.16E+01)† | 7.84E+01(4.65E+01)† | 4.09E+01(7.25E+00) |
| F4 | 50 | 7.53E-01(1.12E-06)† | 4.88E-01(1.32E-02)† | 1.80E-01(1.87E-04)† | 1.32E-01(6.08E-04)‡ | 1.46E-01(1.60E-04) |
| | 100 | 7.57E-01(4.74E-07)† | 4.00E-01(6.47E-03)† | 1.70E-01(8.25E-05)† | 1.27E-01(5.57E-04)‡ | 1.39E-01(1.81E-04) |
| | 200 | 7.45E-01(4.30E-06)† | 3.95E-01(1.01E-02)† | 1.65E-01(8.99E-05)† | 1.57E-01(8.08E-04)† | 1.51E-01(1.00E-04) |
| WFG1 | 50 | 5.62E-01(8.55E-06)† | 9.36E+02(1.23E+03)† | 2.65E-01(4.76E-05)† | 2.42E-01(3.69E-05)† | 2.32E-01(3.96E-05) |
| | 100 | 5.50E-01(4.63E-06)† | 9.23E+02(2.17E+03)† | 2.31E-01(1.64E-05)† | 2.18E-01(9.01E-06)† | 2.13E-01(1.42E-05) |
| | 200 | 5.46E-01(1.07E-05)† | 9.31E+02(9.41E+02)† | 2.20E-01(8.28E-06)† | 2.08E-01(2.23E-05)† | 2.06E-01(1.50E-05) |
| WFG2 | 50 | 9.99E-01(3.65E-04)† | 1.35E+00(2.55E-04)† | 3.78E-01(2.53E-04)‡ | 4.91E-01(3.05E-04)† | 4.67E-01(1.25E-03) |
| | 100 | 1.04E+00(2.44E-05)† | 1.37E+00(1.88E-04)† | 3.69E-01(2.15E-04)‡ | 4.87E-01(2.19E-04)† | 4.85E-01(3.02E-03) |
| | 200 | 1.06E+00(1.07E-05)† | 1.39E+00(2.35E-04)† | 3.64E-01(2.75E-04)‡ | 4.76E-01(3.25E-04)† | 3.95E-01(4.37E-04) |
| WFG3 | 50 | 1.74E+00(9.20E-05)† | 1.42E+00(2.04E-04)† | 1.71E+00(1.66E-03)† | 1.65E+00(4.62E-03)† | 1.39E+00(2.42E-03) |
| | 100 | 1.77E+00(1.07E-04)† | 1.43E+00(2.16E-04)‡ | 1.72E+00(1.91E-03)† | 1.69E+00(5.02E-03)† | 1.53E+00(3.71E-03) |
| | 200 | 1.80E+00(9.62E-04)† | 1.44E+00(2.19E-04)‡ | 1.74E+00(2.28E-03)† | 1.32E+00(2.42E-03)‡ | 1.55E+00(3.80E-03) |
| WFG4 | 50 | 5.44E-01(4.78E-06)† | 8.67E-01(7.75E-05)† | 2.15E-01(5.99E-05)† | 2.02E-01(7.08E-06)† | 2.01E-01(8.41E-06) |
| | 100 | 5.45E-01(3.02E-05)† | 8.50E-01(4.63E-05)† | 2.09E-01(3.20E-05)† | 1.97E-01(6.56E-06)† | 1.95E-01(7.43E-06) |
| | 200 | 5.14E-01(7.70E-06)† | 8.47E-01(2.09E-05)† | 2.03E-01(2.96E-05)† | 1.93E-01(5.31E-06)† | 1.92E-01(1.13E-05) |
| WFG5 | 50 | 6.12E-01(1.28E-05)† | 2.22E-01(2.37E-05)‡ | 2.45E-01(8.69E-06)‡ | 2.61E-01(8.28E-05)† | 2.59E-01(2.10E-05) |
| | 100 | 5.84E-01(1.92E-05)† | 2.28E-01(2.95E-05)‡ | 2.42E-01(8.24E-06)‡ | 2.43E-01(1.74E-05) | 2.44E-01(2.87E-05) |
| | 200 | 5.78E-01(2.01E-05)† | 2.31E-01(2.20E-05)‡ | 2.41E-01(1.14E-05)‡ | 2.38E-01(7.00E-06)‡ | 2.46E-01(2.35E-05) |
| WFG6 | 50 | 7.17E-01(1.11E-05)† | 8.43E-01(1.87E-03)† | 3.72E-01(2.14E-05)‡ | 4.33E-01(6.83E-04)‡ | 4.50E-01(5.24E-04) |
| | 100 | 6.82E-01(2.90E-05)† | 8.38E-01(1.63E-03)† | 3.66E-01(2.32E-05)‡ | 4.00E-01(8.38E-04)‡ | 6.48E-01(2.05E-02) |
| | 200 | 6.83E-01(3.10E-06)† | 8.18E-01(6.27E-04)† | 3.66E-01(1.84E-05)† | 3.78E-01(8.09E-04)† | 3.53E-01(1.63E-05) |
| WFG7 | 50 | 7.08E-01(3.35E-06)† | 3.09E-01(6.88E-05)† | 2.66E-01(1.27E-05)† | 2.58E-01(1.44E-05)† | 2.47E-01(4.42E-06) |
| | 100 | 7.03E-01(3.49E-06)† | 3.14E-01(8.29E-05)† | 2.63E-01(1.03E-05)† | 2.49E-01(9.12E-06)† | 2.40E-01(6.72E-06) |
| | 200 | 6.96E-01(3.57E-05)† | 3.36E-01(4.85E-05)† | 2.60E-01(8.42E-06)† | 2.41E-01(8.25E-06)† | 2.37E-01(8.42E-06) |
| WFG8 | 50 | 7.04E-01(2.35E-06)† | 3.06E-01(1.25E-04)† | 2.67E-01(1.16E-05)† | 2.58E-01(1.71E-05)† | 2.48E-01(9.74E-06) |
| | 100 | 6.99E-01(1.57E-06)† | 3.16E-01(8.69E-05)† | 2.62E-01(3.74E-06)† | 2.50E-01(8.26E-06)† | 2.40E-01(9.69E-06) |
| | 200 | 6.86E-01(2.49E-05)† | 3.37E-01(4.20E-05)† | 2.61E-01(6.64E-06)† | 2.41E-01(7.43E-06)† | 2.37E-01(6.40E-06) |
| WFG9 | 50 | 7.41E-01(1.07E-05)‡ | 5.03E-01(4.12E-02)‡ | 5.03E-01(1.99E-02)‡ | 7.79E-01(4.24E-04) | 7.81E-01(2.11E-04) |
| | 100 | 7.34E-01(3.21E-05)‡ | 4.35E-01(1.61E-02)‡ | 5.50E-01(1.73E-02)‡ | 7.77E-01(1.80E-04) | 7.77E-01(3.13E-04) |
| | 200 | 7.09E-01(1.18E-05)† | 4.51E-01(2.04E-02)‡ | 6.60E-01(1.41E-02)† | 7.74E-01(1.79E-04)† | 5.39E-01(2.11E-02) |
| +-= | | 37/2/0 | 31/8/0 | 26/13/0 | 27/9/3 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.




(a) $gap = 5$ (b) $gap = 25$ (c) $gap = 50$

Figure 6.10: Friedman ranking among MHV of solutions after *gap* generations' optimization by 5 algorithms in the case of $\tau_t$=50 and when $gap$ = 5, 25 and 50, respectively.

changes either when increasing or decreasing NObj. The reason is that on WFG9 with very complex problem features in the fitness landscape including multimodality, nonseparability, deceptiveness and bias, it is difficult for the used genetic operators (SBX and PM) in LEC to reach the global optima, while the DE operator in MOEAD-KF is able to do that. Therefore, LEC does not get better results than MOEAD-KF. However, when comparing LEC with both DNSGA2 and DTAEA, we can find that when the changes are more frequent (i.e. $\tau_t$ = 50 and 100), LEC is worse than these two algorithms, while when the changes are less frequent LEC is no worse than them. The reason begind this result might be that the diversity introduction in DNSGA2 and DTAEA is able to help the population qucikly respond to changes while LEC does not have such enough diversity introduction.

### 6.4.2 How Does Learnt PS Expansion and Contraction Help Optimization Process after the Changes?

In order to verify whether the proposed LEC can find solutions with better convergence and diversity after *gap* generations of running the optimization against all other algorithms, the solution quality of all compared algorithms after *gap* generations' optimization for different frequencies of changes is compared.

#### 6.4.2.1 Overall Statistic and Specific Metric Results

Figures 6.10-6.11, 6.12-6.13 and 6.14-6.15 present the Nemenyi post-tests results among MHV and MGD of obtained solutions after *gap* generations' optimization by 5 algorithms, when $\tau_t$ = 50, 100 and 200, respectively. Table 6.8 shows the number of cases when LEC obtains significant better quality of solutions than (+), worse than (-) and equal to (=) corresponding algorithms after optimization of different generations under different frequency of changes (i.e. $\tau_t$=50, 100 and 200) measured by metrics MHV and MGD averaged across all changes ('Overall'), the changes of increasing NObj from 2 to 7 ('InNObj') and the changes of increasing NObj from 2 to 7 'DeNObj', respectively.

Table 6.8: The number of cases when LEC obtains significant better quality of solutions than (+), worse than (-) and equal to (=) corresponding algorithms after optimization of different generations under different frequency of changes (i.e. $\tau_t$=50, 100 and 200) measured by metrics MHV and MGD averaged across all changes ('Overall'), the changes of increasing NObj from 2 to 7 ('InNObj') and the changes of increasing NObj from 2 to 7 'DeNObj', respectively.

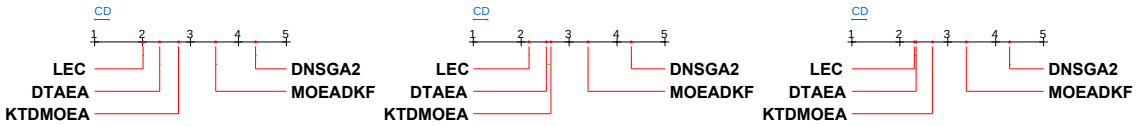
| +/-/= | $\tau_t$ | Metrics | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA |
|---|---|---|---|---|---|---|
| Overall | 50 | MHV | 33/6/0 | 36/3/0 | 30/4/5 | 30/2/7 |
| | | MGD | 36/3/0 | 25/14/0 | 25/11/3 | 21/7/11 |
| | 100 | MHV | 44/8/0 | 48/4/0 | 41/10/1 | 45/4/3 |
| | | MGD | 48/4/0 | 32/20/0 | 31/16/5 | 26/11/15 |
| | 200 | MHV | 55/10/0 | 57/5/3 | 42/5/18 | 42/1/22 |
| | | MGD | 60/0/5 | 42/23/0 | 44/15/6 | 34/10/21 |
| InNObj | 50 | MHV | 34/5/0 | 36/3/0 | 26/7/6 | 3/3/33 |
| | | MGD | 38/1/0 | 23/15/1 | 23/6/10 | 0/3/33 |
| | 100 | MHV | 44/7/1 | 47/5/0 | 35/12/5 | 4/0/48 |
| | | MGD | 52/0/0 | 31/19/2 | 25/13/14 | 1/7/44 |
| | 200 | MHV | 55/5/5 | 56/5/4 | 39/13/13 | 6/0/59 |
| | | MGD | 65/0/0 | 39/23/3 | 30/20/15 | 0/8/57 |
| DeNObj | 50 | MHV | 32/5/1 | 36/3/0 | 23/12/4 | 30/4/5 |
| | | MGD | 36/3/0 | 26/11/2 | 27/11/1 | 22/5/12 |
| | 100 | MHV | 43/8/1 | 48/4/0 | 32/16/4 | 38/6/8 |
| | | MGD | 48/4/0 | 34/18/0 | 41/9/2 | 34/10/8 |
| | 200 | MHV | 55/10/0 | 60/0/5 | 43/2/20 | 43/2/20 |
| | | MGD | 60/3/2 | 42/21/2 | 59/1/5 | 46/13/6 |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.9: The p-values for significant differences in average value detected by Friedman test for MHV and MGD of different *gap* and $\tau_t$.

| $\tau_t$ | *gap* | MHV | MGD |
|---|---|---|---|
| 50 | 5 | 6.3049e-191 | 2.0712e-123 |
| | 25 | 9.1020e-181 | 3.1701e-100 |
| | 50 | 5.7049e-179 | 9.5823e-98 |
| 100 | 5 | 8.4619e-182 | 2.7641e-112 |
| | 25 | 5.4336e-181 | 4.7501e-99 |
| | 50 | 4.3532e-173 | 1.4200e-97 |
| | 50 | 8.6707e-165 | 3.0152e-98 |
| 200 | 5 | 1.2087e-202 | 3.7529e-160 |
| | 25 | 1.4737e-179 | 1.8132e-109 |
| | 50 | 1.8148e-169 | 2.4418e-108 |
| | 100 | 2.9978e-161 | 1.3051e-104 |
| | 200 | 6.1915e-158 | 1.2854e-101 |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

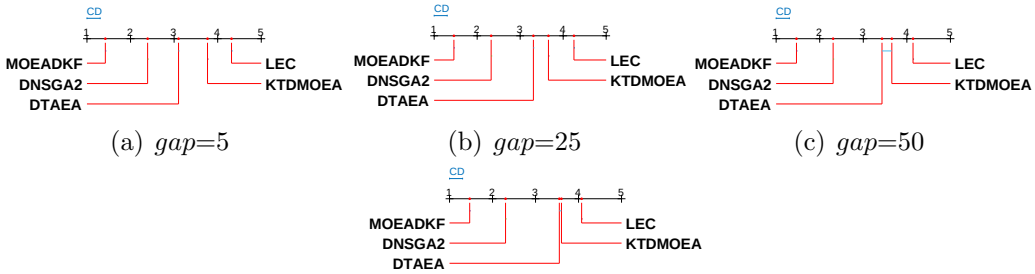



(a) *gap* = 5 (b) *gap* = 25 (c) *gap* = 50

Figure 6.11: Friedman ranking among GD of solutions after *gap* generations' optimization by 5 algorithms in the case of $\tau_t$=50 and when *gap* = 5, 25 and 50, respectively.

Overall, it can be seen from these statistical test results that LEC performs significantly better than or the same as the other approaches at all setting of *gap* under different frequencies of changes, regarding MHV and MGD. Specifically, it is clear from these Figures that LEC gets significantly best results among all compared algorithms regarding MHV and MGD values when *gap* is smaller for all frequencies of changes. In addition, LEC is superior than KTDMOEA regarding GD in almost all cases, which shows that our proposed approach is also able to maintain its superiority over KTDMOEA regarding convergence in the optimization process.

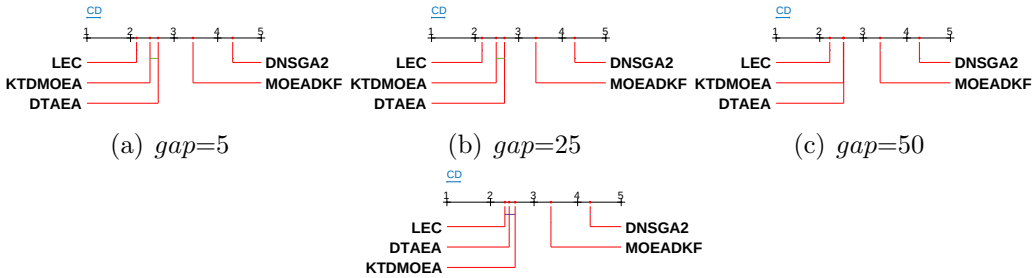
Additionally, it can be seen from the 'InNObj' row of Table 6.8 that LEC gets slightly better MHV and slightly worse MGD than KTDMOEA when increasing NObj from 2 to 7. This implies that LEC also increases the diversity of KTDMOEA after optimization when increasing NObj. In addition, it can be found from the 'DeNObj' row of Table 6.8 that LEC gets significantly better MHV and MGD than all other algorithms when decreasing NObj from 7 to 2, which implies that LEC increases both convergence and diversity after optimization when decreasing NObj.




(a) *gap*=5 (b) *gap*=25 (c) *gap*=50

(d) *gap*=100

Figure 6.12: Friedman ranking among MHV of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=100.



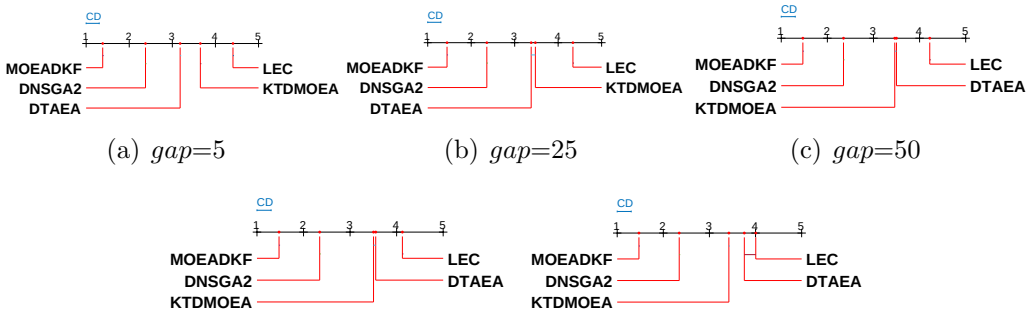

(a) *gap*=5 (b) *gap*=25 (c) *gap*=50

(d) *gap*=100

Figure 6.13: Friedman ranking among GD of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=100.

Note that for all frequencies of changes, when *gap* is smaller, solution quality obtained by LEC is significantly better than DTAEA in most cases. This implies that our proposed LEC is more suitable for dynamic optimization, as better solutions are urged to be found given limited budgets in real-word dynamic scenarios.

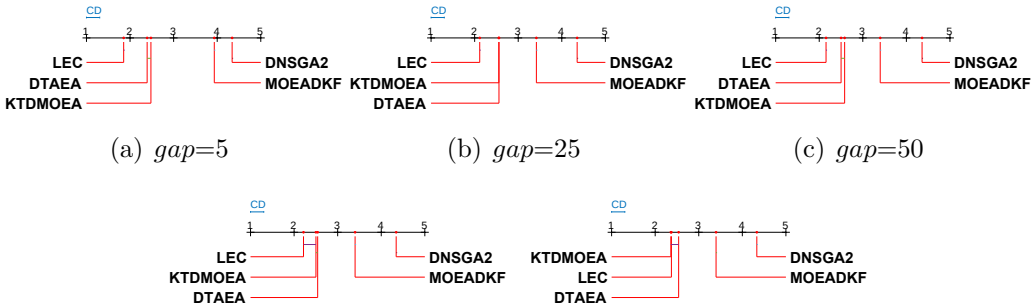
Results of mean and standard deviation values for MHV and MGD of obtained solutions at the last generation after optimization at three different $\tau_t s$ are presented in Tables 6.10-6.11 for $\tau_t s$=50 regarding MHV and MGD, Tables 6.12-6.13 for $\tau_t s$=100 regarding MHV and MGD, and Tables 6.14-6.15 for $\tau_t s$=200 regarding MHV and MGD, respectively. It is clear from Tables 6.10, 6.12 and 6.14 that LEC




(a) *gap*=5 (b) *gap*=25 (c) *gap*=50

(d) *gap*=100 (e) *gap*=200

Figure 6.14: Friedman ranking among MHV of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=200.




(a) *gap*=5 (b) *gap*=25 (c) *gap*=50

(d) *gap*=100 (e) *gap*=200

Figure 6.15: Friedman ranking among GD of optimized solutions after *gap* generations by 5 algorithms when $\tau_t$=200.

performs the best on all test problems except for F4, WFG3, WFG6 and WFG9, at different frequencies of changes. Note for F4 and WFG6, LEC gets worse mean MHV values especially when gap is large.

Table 6.10: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 4.96E-01(1.98E-07)† | 8.70E-03(2.16E-04)† | 8.23E-01(4.40E-03)† | 8.10E-01(5.99E-03)† | 9.12E-01(9.06E-04) |
| | 25 | 4.90E-01(4.62E-04)† | 9.27E-02(6.97E-04)† | 9.54E-01(1.17E-03)† | 9.46E-01(4.50E-03)† | 9.88E-01(1.06E-04) |
| | 50 | 4.73E-01(2.02E-03)† | 1.19E-01(7.89E-04)† | 9.84E-01(1.55E-04)† | 9.79E-01(1.64E-03)† | 9.95E-01(4.73E-06) |
| F2 | 5 | 9.20E-01(3.76E-07)† | 6.42E-01(1.86E-03)† | 9.50E-01(9.27E-07)† | 9.57E-01(6.19E-08) | 9.57E-01(5.54E-07) |
| | 25 | 9.28E-01(4.27E-07)† | 6.59E-01(1.58E-03)† | 9.58E-01(2.16E-09)† | 9.58E-01(5.33E-09) | 9.58E-01(4.73E-08) |
| | 50 | 9.23E-01(7.88E-07)† | 6.57E-01(1.59E-03)† | 9.58E-01(1.39E-09)† | 9.58E-01(2.89E-09) | 9.58E-01(1.56E-09) |
| F3 | 5 | 4.63E-01(3.65E-04)† | 1.02E-05(3.20E-09)† | 5.32E-01(5.00E-03)† | 6.00E-01(1.21E-03)† | 7.11E-01(3.34E-03) |
| | 25 | 4.64E-01(7.89E-05)† | 2.12E-02(3.70E-04)† | 7.12E-01(1.55E-02)† | 7.04E-01(1.34E-02)† | 9.09E-01(4.54E-03) |
| | 50 | 4.24E-01(1.15E-03)† | 8.53E-02(1.48E-04)† | 8.15E-01(1.33E-02)† | 8.02E-01(1.33E-02)† | 9.42E-01(2.24E-03) |
| F4 | 5 | 8.80E-01(9.23E-07)† | 9.21E-01(3.23E-05)† | 9.41E-01(7.20E-06)† | 8.74E-01(8.77E-04)† | 9.45E-01(2.21E-04) |
| | 25 | 8.82E-01(8.14E-06)† | 9.39E-01(9.97E-06)† | 9.58E-01(2.15E-08) | 8.79E-01(7.80E-04)† | 9.49E-01(2.21E-04) |
| | 50 | 8.84E-01(3.34E-06)† | 9.45E-01(3.24E-05)† | 9.58E-01(1.99E-09) | 8.79E-01(7.84E-04)† | 9.49E-01(2.24E-04) |
| WFG1 | 5 | 9.32E-01(4.43E-06)† | 3.06E-01(1.01E-05)† | 9.17E-01(5.22E-06)† | 9.37E-01(5.15E-06)† | 9.44E-01(1.26E-05) |
| | 25 | 9.50E-01(1.96E-06)† | 3.10E-01(1.09E-05)† | 9.45E-01(9.79E-06)† | 9.53E-01(4.14E-06)† | 9.58E-01(4.77E-06) |
| | 50 | 9.57E-01(1.95E-06)† | 3.10E-01(1.64E-05)† | 9.60E-01(2.52E-06)† | 9.62E-01(1.64E-06)† | 9.64E-01(1.82E-06) |
| WFG2 | 5 | 9.50E-01(3.79E-07)† | 2.41E-01(1.65E-08)† | 9.58E-01(1.01E-06)† | 9.64E-01(2.24E-07)‡ | 9.63E-01(4.00E-07) |
| | 25 | 9.64E-01(1.91E-08)† | 2.42E-01(2.71E-09)† | 9.67E-01(1.82E-08)† | 9.67E-01(7.27E-09) | 9.67E-01(1.45E-08) |
| | 50 | 9.64E-01(7.01E-09)† | 2.42E-01(1.68E-09)† | 9.68E-01(3.84E-09)† | 9.68E-01(3.02E-09) | 9.68E-01(3.88E-09) |
| WFG3 | 5 | 6.87E-01(2.98E-07)‡ | 2.52E-01(1.61E-07)† | 6.60E-01(1.72E-05)† | 6.73E-01(2.90E-05)† | 6.80E-01(4.83E-05) |
| | 25 | 7.08E-01(4.24E-07)‡ | 2.53E-01(1.65E-08)† | 6.86E-01(4.90E-06)† | 6.91E-01(1.06E-05)† | 6.95E-01(1.17E-05) |
| | 50 | 7.14E-01(2.45E-08)‡ | 2.54E-01(1.19E-08)† | 6.92E-01(3.37E-06)† | 6.96E-01(5.24E-06)† | 6.98E-01(3.66E-06) |
| WFG4 | 5 | 6.95E-01(5.05E-07)† | 3.19E-01(2.41E-05)† | 7.86E-01(8.74E-06)† | 8.24E-01(2.75E-07)† | 8.25E-01(2.22E-07) |
| | 25 | 7.19E-01(6.30E-06)† | 3.48E-01(8.72E-06)† | 8.29E-01(8.44E-07)† | 8.31E-01(5.87E-08)† | 8.31E-01(4.01E-08) |
| | 50 | 7.32E-01(7.36E-07)† | 3.46E-01(1.42E-05)† | 8.32E-01(2.36E-08) | 8.32E-01(4.55E-08)† | 8.32E-01(1.70E-08) |
| WFG5 | 5 | 6.75E-01(6.58E-06)† | 5.25E-01(9.53E-06)† | 7.61E-01(7.24E-06)† | 7.85E-01(2.01E-06)‡ | 7.83E-01(1.86E-06) |
| | 25 | 7.00E-01(5.39E-06)† | 5.35E-01(2.48E-06)† | 7.93E-01(3.33E-07) | 7.92E-01(9.82E-07) | 7.91E-01(9.13E-07) |
| | 50 | 7.09E-01(1.49E-06)† | 5.32E-01(1.85E-06)† | 7.96E-01(5.36E-07) | 7.95E-01(1.27E-06) | 7.94E-01(1.04E-06) |
| WFG6 | 5 | 6.72E-01(8.66E-07)† | 1.43E-01(7.58E-06)† | 7.36E-01(1.42E-04)† | 7.47E-01(3.73E-04)† | 7.48E-01(2.33E-04) |
| | 25 | 7.18E-01(3.98E-06)† | 1.45E-01(1.55E-05)† | 7.92E-01(1.54E-04)‡ | 7.59E-01(3.62E-04)† | 7.61E-01(2.44E-04) |
| | 50 | 7.35E-01(1.62E-06)† | 1.46E-01(1.85E-05)† | 7.98E-01(1.45E-04)‡ | 7.61E-01(3.49E-04)† | 7.64E-01(2.62E-04) |
| WFG7 | 5 | 7.18E-01(4.27E-06)† | 6.79E-01(2.90E-05)† | 7.97E-01(7.68E-06)† | 8.16E-01(7.27E-07)† | 8.18E-01(9.36E-07) |
| | 25 | 7.60E-01(1.21E-06)† | 7.34E-01(2.83E-05)† | 8.30E-01(6.06E-08)† | 8.31E-01(6.09E-08)† | 8.31E-01(1.14E-07) |
| | 50 | 7.72E-01(1.19E-06)† | 7.46E-01(2.13E-05)† | 8.32E-01(3.60E-08)† | 8.32E-01(2.87E-08)† | 8.32E-01(2.76E-08) |
| WFG8 | 5 | 7.11E-01(1.02E-06)† | 6.76E-01(4.27E-05)† | 7.97E-01(6.73E-06)† | 8.16E-01(8.15E-07)† | 8.18E-01(1.35E-06) |
| | 25 | 7.59E-01(2.56E-07)† | 7.34E-01(2.13E-05)† | 8.30E-01(9.03E-08)† | 8.31E-01(1.02E-07)† | 8.31E-01(4.86E-08) |
| | 50 | 7.70E-01(1.28E-07)† | 7.47E-01(1.60E-05)† | 8.32E-01(3.43E-08)† | 8.32E-01(2.44E-08)† | 8.32E-01(2.62E-08) |
| WFG9 | 5 | 5.48E-01(2.16E-06)‡ | 6.08E-01(1.28E-02)‡ | 5.76E-01(2.51E-03)† | 4.55E-01(5.14E-04)† | 4.63E-01(2.34E-04) |
| | 25 | 6.16E-01(1.00E-05)‡ | 6.72E-01(1.11E-02)‡ | 5.99E-01(2.98E-03)‡ | 4.61E-01(5.22E-04)† | 4.70E-01(2.94E-04) |
| | 50 | 6.32E-01(8.39E-07)‡ | 6.88E-01(1.01E-02)‡ | 6.02E-01(2.99E-03)‡ | 4.64E-01(4.91E-04)† | 4.74E-01(3.29E-04) |
| +-= | | 33/6/0 | 36/3/0 | 30/4/5 | 30/2/7 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.11: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 7.96E+01(5.03E+01)† | 1.09E+03(8.59E+02)† | 8.71E+00(1.12E+00)† | 1.00E+01(1.18E+00)† | 7.22E+00(2.32E+00) |
| | 25 | 7.03E+01(3.58E+01)† | 1.25E+03(1.56E+03)† | 3.06E+00(8.70E-01)† | 3.29E+00(9.81E-01)† | 1.01E+00(3.13E-01) |
| | 50 | 7.44E+01(6.03E+01)† | 1.30E+03(1.27E+03)† | 9.40E-01(1.86E-01)† | 1.17E+00(3.80E-01)† | 1.76E-01(1.40E-02) |
| F2 | 5 | 3.16E-01(4.47E-05)† | 8.94E+02(7.06E+02)† | 6.68E-02(4.36E-06)† | 5.55E-02(2.42E-06)‡ | 5.59E-02(7.30E-06) |
| | 25 | 3.01E-01(3.12E-06)† | 9.43E+02(2.03E+02)† | 4.34E-02(2.33E-06)† | 4.01E-02(1.24E-06)† | 3.87E-02(1.49E-06) |
| | 50 | 3.27E-01(1.61E-05)† | 9.49E+02(1.81E+02)† | 3.68E-02(7.59E-07)† | 3.58E-02(3.94E-07)† | 3.47E-02(3.37E-07) |
| F3 | 5 | 1.36E+02(2.35E+02)† | 1.02E+03(9.46E+02)† | 2.22E+01(6.80E+00)† | 2.55E+01(1.53E+01)† | 1.86E+01(1.13E+01) |
| | 25 | 1.25E+02(2.19E+02)† | 1.02E+03(1.13E+03)† | 9.53E+00(7.73E+00)† | 9.69E+00(5.74E+00)† | 3.34E+00(2.44E+00) |
| | 50 | 1.31E+02(2.02E+02)† | 1.00E+03(2.09E+02)† | 3.15E+00(2.09E+00)† | 4.20E+00(3.49E+00)† | 6.84E-01(4.97E-01) |
| F4 | 5 | 4.64E-01(1.70E-05)† | 1.16E-01(3.27E-04)† | 7.98E-02(6.53E-05)† | 4.70E-02(1.47E-04)‡ | 5.56E-02(3.31E-05) |
| | 25 | 4.04E-01(5.06E-05)† | 4.48E-02(4.02E-05)† | 3.91E-02(2.06E-06)† | 2.49E-02(8.12E-06)‡ | 3.28E-02(4.05E-06) |
| | 50 | 4.29E-01(1.85E-05)† | 3.85E-02(3.41E-05)† | 3.25E-02(2.05E-07)† | 2.34E-02(9.36E-06)‡ | 3.04E-02(1.94E-06) |
| WFG1 | 5 | 3.06E-01(1.01E-05)† | 1.05E+03(1.19E+03)† | 2.48E-01(2.76E-05)† | 2.32E-01(2.68E-05)† | 2.29E-01(2.42E-05) |
| | 25 | 2.91E-01(1.38E-05)† | 1.10E+03(9.49E+02)† | 2.31E-01(3.74E-05)† | 2.06E-01(3.55E-05)† | 1.98E-01(2.24E-05) |
| | 50 | 2.86E-01(3.89E-06)† | 1.13E+03(5.29E+02)† | 2.00E-01(2.80E-05)† | 1.88E-01(3.07E-05)† | 1.86E-01(1.82E-05) |
| WFG2 | 5 | 7.27E-01(2.05E-05)† | 1.31E+00(8.10E-05)† | 4.86E-01(6.66E-05)‡ | 5.11E-01(1.71E-04) | 5.22E-01(1.54E-04) |
| | 25 | 7.23E-01(1.69E-04)† | 1.31E+00(2.89E-05)† | 4.49E-01(5.47E-05)‡ | 4.67E-01(5.20E-05) | 4.72E-01(9.42E-05) |
| | 50 | 7.23E-01(9.85E-06)† | 1.31E+00(1.32E-05)† | 4.53E-01(6.60E-05)‡ | 4.69E-01(5.30E-05) | 4.69E-01(8.75E-05) |
| WFG3 | 5 | 1.72E+00(1.34E-04)† | 1.47E+00(3.45E-05)† | 1.55E+00(9.38E-04)† | 1.50E+00(4.83E-03)† | 1.45E+00(2.91E-03) |
| | 25 | 1.89E+00(1.90E-04)† | 1.46E+00(9.82E-06)‡ | 1.63E+00(4.69E-04)† | 1.51E+00(2.99E-03)† | 1.50E+00(1.73E-03) |
| | 50 | 1.94E+00(1.20E-04)† | 1.46E+00(8.36E-06)‡ | 1.68E+00(3.91E-04)† | 1.57E+00(2.76E-03)† | 1.55E+00(2.07E-03) |
| WFG4 | 5 | 5.36E-01(1.48E-05)† | 1.03E+00(7.50E-05)† | 2.52E-01(9.92E-06)‡ | 2.62E-01(1.37E-05)† | 2.57E-01(9.46E-06) |
| | 25 | 5.19E-01(3.90E-05)† | 9.53E-01(5.27E-05)† | 2.23E-01(1.13E-05)† | 2.22E-01(6.16E-06)† | 2.20E-01(7.48E-06) |
| | 50 | 5.06E-01(1.62E-05)† | 9.26E-01(4.18E-05)† | 2.13E-01(6.88E-06)† | 2.14E-01(6.33E-06)† | 2.12E-01(5.85E-06) |
| WFG5 | 5 | 5.66E-01(7.31E-06)† | 2.47E-01(3.77E-05)‡ | 3.03E-01(1.69E-05)† | 2.94E-01(3.09E-05)‡ | 3.00E-01(3.18E-05) |
| | 25 | 5.26E-01(1.24E-05)† | 2.12E-01(1.67E-04)‡ | 2.64E-01(8.78E-06) | 2.62E-01(1.03E-05)‡ | 2.64E-01(1.07E-05) |
| | 50 | 5.17E-01(1.14E-05)† | 1.86E-01(4.08E-05)‡ | 2.53E-01(4.47E-06)‡ | 2.54E-01(1.06E-05) | 2.56E-01(9.59E-06) |
| WFG6 | 5 | 5.90E-01(8.53E-07)† | 9.26E-01(5.69E-04)† | 3.55E-01(4.85E-04)‡ | 3.84E-01(1.05E-03)‡ | 3.90E-01(5.84E-04) |
| | 25 | 5.28E-01(1.52E-06)† | 9.13E-01(1.49E-03)† | 2.78E-01(5.03E-04)‡ | 3.24E-01(1.14E-03)† | 3.19E-01(7.26E-04) |
| | 50 | 5.00E-01(6.38E-06)† | 9.11E-01(1.84E-03)† | 2.60E-01(5.19E-04)‡ | 3.15E-01(1.20E-03)† | 3.10E-01(7.78E-04) |
| WFG7 | 5 | 5.69E-01(1.34E-05)† | 2.47E-01(7.69E-05)‡ | 2.78E-01(9.30E-06)† | 2.68E-01(1.68E-05)† | 2.66E-01(1.10E-05) |
| | 25 | 5.33E-01(1.03E-05)† | 1.67E-01(5.30E-05)‡ | 2.15E-01(7.69E-06)† | 2.13E-01(7.79E-06) | 2.13E-01(7.91E-06) |
| | 50 | 5.21E-01(1.74E-06)† | 1.59E-01(2.98E-05)‡ | 2.07E-01(6.75E-06) | 2.08E-01(4.82E-06) | 2.07E-01(4.93E-06) |
| WFG8 | 5 | 5.73E-01(1.94E-07)† | 2.45E-01(4.59E-05)‡ | 2.78E-01(1.81E-05)† | 2.69E-01(9.86E-06)† | 2.66E-01(1.45E-05) |
| | 25 | 5.31E-01(5.43E-07)† | 1.67E-01(6.16E-05)‡ | 2.14E-01(6.01E-06)† | 2.13E-01(6.32E-06) | 2.13E-01(7.54E-06) |
| | 50 | 5.11E-01(8.08E-06)† | 1.62E-01(4.05E-05)‡ | 2.07E-01(5.17E-06) | 2.07E-01(5.41E-06) | 2.07E-01(4.55E-06) |
| WFG9 | 5 | 8.95E-01(4.88E-06)‡ | 5.26E-01(5.31E-02)‡ | 7.41E-01(8.86E-03)‡ | 8.96E-01(7.56E-04) | 9.09E-01(3.59E-04) |
| | 25 | 8.23E-01(1.39E-05)‡ | 3.74E-01(4.05E-02)‡ | 6.90E-01(1.10E-02)‡ | 8.83E-01(6.01E-04) | 8.94E-01(2.88E-04) |
| | 50 | 7.91E-01(1.77E-06)‡ | 3.28E-01(3.49E-02)‡ | 6.78E-01(1.05E-02)‡ | 8.78E-01(4.84E-04) | 8.86E-01(2.40E-04) |
| +-= | | 36/3/0 | 25/14/0 | 25/11/3 | 21/7/11 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

**6.4.2.1.1 When Increasing the NObj** Results of mean and standard deviation values for MHV and MGD of obtained solutions at the last generation after optimization at three different $\tau_t s$ ($\tau_t$=50, 100, 200) on all DMOPs with a changing NObjs averaged across the cases of increasing NObj from 2 to 7, are presented in Tables 6.16, 6.18 and 6.20 regarding HV, Tables 6.17, 6.19 and 6.21 for MGD, respectively.

**6.4.2.1.2 When Decreasing NObj** Results of mean and standard deviation values for MHV and MGD of obtained solutions at the last generation after optimization at three different $\tau_t s$ ($\tau_t$=50, 100, 200) on all DMOPs with a changing NObjs averaged across the cases of decreasing NObj from 2 to 7, are presented in Tables 6.22, 6.24 and 6.26 regarding HV, Tables 6.23, 6.25 and 6.27 for GD, respectively.

#### 6.4.2.2 Why Does Learnt PS Expansion and Contraction Help Optimization Process after the Changes?

It has been presented in Section 6.4.1 that the proposed LEC has indeed obtained better solution quality than other state-of-the-arts in the first generation after changes. Given these results, our proposed LEC is able to quickly find solutions with good convergence and diversity at all frequencies of change. In other words, our proposal obtained good solutions at the initial stage of the evolution (i.e. when *gap* is small), which is support by the Friedman test results in Section 6.4.2.1. This means that our proposed approach is robust to different gaps of outputting optimized solutions under different frequencies of change.

Because the transferred solutions are better distributed and converged in the new environment, LEC is able to find better solutions across different frequencies of change. This is also the reason why LEC is able to quickly respond to the changes in the NObj, since finding good solution under high frequency of change means fast response to changes. When *gap* is large, there is less need for obtaining a good initial population right after the changes, because more time can be spend searching for good solutions through the optimization process. There are some specific problem instances where LEC did not perform best when the frequency of changes is high. The specific results and analyses are presented in next section.

#### 6.4.2.3 Why Doesn't Learnt PS Expansion and Contraction sometimes Help Optimization Process after the Changes?

In order to see at which stage of changing LEC performs worse, the HV trajectory of sampled generations over evolution on these four problems (F4, WFG3, WFG6

Table 6.12: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 4.06E-01(2.96E-06)† | 1.04E-02(2.88E-04)† | 8.92E-01(2.79E-03)† | 8.93E-01(2.68E-03)† | 9.18E-01(5.19E-04) |
| | 25 | 4.75E-01(1.19E-04)† | 1.07E-01(1.00E-03)† | 9.74E-01(1.12E-03)† | 9.85E-01(3.69E-04)† | 9.94E-01(1.52E-06) |
| | 50 | 4.92E-01(2.20E-04)† | 1.29E-01(1.10E-03)† | 9.86E-01(4.17E-04)† | 9.94E-01(1.22E-05)† | 9.96E-01(1.27E-07) |
| | 100 | 4.91E-01(4.43E-04)† | 1.31E-01(1.44E-03)† | 9.95E-01(9.94E-06)† | 9.96E-01(3.66E-07)† | 9.96E-01(2.02E-08) |
| F2 | 5 | 9.18E-01(2.47E-06)† | 7.39E-01(5.15E-04)† | 9.50E-01(7.39E-07)† | 9.57E-01(5.48E-08) | 9.57E-01(7.98E-08) |
| | 25 | 9.25E-01(6.68E-06)† | 7.50E-01(3.83E-04)† | 9.58E-01(3.05E-09)† | 9.58E-01(1.16E-08)† | 9.58E-01(2.77E-09) |
| | 50 | 9.21E-01(1.73E-05)† | 7.46E-01(3.90E-04)† | 9.58E-01(9.79E-10)† | 9.58E-01(1.49E-09)† | 9.58E-01(8.29E-10) |
| | 100 | 9.18E-01(2.98E-06)† | 7.46E-01(3.89E-04)† | 9.59E-01(4.54E-10)† | 9.59E-01(6.39E-10)† | 9.59E-01(4.15E-10) |
| F3 | 5 | 3.99E-01(1.22E-03)† | 1.86E-04(1.07E-06)† | 5.79E-01(3.50E-03)† | 6.49E-01(3.72E-03)† | 7.36E-01(2.34E-03) |
| | 25 | 3.87E-01(3.41E-04)† | 2.36E-02(2.94E-04)† | 8.09E-01(8.94E-03)† | 8.37E-01(1.16E-02)† | 9.29E-01(5.11E-04) |
| | 50 | 4.07E-01(4.85E-04)† | 8.27E-02(8.71E-05)† | 8.89E-01(5.70E-03)† | 9.03E-01(5.64E-03)† | 9.53E-01(1.91E-05) |
| | 100 | 3.97E-01(1.56E-03)† | 9.22E-02(3.33E-08)† | 9.38E-01(1.23E-03)† | 9.38E-01(9.75E-04)† | 9.56E-01(1.55E-06) |
| F4 | 5 | 8.79E-01(3.44E-06)† | 9.23E-01(1.63E-05)† | 9.41E-01(7.55E-06)† | 8.96E-01(6.06E-04)† | 9.43E-01(6.67E-04) |
| | 25 | 8.80E-01(5.60E-06)† | 9.39E-01(1.52E-05)† | 9.58E-01(1.07E-08)‡ | 8.99E-01(5.67E-04)† | 9.46E-01(6.41E-04) |
| | 50 | 8.71E-01(7.16E-07)† | 9.46E-01(3.34E-05)† | 9.58E-01(1.98E-09)‡ | 8.99E-01(5.74E-04)† | 9.46E-01(6.50E-04) |
| | 100 | 8.64E-01(8.96E-07)† | 9.51E-01(2.97E-05)† | 9.59E-01(5.11E-10)‡ | 9.00E-01(5.84E-04)† | 9.46E-01(6.54E-04) |
| WFG1 | 5 | 9.27E-01(3.69E-07)† | 3.08E-01(7.97E-06)† | 9.25E-01(4.08E-06)† | 9.40E-01(3.49E-06)† | 9.45E-01(3.08E-06) |
| | 25 | 9.48E-01(7.79E-07)† | 3.12E-01(6.75E-06)† | 9.51E-01(5.73E-06)† | 9.56E-01(2.97E-06)† | 9.60E-01(1.31E-06) |
| | 50 | 9.56E-01(3.61E-07)† | 3.12E-01(1.08E-05)† | 9.63E-01(1.41E-06)† | 9.64E-01(9.04E-07)† | 9.66E-01(9.90E-07) |
| | 100 | 9.62E-01(1.39E-07)† | 3.12E-01(9.64E-06)† | 9.69E-01(4.01E-07)† | 9.69E-01(3.57E-07)† | 9.69E-01(6.05E-07) |
| WFG2 | 5 | 9.49E-01(1.56E-07)† | 2.42E-01(7.83E-09)† | 9.56E-01(1.65E-05)† | 9.62E-01(4.91E-07)† | 9.64E-01(2.56E-07) |
| | 25 | 9.64E-01(6.61E-09)† | 2.42E-01(1.60E-09)† | 9.67E-01(1.81E-08)† | 9.68E-01(1.01E-08)† | 9.68E-01(9.70E-09) |
| | 50 | 9.65E-01(5.16E-09)† | 2.42E-01(9.40E-10)† | 9.68E-01(2.12E-09)† | 9.68E-01(3.43E-09)† | 9.68E-01(2.57E-09) |
| | 100 | 9.65E-01(3.17E-10)† | 2.42E-01(1.51E-09)† | 9.68E-01(2.86E-09) | 9.68E-01(2.56E-09) | 9.68E-01(4.01E-09) |
| WFG3 | 5 | 6.84E-01(1.79E-06)‡ | 2.52E-01(8.69E-08)† | 6.59E-01(1.65E-05)† | 6.71E-01(2.89E-05)† | 6.74E-01(2.84E-05) |
| | 25 | 7.12E-01(5.07E-08)‡ | 2.53E-01(1.42E-08)† | 6.86E-01(7.32E-06)† | 6.91E-01(1.36E-05)† | 6.93E-01(7.07E-06) |
| | 50 | 7.13E-01(7.06E-08)‡ | 2.54E-01(8.29E-09)† | 6.93E-01(5.43E-06)† | 6.96E-01(6.74E-06)† | 6.97E-01(4.11E-06) |
| | 100 | 7.14E-01(1.45E-07)‡ | 2.54E-01(5.01E-09)† | 6.97E-01(2.88E-06)† | 6.98E-01(3.97E-06)† | 6.99E-01(2.40E-06) |
| WFG4 | 5 | 7.02E-01(4.06E-07)† | 3.22E-01(2.02E-05)† | 7.89E-01(1.31E-05)† | 8.24E-01(1.91E-07)† | 8.26E-01(1.44E-07) |
| | 25 | 7.21E-01(6.93E-06)† | 3.48E-01(7.49E-06)† | 8.30E-01(1.60E-07)† | 8.31E-01(2.63E-08)† | 8.31E-01(3.91E-08) |
| | 50 | 7.29E-01(8.01E-06)† | 3.48E-01(7.98E-06)† | 8.32E-01(3.20E-08)† | 8.32E-01(2.51E-08)† | 8.32E-01(1.84E-08) |
| | 100 | 7.41E-01(6.46E-06)† | 3.45E-01(4.27E-06)† | 8.32E-01(2.20E-08)† | 8.33E-01(1.98E-08)† | 8.33E-01(1.65E-08) |
| WFG5 | 5 | 6.77E-01(5.72E-06)† | 5.25E-01(9.67E-06)† | 7.62E-01(6.87E-06)† | 7.88E-01(1.38E-06) | 7.87E-01(4.63E-07) |
| | 25 | 7.00E-01(1.19E-06)† | 5.36E-01(1.67E-06)† | 7.93E-01(1.70E-07)† | 7.94E-01(1.28E-06)† | 7.95E-01(3.57E-07) |
| | 50 | 7.06E-01(8.23E-08)† | 5.33E-01(2.66E-06)† | 7.96E-01(2.17E-07)† | 7.97E-01(1.45E-06)† | 7.97E-01(4.13E-07) |
| | 100 | 7.16E-01(8.56E-07)† | 5.31E-01(1.21E-06)† | 7.98E-01(7.09E-08)† | 7.98E-01(1.31E-06)† | 7.98E-01(2.68E-07) |
| WFG6 | 5 | 6.73E-01(2.09E-05)† | 1.44E-01(6.58E-06)† | 7.27E-01(3.16E-04)† | 7.63E-01(2.86E-04)‡ | 7.47E-01(4.32E-04) |
| | 25 | 7.15E-01(2.21E-07)† | 1.46E-01(1.42E-05)† | 7.82E-01(3.90E-04)‡ | 7.76E-01(3.30E-04)‡ | 7.68E-01(3.59E-04) |
| | 50 | 7.27E-01(4.31E-07)† | 1.46E-01(1.62E-05)† | 7.88E-01(3.72E-04)‡ | 7.79E-01(3.49E-04)‡ | 7.71E-01(3.80E-04) |
| | 100 | 7.40E-01(3.22E-06)† | 1.46E-01(1.72E-05)† | 7.94E-01(3.67E-04)‡ | 7.82E-01(3.62E-04)‡ | 7.75E-01(4.00E-04) |
| WFG7 | 5 | 7.15E-01(5.73E-07)† | 6.97E-01(4.49E-05)† | 7.98E-01(3.27E-06)† | 8.16E-01(5.95E-07)† | 8.18E-01(6.53E-07) |
| | 25 | 7.60E-01(5.50E-07)† | 7.47E-01(2.77E-05)† | 8.30E-01(4.61E-08)† | 8.31E-01(1.02E-07)† | 8.31E-01(4.49E-08) |
| | 50 | 7.70E-01(5.65E-07)† | 7.55E-01(2.11E-05)† | 8.32E-01(1.94E-08)† | 8.32E-01(3.75E-08)† | 8.32E-01(2.54E-08) |
| | 100 | 7.76E-01(6.27E-07)† | 7.62E-01(2.75E-05)† | 8.32E-01(1.98E-08)† | 8.33E-01(1.51E-08)† | 8.33E-01(1.78E-08) |
| WFG8 | 5 | 7.18E-01(6.94E-06)† | 6.97E-01(3.43E-05)† | 7.98E-01(4.71E-06)† | 8.16E-01(1.18E-06)† | 8.18E-01(9.88E-07) |
| | 25 | 7.62E-01(1.42E-06)† | 7.48E-01(2.18E-05)† | 8.30E-01(1.05E-07)† | 8.31E-01(7.74E-08)† | 8.31E-01(6.62E-08) |
| | 50 | 7.70E-01(9.32E-07)† | 7.57E-01(1.85E-05)† | 8.32E-01(4.14E-08)† | 8.32E-01(4.92E-08)† | 8.32E-01(3.08E-08) |
| | 100 | 7.78E-01(4.01E-07)† | 7.64E-01(2.19E-05)† | 8.32E-01(2.26E-08)† | 8.33E-01(1.53E-08)† | 8.33E-01(1.60E-08) |
| WFG9 | 5 | 5.62E-01(2.50E-05)‡ | 6.40E-01(8.21E-03)‡ | 5.60E-01(1.98E-03)‡ | 4.60E-01(2.94E-04)† | 4.66E-01(2.51E-04) |
| | 25 | 6.21E-01(2.88E-05)‡ | 6.96E-01(5.36E-03)‡ | 5.81E-01(2.41E-03)‡ | 4.66E-01(3.48E-04)† | 4.72E-01(3.29E-04) |
| | 50 | 6.36E-01(5.38E-05)‡ | 7.11E-01(4.37E-03)‡ | 5.84E-01(2.38E-03)‡ | 4.68E-01(3.59E-04)† | 4.75E-01(3.62E-04) |
| | 100 | 6.44E-01(5.62E-05)‡ | 7.18E-01(3.79E-03)‡ | 5.88E-01(2.27E-03)‡ | 4.73E-01(3.00E-04)† | 4.79E-01(3.57E-04) |
| +-= | | 44/8/0 | 48/4/0 | 41/10/1 | 45/4/3 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.13: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), when firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 9.75E+01(4.44E+00)† | 1.11E+03(4.77E+02)† | 7.58E+00(8.46E-01)† | 9.31E+00(1.30E+00)† | 6.19E+00(8.09E-01) |
| | 25 | 8.93E+01(3.78E+00)† | 1.25E+03(1.20E+03)† | 2.15E+00(4.17E-01)† | 2.53E+00(5.29E-01)† | 7.49E-01(1.26E-01) |
| | 50 | 8.72E+01(1.24E+00)† | 1.31E+03(6.07E+02)† | 6.48E-01(9.55E-02)† | 7.17E-01(5.41E-02)† | 1.72E-01(1.44E-02) |
| | 100 | 9.26E+01(9.54E+00)† | 1.35E+03(6.78E+02)† | 1.87E-01(1.30E-02)† | 2.97E-01(1.39E-02)† | 6.10E-02(1.50E-03) |
| F2 | 5 | 3.08E-01(6.35E-06)† | 8.65E+02(1.34E+03)† | 6.49E-02(4.17E-06)† | 5.42E-02(4.41E-06)‡ | 5.65E-02(4.53E-06) |
| | 25 | 3.03E-01(8.01E-06)† | 9.32E+02(5.25E+02)† | 4.24E-02(1.39E-06)† | 3.91E-02(1.51E-06)† | 3.88E-02(1.03E-06) |
| | 50 | 3.21E-01(1.97E-05)† | 9.41E+02(3.49E+02)† | 3.59E-02(3.42E-07)† | 3.49E-02(3.85E-07) | 3.50E-02(4.19E-07) |
| | 100 | 3.42E-01(2.75E-05)† | 9.51E+02(3.37E+02)† | 3.27E-02(2.36E-07)‡ | 3.26E-02(9.19E-08)‡ | 3.27E-02(1.51E-07) |
| F3 | 5 | 1.91E+02(1.97E+02)† | 1.03E+03(8.50E+02)† | 1.85E+01(5.86E+00)† | 2.16E+01(6.94E+00)† | 1.64E+01(9.62E+00) |
| | 25 | 1.73E+02(2.66E+02)† | 1.01E+03(9.27E+02)† | 6.54E+00(4.25E+00)† | 7.12E+00(2.23E+00)† | 2.66E+00(1.39E+00) |
| | 50 | 1.76E+02(2.04E+02)† | 9.98E+02(1.06E+02)† | 2.08E+00(8.04E-01)† | 2.59E+00(9.27E-01)† | 5.01E-01(1.48E-01) |
| | 100 | 1.90E+02(2.58E+02)† | 9.74E+02(7.93E+01)† | 6.69E-01(2.28E-01)† | 9.68E-01(1.69E-01)† | 1.78E-01(1.56E-02) |
| F4 | 5 | 4.83E-01(1.92E-06)† | 1.05E-01(2.22E-04)† | 7.48E-02(3.83E-05)† | 4.61E-02(1.26E-04)‡ | 5.41E-02(5.59E-05) |
| | 25 | 4.41E-01(1.15E-06)† | 4.30E-02(2.68E-05)† | 3.74E-02(1.32E-06)† | 2.67E-02(6.59E-06)‡ | 3.21E-02(6.34E-06) |
| | 50 | 4.48E-01(1.46E-06)† | 3.94E-02(2.28E-05)† | 3.22E-02(2.77E-07)† | 2.55E-02(6.36E-06)‡ | 3.00E-02(3.97E-06) |
| | 100 | 4.56E-01(1.71E-07)† | 3.90E-02(2.18E-05)† | 3.14E-02(1.23E-07)† | 2.51E-02(7.09E-06)‡ | 3.04E-02(4.23E-06) |
| WFG1 | 5 | 3.15E-01(6.57E-06)† | 1.06E+03(6.52E+02)† | 2.29E-01(1.36E-05)† | 2.22E-01(1.53E-05)† | 2.22E-01(1.16E-05) |
| | 25 | 3.02E-01(7.42E-06)† | 1.11E+03(5.67E+02)† | 2.14E-01(3.38E-05)† | 1.98E-01(2.48E-05)† | 1.90E-01(9.71E-06) |
| | 50 | 2.86E-01(3.34E-06)† | 1.13E+03(5.09E+02)† | 1.90E-01(2.86E-05)† | 1.81E-01(1.96E-05)† | 1.80E-01(1.25E-05) |
| | 100 | 2.84E-01(1.76E-06)† | 1.15E+03(2.60E+02)† | 1.79E-01(1.55E-05)† | 1.74E-01(8.85E-06) | 1.74E-01(1.42E-05) |
| WFG2 | 5 | 7.54E-01(1.61E-04)† | 1.32E+00(7.18E-05)† | 4.99E-01(2.07E-04)‡ | 5.16E-01(1.54E-04)† | 5.09E-01(1.05E-04) |
| | 25 | 6.63E-01(2.45E-05)† | 1.31E+00(8.62E-06)† | 4.57E-01(8.07E-05)‡ | 4.66E-01(9.14E-05)† | 4.64E-01(9.23E-05) |
| | 50 | 6.79E-01(1.05E-04)† | 1.31E+00(1.03E-05)† | 4.59E-01(8.09E-05)‡ | 4.65E-01(6.75E-05)† | 4.64E-01(6.06E-05) |
| | 100 | 7.41E-01(1.18E-04)† | 1.31E+00(6.86E-06)† | 4.57E-01(4.61E-05)‡ | 4.59E-01(7.59E-05) | 4.61E-01(1.02E-04) |
| WFG3 | 5 | 1.82E+00(7.86E-04)† | 1.47E+00(2.74E-05)‡ | 1.57E+00(7.55E-04)† | 1.55E+00(2.88E-03) | 1.51E+00(2.14E-03) |
| | 25 | 1.93E+00(4.08E-04)† | 1.46E+00(1.24E-05)‡ | 1.64E+00(9.34E-04)† | 1.53E+00(2.42E-03) | 1.53E+00(2.10E-03) |
| | 50 | 1.98E+00(2.71E-04)† | 1.46E+00(1.00E-05)‡ | 1.68E+00(5.62E-04)† | 1.57E+00(2.39E-03) | 1.57E+00(1.94E-03) |
| | 100 | 1.99E+00(8.81E-06)† | 1.46E+00(6.16E-06)‡ | 1.70E+00(3.56E-04)† | 1.61E+00(1.90E-03) | 1.62E+00(1.05E-03) |
| WFG4 | 5 | 5.24E-01(1.51E-06)† | 1.01E+00(4.29E-05)† | 2.49E-01(1.03E-05)‡ | 2.60E-01(1.05E-05)† | 2.55E-01(8.31E-06) |
| | 25 | 5.11E-01(1.90E-05)† | 9.43E-01(4.51E-05)† | 2.19E-01(8.08E-06)† | 2.19E-01(6.43E-06)† | 2.19E-01(4.78E-06) |
| | 50 | 5.07E-01(1.50E-05)† | 9.20E-01(4.08E-05)† | 2.12E-01(5.38E-06)† | 2.14E-01(4.52E-06)† | 2.11E-01(6.76E-06) |
| | 100 | 5.00E-01(2.62E-05)† | 9.05E-01(1.74E-05)† | 2.10E-01(3.59E-06)† | 2.11E-01(4.60E-06)† | 2.08E-01(6.95E-06) |
| WFG5 | 5 | 5.55E-01(3.88E-06)† | 2.50E-01(3.06E-05)‡ | 3.00E-01(1.38E-05)† | 2.87E-01(2.36E-05)‡ | 2.91E-01(1.05E-05) |
| | 25 | 5.19E-01(1.52E-06)† | 2.13E-01(8.81E-05)‡ | 2.62E-01(5.71E-06)† | 2.59E-01(8.60E-06) | 2.59E-01(5.43E-06) |
| | 50 | 5.00E-01(2.81E-05)† | 1.89E-01(9.05E-05)‡ | 2.53E-01(2.85E-06)† | 2.53E-01(4.31E-06)† | 2.52E-01(7.61E-06) |
| | 100 | 4.97E-01(6.04E-06)† | 1.90E-01(7.50E-05)‡ | 2.47E-01(3.39E-06) | 2.47E-01(7.39E-06) | 2.47E-01(5.81E-06) |
| WFG6 | 5 | 5.88E-01(3.88E-05)† | 9.23E-01(4.21E-04)† | 3.82E-01(2.04E-03)‡ | 3.63E-01(6.92E-04)‡ | 4.00E-01(1.92E-03) |
| | 25 | 5.42E-01(3.74E-06)† | 9.09E-01(1.32E-03)† | 2.99E-01(1.50E-03)‡ | 2.96E-01(8.81E-04)‡ | 3.09E-01(1.00E-03) |
| | 50 | 5.18E-01(2.92E-06)† | 9.08E-01(1.63E-03)† | 2.79E-01(1.50E-03)‡ | 2.87E-01(9.54E-04)‡ | 2.99E-01(1.12E-03) |
| | 100 | 5.04E-01(6.82E-06)† | 9.07E-01(1.63E-03)† | 2.70E-01(1.49E-03)‡ | 2.82E-01(1.01E-03)‡ | 2.93E-01(1.14E-03) |
| WFG7 | 5 | 5.77E-01(2.60E-06)† | 2.50E-01(4.00E-05)‡ | 2.72E-01(9.86E-06)† | 2.65E-01(2.10E-05)† | 2.59E-01(1.10E-05) |
| | 25 | 5.32E-01(2.69E-06)† | 1.82E-01(5.64E-05)‡ | 2.10E-01(4.30E-06) | 2.12E-01(6.14E-06)† | 2.11E-01(7.29E-06) |
| | 50 | 5.23E-01(2.38E-06)† | 1.72E-01(4.48E-05)‡ | 2.04E-01(4.43E-06) | 2.08E-01(6.39E-06)† | 2.06E-01(7.11E-06) |
| | 100 | 5.13E-01(1.37E-06)† | 1.83E-01(5.22E-05)‡ | 2.07E-01(6.57E-06)‡ | 2.11E-01(1.11E-05) | 2.11E-01(4.85E-06) |
| WFG8 | 5 | 5.75E-01(7.23E-06)† | 2.54E-01(5.95E-05)‡ | 2.72E-01(1.38E-05)† | 2.64E-01(1.55E-05)† | 2.61E-01(1.12E-05) |
| | 25 | 5.27E-01(1.98E-06)† | 1.83E-01(8.16E-05)‡ | 2.10E-01(5.53E-06) | 2.11E-01(5.95E-06)† | 2.11E-01(5.94E-06) |
| | 50 | 5.21E-01(1.17E-06)† | 1.74E-01(5.01E-05)‡ | 2.05E-01(2.75E-06) | 2.07E-01(4.83E-06)† | 2.06E-01(7.30E-06) |
| | 100 | 5.17E-01(2.23E-06)† | 1.83E-01(4.35E-05)‡ | 2.08E-01(7.72E-06)‡ | 2.11E-01(7.61E-06) | 2.11E-01(6.40E-06) |
| WFG9 | 5 | 8.85E-01(6.58E-05)‡ | 4.75E-01(3.73E-02)‡ | 7.68E-01(7.08E-03)‡ | 8.99E-01(5.33E-04) | 9.04E-01(4.65E-04) |
| | 25 | 8.15E-01(3.83E-05)‡ | 3.26E-01(2.20E-02)‡ | 7.19E-01(8.94E-03)‡ | 8.88E-01(5.47E-04) | 8.89E-01(4.93E-04) |
| | 50 | 7.94E-01(8.11E-05)‡ | 2.86E-01(1.70E-02)‡ | 7.07E-01(8.61E-03)‡ | 8.84E-01(4.63E-04) | 8.84E-01(4.64E-04) |
| | 100 | 7.78E-01(1.33E-04)‡ | 2.63E-01(1.46E-02)‡ | 6.96E-01(7.87E-03)‡ | 8.82E-01(3.30E-04) | 8.79E-01(3.20E-04) |
| +-= | | 48/4/0 | 32/20/0 | 31/16/5 | 26/11/15 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.14: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 3.15E-01(1.01E-05)† | 5.56E-03(1.06E-04)† | 9.37E-01(1.18E-03)‡ | 9.05E-01(1.94E-03)† | 9.19E-01(1.99E-04) |
| | 25 | 3.94E-01(2.83E-08)† | 8.75E-02(6.13E-04)† | 9.94E-01(5.29E-06) | 9.78E-01(9.18E-04)† | 9.95E-01(1.01E-06) |
| | 50 | 3.96E-01(1.97E-08)† | 1.19E-01(7.65E-04)† | 9.96E-01(6.45E-07) | 9.90E-01(3.29E-04)† | 9.96E-01(3.76E-08) |
| | 100 | 4.56E-01(6.99E-05)† | 1.28E-01(1.38E-03)† | 9.96E-01(1.87E-07)† | 9.94E-01(3.97E-05)† | 9.96E-01(6.48E-09) |
| | 200 | 3.96E-01(1.13E-10)† | 1.89E-01(3.32E-03)† | 9.96E-01(8.77E-09)† | 9.96E-01(3.57E-08) | 9.96E-01(5.83E-09) |
| F2 | 5 | 9.19E-01(9.19E-07)† | 7.62E-01(2.97E-05)† | 9.50E-01(8.38E-07)† | 9.57E-01(2.15E-08) | 9.57E-01(4.39E-08) |
| | 25 | 9.23E-01(4.96E-06)† | 7.65E-01(1.70E-05)† | 9.58E-01(2.59E-09)† | 9.58E-01(5.71E-09)† | 9.58E-01(2.50E-09) |
| | 50 | 9.23E-01(3.09E-06)† | 7.62E-01(5.88E-06)† | 9.58E-01(1.07E-09)† | 9.58E-01(2.97E-09)† | 9.58E-01(1.53E-09) |
| | 100 | 9.16E-01(1.09E-05)† | 7.62E-01(5.85E-06)† | 9.59E-01(6.62E-10)† | 9.59E-01(5.92E-10)† | 9.59E-01(2.96E-10) |
| | 200 | 9.17E-01(7.42E-06)† | 7.62E-01(5.93E-06)† | 9.59E-01(1.23E-10)† | 9.59E-01(2.47E-10)† | 9.59E-01(1.40E-10) |
| F3 | 5 | 2.80E-01(1.46E-04)† | 0.00E+00(0.00E+00)† | 6.28E-01(3.94E-03)† | 6.90E-01(3.04E-03)† | 7.26E-01(2.58E-03) |
| | 25 | 3.57E-01(1.73E-05)† | 2.13E-02(2.11E-04)† | 8.78E-01(2.32E-03)† | 8.61E-01(4.24E-03)† | 9.32E-01(3.20E-04) |
| | 50 | 3.66E-01(1.12E-04)† | 8.21E-02(6.18E-05)† | 9.34E-01(6.85E-04)† | 9.11E-01(1.74E-03)† | 9.52E-01(2.04E-05) |
| | 100 | 3.96E-01(1.03E-04)† | 1.01E-01(4.51E-04)† | 9.51E-01(1.53E-04)† | 9.45E-01(4.86E-04)† | 9.57E-01(1.23E-06) |
| | 200 | 3.67E-01(4.57E-04)† | 1.35E-01(1.43E-03)† | 9.57E-01(4.96E-06)† | 9.53E-01(2.77E-04)† | 9.57E-01(5.03E-07) |
| F4 | 5 | 8.73E-01(1.98E-05)† | 9.25E-01(1.56E-05)† | 9.41E-01(1.12E-05)† | 8.70E-01(1.67E-03)† | 9.42E-01(1.73E-04) |
| | 25 | 8.67E-01(2.02E-05)† | 9.39E-01(2.49E-05)† | 9.58E-01(5.60E-08) | 8.77E-01(1.45E-03)† | 9.46E-01(1.84E-04) |
| | 50 | 8.73E-01(4.60E-06)† | 9.44E-01(2.91E-05) | 9.58E-01(1.20E-08) | 8.77E-01(1.51E-03)† | 9.47E-01(1.86E-04) |
| | 100 | 8.62E-01(2.81E-06)† | 9.52E-01(2.48E-05) | 9.59E-01(4.12E-10)‡ | 8.77E-01(1.52E-03)† | 9.47E-01(1.85E-04) |
| | 200 | 8.52E-01(5.09E-06)† | 9.54E-01(5.24E-06) | 9.59E-01(1.58E-10)‡ | 8.77E-01(1.52E-03)† | 9.47E-01(1.86E-04) |
| WFG1 | 5 | 9.25E-01(7.89E-07)† | 3.09E-01(2.84E-06)† | 9.27E-01(2.03E-06)† | 9.40E-01(4.15E-06)† | 9.46E-01(2.74E-06) |
| | 25 | 9.51E-01(1.23E-06)† | 3.13E-01(4.60E-06)† | 9.52E-01(7.23E-06)† | 9.57E-01(5.34E-06)† | 9.62E-01(1.35E-06) |
| | 50 | 9.60E-01(2.80E-07)† | 3.13E-01(3.19E-06)† | 9.64E-01(7.50E-07)† | 9.65E-01(1.26E-06)† | 9.67E-01(4.19E-07) |
| | 100 | 9.65E-01(4.36E-08)† | 3.13E-01(2.37E-06)† | 9.70E-01(1.41E-07)† | 9.69E-01(3.93E-07) | 9.70E-01(1.26E-07) |
| | 200 | 9.68E-01(1.18E-08)† | 3.14E-01(3.20E-06)† | 9.71E-01(6.19E-08) | 9.70E-01(2.81E-07) | 9.71E-01(9.10E-08) |
| WFG2 | 5 | 9.45E-01(1.20E-07)† | 2.42E-01(6.71E-09)† | 9.58E-01(1.35E-06)† | 9.63E-01(5.81E-07)† | 9.64E-01(3.83E-07) |
| | 25 | 9.64E-01(2.78E-08)† | 2.42E-01(1.10E-09)† | 9.67E-01(1.23E-08) | 9.68E-01(9.61E-09) | 9.68E-01(7.07E-09) |
| | 50 | 9.65E-01(3.90E-09)† | 2.42E-01(1.64E-09)† | 9.68E-01(2.39E-09) | 9.68E-01(3.50E-09) | 9.68E-01(2.70E-09) |
| | 100 | 9.65E-01(2.26E-09)† | 2.42E-01(1.66E-09)† | 9.68E-01(1.70E-09) | 9.68E-01(4.40E-09) | 9.68E-01(3.61E-09) |
| | 200 | 9.66E-01(1.62E-09)† | 2.42E-01(8.47E-10)† | 9.68E-01(5.25E-09) | 9.68E-01(3.78E-09) | 9.68E-01(3.83E-09) |
| WFG3 | 5 | 6.85E-01(2.63E-07)‡ | 2.53E-01(7.73E-08)† | 6.61E-01(1.71E-05)† | 6.59E-01(5.01E-05)† | 6.66E-01(4.76E-05) |
| | 25 | 7.11E-01(6.96E-08)‡ | 2.53E-01(1.06E-08)† | 6.87E-01(7.02E-06)† | 6.86E-01(1.69E-05)† | 6.88E-01(1.63E-05) |
| | 50 | 7.15E-01(2.48E-07)‡ | 2.54E-01(7.11E-09)† | 6.94E-01(4.84E-06)† | 6.93E-01(8.12E-06)† | 6.95E-01(2.38E-06) |
| | 100 | 7.15E-01(2.06E-08)‡ | 2.54E-01(5.50E-09)† | 6.98E-01(1.85E-06) | 6.97E-01(3.67E-06) | 6.97E-01(1.82E-06) |
| | 200 | 7.16E-01(2.30E-08)‡ | 2.54E-01(2.38E-09)† | 7.00E-01(1.33E-06) | 6.99E-01(2.80E-06) | 6.99E-01(1.52E-06) |
| WFG4 | 5 | 7.05E-01(3.10E-07)† | 3.24E-01(1.87E-05)† | 7.87E-01(8.48E-06)† | 8.24E-01(1.67E-07)† | 8.25E-01(1.68E-07) |
| | 25 | 7.26E-01(6.76E-09)† | 3.50E-01(9.66E-06)† | 8.29E-01(1.30E-07)† | 8.31E-01(2.89E-08)† | 8.31E-01(3.52E-08) |
| | 50 | 7.32E-01(1.89E-06)† | 3.48E-01(7.77E-06)† | 8.32E-01(1.86E-08)† | 8.32E-01(2.57E-08)† | 8.32E-01(2.37E-08) |
| | 100 | 7.38E-01(9.45E-06)† | 3.46E-01(5.66E-06)† | 8.32E-01(1.80E-08)† | 8.33E-01(1.93E-08)† | 8.33E-01(1.44E-08) |
| | 200 | 7.58E-01(1.51E-06)† | 3.47E-01(1.59E-06)† | 8.32E-01(1.80E-08)† | 8.32E-01(2.13E-08)† | 8.32E-01(1.27E-08) |
| WFG5 | 5 | 6.88E-01(2.13E-06)† | 5.27E-01(1.56E-05)† | 7.63E-01(6.74E-06)† | 7.89E-01(5.69E-07)‡ | 7.88E-01(6.30E-07) |
| | 25 | 7.12E-01(5.97E-06)† | 5.36E-01(1.25E-06)† | 7.94E-01(1.82E-07)† | 7.95E-01(2.70E-07) | 7.95E-01(3.05E-07) |
| | 50 | 7.17E-01(4.61E-07)† | 5.33E-01(2.59E-06)† | 7.96E-01(1.86E-07)† | 7.97E-01(2.04E-07)† | 7.97E-01(1.66E-07) |
| | 100 | 7.21E-01(4.86E-07)† | 5.30E-01(1.97E-06)† | 7.98E-01(1.08E-07)† | 7.98E-01(7.20E-08)† | 7.98E-01(8.90E-08) |
| | 200 | 7.26E-01(1.01E-06)† | 5.30E-01(1.87E-06)† | 7.98E-01(5.51E-08)† | 7.99E-01(1.93E-08) | 7.99E-01(4.55E-08) |
| WFG6 | 5 | 6.90E-01(1.95E-05)† | 1.45E-01(4.91E-06)† | 7.33E-01(4.86E-05)† | 7.67E-01(1.75E-04)† | 7.73E-01(7.90E-05) |
| | 25 | 7.28E-01(5.03E-06)† | 1.48E-01(7.37E-06)† | 7.90E-01(6.02E-05) | 7.81E-01(1.93E-04)† | 7.91E-01(7.28E-05) |
| | 50 | 7.34E-01(3.98E-06)† | 1.49E-01(7.96E-06)† | 7.96E-01(5.64E-05) | 7.84E-01(1.94E-04)† | 7.94E-01(6.84E-05) |
| | 100 | 7.43E-01(4.11E-06)† | 1.49E-01(8.45E-06)† | 8.00E-01(4.93E-05)‡ | 7.87E-01(1.97E-04)† | 7.97E-01(7.37E-05) |
| | 200 | 7.48E-01(2.16E-06)† | 1.49E-01(8.61E-06)† | 8.05E-01(4.48E-05)‡ | 7.91E-01(1.95E-04)† | 8.01E-01(6.85E-05) |
| WFG7 | 5 | 7.14E-01(4.08E-06)† | 7.04E-01(2.28E-05)† | 7.98E-01(5.17E-06)† | 8.17E-01(9.30E-07)† | 8.18E-01(6.84E-07) |
| | 25 | 7.63E-01(2.00E-06)† | 7.54E-01(1.75E-05)† | 8.30E-01(6.61E-08)† | 8.31E-01(5.08E-08)† | 8.31E-01(3.97E-08) |
| | 50 | 7.72E-01(2.36E-07)† | 7.63E-01(1.46E-05)† | 8.32E-01(2.91E-08)† | 8.32E-01(3.31E-08)† | 8.32E-01(2.52E-08) |
| | 100 | 7.77E-01(1.37E-06)† | 7.70E-01(9.19E-06)† | 8.33E-01(1.38E-08)† | 8.33E-01(1.18E-08) | 8.33E-01(7.54E-09) |
| | 200 | 7.83E-01(5.28E-07)† | 7.73E-01(6.98E-06)† | 8.33E-01(1.42E-08)† | 8.33E-01(8.56E-09) | 8.33E-01(1.94E-08) |
| WFG8 | 5 | 7.21E-01(1.56E-06)† | 7.05E-01(3.72E-05)† | 7.98E-01(9.27E-06)† | 8.17E-01(8.24E-07)† | 8.18E-01(6.31E-07) |
| | 25 | 7.64E-01(1.60E-06)† | 7.56E-01(1.42E-05)† | 8.30E-01(7.98E-08)† | 8.31E-01(7.59E-08)† | 8.31E-01(8.27E-08) |
| | 50 | 7.72E-01(8.54E-07)† | 7.64E-01(1.52E-05)† | 8.32E-01(2.79E-08)† | 8.32E-01(2.03E-08) | 8.32E-01(2.32E-08) |
| | 100 | 7.79E-01(9.95E-08)† | 7.71E-01(9.79E-06)† | 8.33E-01(1.22E-08)† | 8.33E-01(1.84E-08) | 8.33E-01(1.86E-08) |
| | 200 | 7.81E-01(1.65E-07)† | 7.73E-01(3.93E-06)† | 8.33E-01(9.98E-09)† | 8.33E-01(9.13E-09) | 8.33E-01(1.24E-08) |
| WFG9 | 5 | 6.02E-01(1.27E-04)‡ | 6.22E-01(9.27E-03)‡ | 5.22E-01(1.77E-03) | 4.68E-01(2.60E-04) | 5.44E-01(2.57E-03) |
| | 25 | 6.34E-01(1.22E-04)‡ | 6.85E-01(7.29E-03)‡ | 5.40E-01(2.15E-03) | 4.73E-01(3.12E-04) | 5.55E-01(3.00E-03) |
| | 50 | 6.42E-01(3.06E-05)‡ | 6.99E-01(6.63E-03)‡ | 5.42E-01(2.18E-03) | 4.76E-01(3.15E-04) | 5.57E-01(3.08E-03) |
| | 100 | 6.68E-01(6.45E-05)‡ | 7.04E-01(6.00E-03)‡ | 5.43E-01(2.17E-03) | 4.81E-01(3.10E-04) | 5.59E-01(3.03E-03) |
| | 200 | 6.88E-01(1.11E-04)‡ | 7.05E-01(5.22E-03)‡ | 5.45E-01(2.16E-03) | 4.85E-01(2.80E-04) | 5.63E-01(2.87E-03) |
| +-= | | 55/10/0 | 57/5/3 | 42/5/18 | 42/1/22 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.15: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 1.22E+02(1.14E+00)† | 1.12E+03(5.52E+02)† | 6.97E+00(1.01E+00)† | 7.78E+00(1.28E+00)† | 6.21E+00(1.26E+00) |
| | 25 | 1.17E+02(2.43E+00)† | 1.27E+03(1.25E+03)† | 1.51E+00(2.29E-01)† | 2.14E+00(1.85E-01)† | 6.44E-01(1.06E-01) |
| | 50 | 1.16E+02(6.07E-01)† | 1.31E+03(8.76E+02)† | 3.95E-01(2.78E-02)† | 6.73E-01(5.63E-02)† | 1.24E-01(6.73E-03) |
| | 100 | 1.15E+02(2.70E+00)† | 1.35E+03(9.10E+02)† | 1.56E-01(7.46E-03)† | 2.94E-01(8.40E-03)† | 5.16E-02(1.33E-03) |
| | 200 | 1.24E+02(8.19E-01)† | 1.37E+03(7.42E+02)† | 6.88E-02(2.54E-03)† | 1.43E-01(4.81E-03)† | 4.58E-02(1.72E-03) |
| F2 | 5 | 3.21E-01(5.74E-06)† | 8.63E+02(4.02E+02)† | 6.40E-02(5.17E-06)† | 5.50E-02(3.49E-06)‡ | 5.97E-02(2.50E-06) |
| | 25 | 3.26E-01(2.19E-05)† | 9.33E+02(1.50E+02)† | 4.14E-02(9.21E-07)† | 3.88E-02(1.24E-06)‡ | 4.00E-02(5.95E-07) |
| | 50 | 3.25E-01(5.87E-06)† | 9.41E+02(1.16E+02)† | 3.53E-02(3.08E-07)† | 3.46E-02(4.25E-07)‡ | 3.49E-02(4.01E-07) |
| | 100 | 3.43E-01(3.87E-05)† | 9.51E+02(1.38E+02)† | 3.23E-02(1.93E-07) | 3.24E-02(1.40E-07) | 3.24E-02(1.24E-07) |
| | 200 | 3.54E-01(3.30E-05)† | 9.65E+02(9.23E+01)† | 3.16E-02(1.37E-07)† | 3.15E-02(1.35E-07) | 3.15E-02(7.63E-08) |
| F3 | 5 | 2.56E+02(1.67E+01)† | 1.01E+03(7.96E+02)† | 1.73E+01(3.23E+00)† | 1.96E+01(5.73E+00)† | 1.46E+01(5.31E+00) |
| | 25 | 2.40E+02(1.85E+01)† | 9.98E+02(7.75E+02)† | 5.18E+00(1.71E+00)† | 5.79E+00(1.77E+00)† | 2.14E+00(6.94E-01) |
| | 50 | 2.37E+02(1.97E+01)† | 9.84E+02(2.62E+02)† | 1.27E+00(3.76E-01)† | 1.85E+00(8.32E-01)† | 3.59E-01(6.70E-02) |
| | 100 | 2.48E+02(4.58E+01)† | 9.65E+02(1.74E+02)† | 4.16E-01(5.11E-02)† | 7.49E-01(9.39E-02)† | 1.27E-01(7.20E-03) |
| | 200 | 2.61E+02(1.97E+01)† | 9.67E+02(1.41E+02)† | 1.63E-01(1.22E-02)† | 4.25E-01(3.26E-02)† | 1.25E-01(1.28E-02) |
| F4 | 5 | 4.87E-01(2.31E-05)† | 9.63E-02(2.70E-04)† | 7.29E-02(4.16E-05)† | 5.87E-02(2.22E-04)‡ | 6.26E-02(3.99E-05) |
| | 25 | 4.40E-01(2.06E-05)† | 4.26E-02(1.80E-05)† | 3.73E-02(2.26E-06)† | 2.54E-02(1.21E-05)‡ | 3.41E-02(3.15E-06) |
| | 50 | 4.51E-01(4.00E-06)† | 3.93E-02(1.45E-05)† | 3.21E-02(2.18E-07)† | 2.34E-02(1.64E-05)‡ | 3.03E-02(1.29E-06) |
| | 100 | 4.75E-01(3.24E-05)† | 3.99E-02(1.46E-05)† | 3.14E-02(1.30E-07)† | 2.33E-02(1.63E-05)‡ | 3.06E-02(1.26E-06) |
| | 200 | 4.76E-01(1.30E-05)† | 4.13E-02(1.22E-05)† | 3.33E-02(6.85E-08)† | 2.33E-02(1.70E-05)‡ | 3.22E-02(2.39E-06) |
| WFG1 | 5 | 3.21E-01(3.28E-06)† | 1.05E+03(5.31E+02)† | 2.21E-01(1.32E-05)† | 2.20E-01(1.64E-05)† | 2.18E-01(1.14E-05) |
| | 25 | 2.98E-01(4.27E-06)† | 1.11E+03(4.41E+02)† | 2.10E-01(1.79E-05)† | 1.97E-01(2.85E-05)† | 1.87E-01(2.31E-05) |
| | 50 | 2.86E-01(2.23E-06)† | 1.13E+03(3.38E+02)† | 1.88E-01(2.01E-05)† | 1.80E-01(1.96E-05)† | 1.78E-01(1.40E-05) |
| | 100 | 2.78E-01(9.34E-06)† | 1.15E+03(1.85E+02)† | 1.76E-01(1.24E-05)† | 1.74E-01(1.37E-05)† | 1.73E-01(1.13E-05) |
| | 200 | 2.79E-01(1.64E-06)† | 1.17E+03(1.37E+02)† | 1.68E-01(1.08E-05) | 1.68E-01(6.79E-06) | 1.69E-01(1.38E-05) |
| WFG2 | 5 | 7.31E-01(2.27E-05)† | 1.32E+00(8.47E-05)† | 4.88E-01(5.03E-05)‡ | 5.05E-01(1.27E-04) | 4.98E-01(1.10E-04) |
| | 25 | 6.86E-01(1.24E-05)† | 1.31E+00(1.39E-05)† | 4.48E-01(4.65E-05)‡ | 4.60E-01(4.79E-05) | 4.60E-01(4.34E-05) |
| | 50 | 6.87E-01(8.30E-05)† | 1.31E+00(8.10E-06)† | 4.54E-01(4.00E-05)‡ | 4.60E-01(3.55E-05) | 4.59E-01(4.00E-05) |
| | 100 | 7.02E-01(2.48E-05)† | 1.31E+00(7.86E-06)† | 4.52E-01(4.52E-05)‡ | 4.55E-01(5.00E-05) | 4.56E-01(3.46E-05) |
| | 200 | 7.12E-01(5.87E-06)† | 1.31E+00(4.79E-06)† | 4.48E-01(3.01E-05)† | 4.48E-01(3.05E-05)† | 4.45E-01(2.75E-05) |
| WFG3 | 5 | 1.80E+00(1.26E-04)† | 1.46E+00(2.29E-05)‡ | 1.57E+00(1.22E-03)† | 1.44E+00(3.05E-03)‡ | 1.55E+00(4.32E-03) |
| | 25 | 1.96E+00(1.59E-04)† | 1.46E+00(1.69E-05)‡ | 1.65E+00(6.08E-04)† | 1.51E+00(1.95E-03)‡ | 1.55E+00(2.29E-03) |
| | 50 | 2.00E+00(5.03E-05)† | 1.46E+00(1.30E-05)‡ | 1.69E+00(4.14E-04)† | 1.57E+00(1.56E-03) | 1.59E+00(1.46E-03) |
| | 100 | 1.98E+00(5.02E-05)† | 1.46E+00(4.59E-06)‡ | 1.71E+00(1.76E-04)† | 1.62E+00(1.21E-03) | 1.62E+00(1.35E-03) |
| | 200 | 2.03E+00(1.16E-05)† | 1.46E+00(4.75E-06)‡ | 1.71E+00(2.15E-04)† | 1.65E+00(4.90E-04) | 1.65E+00(6.14E-04) |
| WFG4 | 5 | 5.07E-01(2.78E-06)† | 1.00E+00(3.54E-05)† | 2.43E-01(1.12E-05)‡ | 2.59E-01(1.30E-05)† | 2.56E-01(1.52E-05) |
| | 25 | 4.96E-01(3.92E-07)† | 9.39E-01(2.56E-05)† | 2.16E-01(1.22E-05)‡ | 2.19E-01(6.09E-06) | 2.20E-01(6.99E-06) |
| | 50 | 5.00E-01(6.61E-06)† | 9.15E-01(1.63E-05)† | 2.08E-01(8.63E-06)‡ | 2.11E-01(4.18E-06) | 2.11E-01(5.50E-06) |
| | 100 | 5.00E-01(1.07E-05)† | 9.03E-01(1.33E-05)† | 2.08E-01(6.18E-06)† | 2.08E-01(4.01E-06)† | 2.07E-01(5.52E-06) |
| | 200 | 4.74E-01(5.68E-06)† | 9.03E-01(4.22E-06)† | 2.14E-01(3.55E-06)† | 2.09E-01(5.26E-06) | 2.09E-01(7.02E-06) |
| WFG5 | 5 | 5.45E-01(1.14E-05)† | 2.52E-01(2.13E-05)‡ | 2.98E-01(1.29E-05)† | 2.86E-01(1.40E-05) | 2.89E-01(1.24E-05) |
| | 25 | 5.15E-01(4.84E-06)† | 2.13E-01(5.98E-05)‡ | 2.61E-01(4.50E-06)† | 2.57E-01(4.91E-06)† | 2.56E-01(9.14E-06) |
| | 50 | 5.01E-01(1.87E-06)† | 1.84E-01(9.30E-05)‡ | 2.52E-01(4.23E-06)† | 2.51E-01(5.25E-06)† | 2.50E-01(7.09E-06) |
| | 100 | 4.92E-01(9.28E-06)† | 1.88E-01(3.35E-05)‡ | 2.47E-01(4.97E-06)† | 2.47E-01(9.20E-06)† | 2.46E-01(3.64E-06) |
| | 200 | 4.82E-01(5.44E-06)† | 1.95E-01(2.72E-05)‡ | 2.47E-01(4.22E-06) | 2.47E-01(3.84E-06) | 2.47E-01(4.90E-06) |
| WFG6 | 5 | 5.72E-01(1.24E-05)† | 9.10E-01(4.75E-04)† | 3.56E-01(1.83E-04)† | 3.57E-01(4.34E-04)† | 3.51E-01(1.71E-04) |
| | 25 | 5.08E-01(7.50E-06)† | 8.88E-01(6.82E-04)† | 2.82E-01(1.92E-04)† | 2.87E-01(5.10E-04)† | 2.74E-01(1.89E-04) |
| | 50 | 5.04E-01(1.06E-05)† | 8.86E-01(8.12E-04)† | 2.64E-01(2.08E-04)‡ | 2.79E-01(5.28E-04)† | 2.66E-01(1.89E-04) |
| | 100 | 4.87E-01(1.13E-05)† | 8.84E-01(8.27E-04)† | 2.57E-01(1.85E-04)‡ | 2.74E-01(5.01E-04)† | 2.63E-01(1.98E-04) |
| | 200 | 4.95E-01(1.44E-05)† | 8.83E-01(7.92E-04)† | 2.49E-01(1.51E-04)‡ | 2.68E-01(5.11E-04)† | 2.57E-01(2.11E-04) |
| WFG7 | 5 | 5.69E-01(5.47E-06)† | 2.75E-01(4.71E-05)† | 2.59E-01(8.16E-06)† | 2.56E-01(1.29E-05)† | 2.51E-01(2.02E-05) |
| | 25 | 5.23E-01(4.13E-07)† | 1.98E-01(2.73E-05)‡ | 2.07E-01(4.57E-06)‡ | 2.11E-01(8.32E-06)† | 2.10E-01(5.01E-06) |
| | 50 | 5.20E-01(5.65E-06)† | 1.87E-01(4.00E-05)‡ | 2.04E-01(4.49E-06) | 2.06E-01(7.55E-06)† | 2.05E-01(6.00E-06) |
| | 100 | 5.17E-01(1.60E-05)† | 1.90E-01(4.30E-05)‡ | 2.06E-01(6.97E-06) | 2.10E-01(7.54E-06) | 2.10E-01(6.83E-06) |
| | 200 | 4.96E-01(5.56E-06)† | 2.06E-01(3.87E-05)‡ | 2.23E-01(6.08E-06) | 2.24E-01(6.22E-06) | 2.23E-01(7.74E-06) |
| WFG8 | 5 | 5.66E-01(3.00E-07)† | 2.75E-01(6.10E-05)† | 2.61E-01(1.69E-05)† | 2.53E-01(8.65E-06)† | 2.51E-01(1.31E-05) |
| | 25 | 5.24E-01(1.54E-06)† | 1.96E-01(3.26E-05)‡ | 2.07E-01(5.44E-06)‡ | 2.09E-01(7.61E-06) | 2.11E-01(6.16E-06) |
| | 50 | 5.22E-01(6.15E-06)† | 1.85E-01(4.75E-05)‡ | 2.03E-01(3.75E-06)‡ | 2.07E-01(5.80E-06) | 2.07E-01(6.27E-06) |
| | 100 | 5.17E-01(1.82E-06)† | 1.89E-01(4.44E-05)‡ | 2.07E-01(6.56E-06)‡ | 2.11E-01(6.04E-06) | 2.11E-01(1.02E-05) |
| | 200 | 5.09E-01(1.51E-06)† | 2.06E-01(2.34E-05)‡ | 2.22E-01(6.65E-06)‡ | 2.24E-01(6.37E-06) | 2.24E-01(9.74E-06) |
| WFG9 | 5 | 8.28E-01(2.92E-04) | 5.35E-01(4.35E-02)‡ | 8.36E-01(6.39E-03)† | 8.97E-01(2.94E-04)† | 7.61E-01(9.05E-03) |
| | 25 | 7.78E-01(3.43E-04) | 3.56E-01(2.86E-02)‡ | 8.02E-01(8.87E-03)† | 8.84E-01(3.23E-04)† | 7.18E-01(1.10E-02) |
| | 50 | 7.78E-01(6.56E-05) | 3.11E-01(2.48E-02)‡ | 7.91E-01(8.79E-03)† | 8.81E-01(2.92E-04)† | 7.13E-01(1.08E-02) |
| | 100 | 7.57E-01(7.18E-05) | 2.87E-01(2.24E-02)‡ | 7.85E-01(8.61E-03)† | 8.79E-01(2.29E-04)† | 7.09E-01(1.05E-02) |
| | 200 | 7.12E-01(3.28E-04) | 2.73E-01(2.06E-02)‡ | 7.79E-01(8.42E-03)† | 8.79E-01(1.71E-04)† | 7.11E-01(1.05E-02) |
| +-= | | 60/0/5 | 42/23/0 | 44/15/6 | 34/10/21 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.16: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 9.93E-01(7.91E-07)† | 1.74E-02(8.64E-04)† | 9.94E-01(9.29E-08)† | 9.99E-01(1.92E-10) | 9.99E-01(2.53E-10) |
| | 25 | 9.77E-01(1.58E-03)† | 1.85E-01(2.79E-03)† | 9.99E-01(1.56E-07)† | 9.99E-01(1.04E-10) | 9.99E-01(2.59E-10) |
| | 50 | 8.85E-01(4.37E-03)† | 2.38E-01(3.16E-03)† | 9.99E-01(5.94E-09)† | 9.99E-01(8.62E-09) | 9.99E-01(4.79E-09) |
| F2 | 5 | 9.54E-01(6.59E-08)† | 7.46E-01(1.19E-03)† | 9.64E-01(3.62E-06)† | 9.77E-01(5.36E-08) | 9.77E-01(1.18E-06) |
| | 25 | 9.55E-01(4.91E-07)† | 7.22E-01(1.38E-03)† | 9.78E-01(2.86E-09)† | 9.78E-01(3.38E-09) | 9.78E-01(1.78E-07) |
| | 50 | 9.44E-01(1.77E-06)† | 7.17E-01(1.36E-03)† | 9.78E-01(2.30E-09)† | 9.78E-01(6.49E-10) | 9.78E-01(7.86E-10) |
| F3 | 5 | 9.08E-01(2.34E-05)† | 2.03E-05(1.28E-08)† | 9.30E-01(8.33E-06)† | 9.77E-01(1.98E-07) | 9.77E-01(3.06E-07) |
| | 25 | 9.27E-01(2.67E-04)† | 4.24E-02(1.48E-03)† | 9.73E-01(3.09E-06)† | 9.77E-01(4.27E-07) | 9.78E-01(6.96E-07) |
| | 50 | 8.47E-01(4.48E-03)† | 1.71E-01(5.92E-04)† | 9.76E-01(1.16E-06)† | 9.77E-01(4.66E-07) | 9.77E-01(2.11E-06) |
| F4 | 5 | 9.25E-01(2.66E-06)† | 9.28E-01(4.66E-05)† | 9.54E-01(2.21E-05)† | 8.69E-01(2.03E-03)† | 9.63E-01(3.48E-04) |
| | 25 | 9.16E-01(1.66E-05)† | 9.44E-01(4.04E-05)† | 9.78E-01(5.58E-08) | 8.66E-01(2.04E-03)† | 9.65E-01(3.69E-04) |
| | 50 | 9.17E-01(2.26E-05)† | 9.56E-01(1.32E-04)† | 9.78E-01(3.06E-09) | 8.66E-01(2.03E-03)† | 9.65E-01(3.70E-04) |
| WFG1 | 5 | 9.65E-01(5.63E-06)† | 2.96E-01(8.92E-06)† | 9.62E-01(5.73E-06)† | 9.87E-01(6.99E-06) | 9.87E-01(2.06E-06) |
| | 25 | 9.82E-01(3.25E-06)† | 3.03E-01(8.95E-06)† | 9.83E-01(4.07E-06)† | 9.89E-01(5.01E-06) | 9.89E-01(1.65E-06) |
| | 50 | 9.84E-01(2.59E-06)† | 3.03E-01(9.71E-06)† | 9.88E-01(2.36E-06)† | 9.89E-01(3.97E-06) | 9.89E-01(1.31E-06) |
| WFG2 | 5 | 9.81E-01(3.71E-07)† | 2.24E-01(4.18E-09)† | 9.83E-01(2.01E-06)† | 9.91E-01(5.50E-09) | 9.91E-01(5.46E-09) |
| | 25 | 9.88E-01(9.99E-09)† | 2.24E-01(2.24E-09)† | 9.91E-01(3.23E-09) | 9.92E-01(3.09E-09)‡ | 9.91E-01(4.85E-09) |
| | 50 | 9.88E-01(5.92E-09)† | 2.25E-01(2.38E-09)† | 9.92E-01(7.18E-09)‡ | 9.91E-01(5.53E-09) | 9.91E-01(5.64E-09) |
| WFG3 | 5 | 7.09E-01(4.86E-07)‡ | 2.34E-01(4.57E-08)† | 7.01E-01(2.85E-06)‡ | 6.98E-01(6.02E-06) | 6.98E-01(9.67E-06) |
| | 25 | 7.10E-01(2.15E-07)‡ | 2.35E-01(9.78E-09)† | 6.98E-01(2.06E-06)‡ | 6.94E-01(4.67E-06) | 6.95E-01(4.89E-06) |
| | 50 | 7.13E-01(2.64E-09)‡ | 2.35E-01(9.53E-09)† | 6.96E-01(2.21E-06)‡ | 6.93E-01(5.65E-06) | 6.93E-01(8.72E-06) |
| WFG4 | 5 | 7.15E-01(1.76E-06)† | 3.00E-01(7.41E-05)† | 7.86E-01(3.04E-05)† | 8.61E-01(3.88E-07) | 8.62E-01(2.58E-07) |
| | 25 | 7.41E-01(4.83E-06)† | 3.44E-01(2.17E-05)† | 8.65E-01(2.81E-06)† | 8.68E-01(1.02E-07) | 8.68E-01(9.40E-08) |
| | 50 | 7.48E-01(2.69E-06)† | 3.43E-01(3.99E-05)† | 8.69E-01(4.53E-08)† | 8.69E-01(7.08E-08) | 8.69E-01(4.89E-08) |
| WFG5 | 5 | 6.95E-01(1.17E-05)† | 5.02E-01(3.00E-05)† | 7.60E-01(2.68E-05)† | 8.12E-01(1.44E-06)‡ | 8.11E-01(3.13E-06) |
| | 25 | 7.16E-01(4.18E-06)† | 5.24E-01(4.52E-06)† | 8.16E-01(1.18E-06)† | 8.20E-01(1.39E-06) | 8.19E-01(1.48E-06) |
| | 50 | 7.23E-01(4.03E-06)† | 5.20E-01(3.87E-06)† | 8.21E-01(2.08E-06)† | 8.24E-01(7.62E-07)‡ | 8.23E-01(1.29E-06) |
| WFG6 | 5 | 6.76E-01(7.05E-06)† | 1.26E-01(3.11E-06)† | 7.19E-01(5.42E-04)† | 7.83E-01(8.37E-04) | 7.91E-01(5.41E-04) |
| | 25 | 7.26E-01(1.41E-05)† | 1.27E-01(5.29E-06)† | 7.99E-01(6.13E-04) | 7.89E-01(8.27E-04) | 7.96E-01(5.48E-04) |
| | 50 | 7.41E-01(3.98E-06)† | 1.27E-01(5.92E-06)† | 8.05E-01(5.82E-04) | 7.90E-01(8.06E-04) | 7.97E-01(5.49E-04) |
| WFG7 | 5 | 7.61E-01(1.14E-05)† | 6.91E-01(1.28E-04)† | 8.24E-01(3.22E-05)† | 8.60E-01(1.07E-06) | 8.60E-01(1.48E-06) |
| | 25 | 7.84E-01(6.85E-06)† | 7.53E-01(7.58E-05)† | 8.68E-01(4.41E-08)† | 8.69E-01(4.74E-08) | 8.69E-01(3.30E-08) |
| | 50 | 8.00E-01(1.92E-06)† | 7.67E-01(4.55E-05)† | 8.69E-01(7.72E-08) | 8.69E-01(4.20E-08) | 8.69E-01(5.35E-08) |
| WFG8 | 5 | 7.50E-01(2.00E-06)† | 6.85E-01(1.00E-04)† | 8.23E-01(2.45E-05)† | 8.60E-01(9.94E-07) | 8.60E-01(1.51E-06) |
| | 25 | 7.88E-01(8.97E-07)† | 7.54E-01(3.79E-05)† | 8.68E-01(8.96E-08)† | 8.69E-01(6.14E-08) | 8.69E-01(7.39E-08) |
| | 50 | 7.98E-01(3.13E-07)† | 7.68E-01(3.61E-05)† | 8.69E-01(4.17E-08)† | 8.69E-01(2.64E-08) | 8.69E-01(3.48E-08) |
| WFG9 | 5 | 4.37E-01(7.41E-06)† | 5.97E-01(1.91E-02)‡ | 4.68E-01(4.86E-05)‡ | 4.38E-01(9.85E-04) | 4.44E-01(7.11E-04) |
| | 25 | 5.01E-01(3.21E-05)‡ | 6.74E-01(1.32E-02)‡ | 4.90E-01(1.04E-04)‡ | 4.45E-01(1.10E-03) | 4.52E-01(8.68E-04) |
| | 50 | 5.21E-01(6.29E-06)‡ | 6.98E-01(1.16E-02)‡ | 4.93E-01(8.49E-05)‡ | 4.48E-01(1.13E-03) | 4.58E-01(9.85E-04) |
| +-= | | 34/5/0 | 36/3/0 | 26/7/6 | 3/3/33 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.17: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 4.21E-01(1.93E-01)† | 1.07E+03(2.17E+03)† | 3.68E-02(1.28E-03)† | 2.02E-02(2.18E-04) | 4.25E-02(1.32E-02) |
| | 25 | 3.46E+00(1.18E+01)† | 1.36E+03(4.30E+03)† | 4.63E-02(9.34E-04)† | 3.05E-02(1.64E-03) | 2.18E-02(6.45E-04) |
| | 50 | 2.19E+01(9.94E+01)† | 1.45E+03(2.42E+03)† | 4.95E-02(1.74E-03)† | 2.11E-02(3.82E-04) | 2.12E-02(2.32E-04) |
| F2 | 5 | 2.09E-01(1.43E-04)† | 1.02E+03(2.49E+03)† | 5.64E-02(4.89E-06)† | 4.71E-02(2.81E-07) | 4.68E-02(4.38E-07) |
| | 25 | 2.84E-01(5.17E-05)† | 1.11E+03(5.06E+02)† | 4.42E-02(8.87E-07)† | 4.12E-02(3.42E-07) | 4.12E-02(4.58E-07) |
| | 50 | 3.54E-01(6.55E-05)† | 1.12E+03(3.10E+02)† | 4.17E-02(8.07E-07)† | 4.05E-02(2.42E-07) | 4.03E-02(4.13E-07) |
| F3 | 5 | 2.56E-01(3.82E-02)† | 1.01E+03(2.76E+03)† | 9.07E-02(5.91E-03) | 1.63E-01(3.36E-02) | 9.87E-02(1.33E-02) |
| | 25 | 1.19E+00(8.18E-01)† | 1.14E+03(3.93E+03)† | 1.35E-01(1.12E-02) | 1.46E-01(3.31E-02) | 1.82E-01(6.16E-02) |
| | 50 | 1.26E+01(1.35E+02)† | 1.15E+03(5.56E+02)† | 1.35E-01(1.45E-02)† | 1.26E-01(1.24E-02) | 1.34E-01(5.49E-02) |
| F4 | 5 | 2.76E-01(4.32E-05)† | 4.79E-02(4.07E-04)† | 4.19E-02(1.89E-06)† | 2.28E-02(8.24E-06)‡ | 3.96E-02(9.29E-06) |
| | 25 | 3.65E-01(1.56E-04)† | 4.26E-02(5.71E-05)† | 4.08E-02(8.51E-07)† | 2.69E-02(2.22E-05)‡ | 3.81E-02(3.69E-06) |
| | 50 | 4.53E-01(8.02E-05)† | 3.83E-02(1.44E-05) | 3.86E-02(2.31E-07)† | 2.68E-02(2.81E-05)‡ | 3.69E-02(2.75E-06) |
| WFG1 | 5 | 3.01E-01(2.43E-05)† | 1.13E+03(4.00E+03)† | 2.51E-01(4.35E-05)† | 2.32E-01(5.13E-05) | 2.33E-01(4.01E-05) |
| | 25 | 3.15E-01(3.10E-05)† | 1.23E+03(2.74E+03)† | 2.52E-01(5.95E-05)† | 2.18E-01(4.72E-05) | 2.20E-01(4.04E-05) |
| | 50 | 3.26E-01(1.55E-05)† | 1.28E+03(1.73E+03)† | 2.34E-01(4.74E-05)† | 2.14E-01(6.36E-05) | 2.14E-01(4.03E-05) |
| WFG2 | 5 | 8.89E-01(1.95E-05)† | 1.42E+00(7.99E-05)† | 6.35E-01(1.32E-04)‡ | 6.96E-01(3.31E-04) | 7.05E-01(3.84E-04) |
| | 25 | 9.26E-01(4.81E-04)† | 1.42E+00(4.47E-05)† | 5.80E-01(1.77E-04)‡ | 6.09E-01(1.44E-04) | 6.14E-01(1.94E-04) |
| | 50 | 9.38E-01(2.31E-04)† | 1.42E+00(2.27E-05)† | 5.83E-01(1.39E-04)‡ | 6.08E-01(1.86E-04) | 6.07E-01(1.46E-04) |
| WFG3 | 5 | 1.70E+00(1.02E-03)† | 1.61E+00(1.88E-05)† | 1.55E+00(1.89E-03) | 1.51E+00(1.27E-02) | 1.51E+00(9.13E-03) |
| | 25 | 2.24E+00(5.81E-04)† | 1.62E+00(3.25E-05)‡ | 1.89E+00(1.75E-03)† | 1.69E+00(8.24E-03) | 1.68E+00(6.82E-03) |
| | 50 | 2.34E+00(2.00E-04)† | 1.62E+00(1.77E-05)‡ | 2.00E+00(9.73E-04)† | 1.81E+00(7.68E-03) | 1.79E+00(7.73E-03) |
| WFG4 | 5 | 6.11E-01(2.12E-05)† | 1.15E+00(1.00E-04)† | 3.13E-01(3.37E-05)‡ | 3.35E-01(3.67E-05) | 3.35E-01(3.52E-05) |
| | 25 | 6.50E-01(6.47E-05)† | 1.06E+00(8.79E-05)† | 2.87E-01(3.24E-05)† | 2.86E-01(1.75E-05) | 2.84E-01(2.34E-05) |
| | 50 | 6.62E-01(2.89E-05)‡ | 1.03E+00(1.01E-04)‡ | 2.72E-01(2.27E-05) | 2.74E-01(2.15E-05) | 2.73E-01(1.54E-05) |
| WFG5 | 5 | 6.55E-01(1.09E-05)† | 3.15E-01(1.23E-04)‡ | 3.86E-01(4.76E-05)† | 3.66E-01(7.53E-05) | 3.68E-01(8.01E-05) |
| | 25 | 6.57E-01(1.46E-05)† | 2.72E-01(3.60E-04)‡ | 3.43E-01(2.68E-05)† | 3.33E-01(2.74E-05) | 3.33E-01(2.39E-05) |
| | 50 | 6.68E-01(1.30E-05)† | 2.36E-01(1.19E-04)‡ | 3.28E-01(1.28E-05)† | 3.22E-01(2.40E-05) | 3.24E-01(2.19E-05) |
| WFG6 | 5 | 6.63E-01(1.73E-06)† | 1.06E+00(4.30E-04)† | 4.08E-01(1.85E-03) | 4.24E-01(2.14E-03) | 4.12E-01(1.17E-03) |
| | 25 | 6.79E-01(9.47E-06)† | 1.06E+00(8.79E-04)† | 3.81E-01(1.90E-03) | 3.86E-01(2.59E-03) | 3.73E-01(1.59E-03) |
| | 50 | 6.61E-01(1.78E-05)† | 1.05E+00(1.03E-03)† | 3.59E-01(2.00E-03) | 3.81E-01(2.80E-03) | 3.70E-01(1.70E-03) |
| WFG7 | 5 | 6.30E-01(3.91E-05)† | 2.38E-01(1.32E-04)‡ | 3.26E-01(3.09E-05)† | 3.10E-01(4.20E-05) | 3.10E-01(2.51E-05) |
| | 25 | 6.74E-01(2.89E-05)† | 1.95E-01(1.06E-04)‡ | 2.66E-01(2.32E-05) | 2.64E-01(1.78E-05) | 2.65E-01(2.31E-05) |
| | 50 | 6.73E-01(1.16E-05)† | 2.02E-01(5.65E-05)‡ | 2.57E-01(2.34E-05)‡ | 2.59E-01(1.60E-05) | 2.59E-01(2.24E-05) |
| WFG8 | 5 | 6.34E-01(1.45E-05)† | 2.36E-01(9.69E-05)‡ | 3.27E-01(4.31E-05)† | 3.13E-01(2.75E-05) | 3.13E-01(4.32E-05) |
| | 25 | 6.66E-01(2.06E-07)† | 1.95E-01(8.68E-05)‡ | 2.66E-01(1.66E-05) | 2.63E-01(1.68E-05) | 2.63E-01(2.31E-05) |
| | 50 | 6.59E-01(4.42E-06)† | 2.06E-01(9.29E-05)‡ | 2.56E-01(1.64E-05)‡ | 2.58E-01(1.68E-05) | 2.58E-01(1.11E-05) |
| WFG9 | 5 | 1.23E+00(1.66E-05)† | 6.33E-01(8.60E-02)‡ | 1.05E+00(1.01E-03) | 1.02E+00(2.53E-03) | 1.03E+00(1.64E-03) |
| | 25 | 1.25E+00(7.71E-06)† | 4.68E-01(5.05E-02)‡ | 1.06E+00(7.72E-04)† | 1.02E+00(2.11E-03) | 1.02E+00(1.27E-03) |
| | 50 | 1.21E+00(5.21E-07)† | 4.08E-01(4.09E-02)‡ | 1.04E+00(7.04E-04)† | 1.01E+00(1.70E-03) | 1.02E+00(1.03E-03) |
| +-= | | 38/1/0 | 23/15/1 | 23/6/10 | 0/3/33 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.18: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t$ = 100), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 8.05E-01(1.91E-05)† | 2.09E-02(1.15E-03)† | 9.94E-01(1.16E-07)† | 9.99E-01(2.06E-09) | 9.99E-01(5.36E-10) |
| | 25 | 7.97E-01(3.46E-05)† | 2.15E-01(4.00E-03)† | 9.99E-01(1.34E-07)† | 9.99E-01(6.24E-10) | 9.99E-01(3.44E-10) |
| | 50 | 7.94E-01(4.47E-04)† | 2.57E-01(4.41E-03)† | 9.99E-01(3.05E-08)† | 9.99E-01(8.04E-10) | 9.99E-01(1.63E-08) |
| | 100 | 6.11E-01(2.49E-04)† | 2.62E-01(5.76E-03)† | 9.99E-01(3.45E-08)† | 9.99E-01(1.57E-08) | 9.99E-01(1.56E-09) |
| F2 | 5 | 9.49E-01(1.46E-06)† | 7.82E-01(1.20E-04)† | 9.63E-01(2.62E-06)† | 9.77E-01(1.61E-07) | 9.77E-01(2.00E-08) |
| | 25 | 9.48E-01(8.80E-06)† | 7.62E-01(9.51E-05)† | 9.78E-01(1.76E-09)† | 9.78E-01(4.28E-08) | 9.78E-01(2.04E-09) |
| | 50 | 9.39E-01(1.23E-05)† | 7.54E-01(7.51E-05)† | 9.78E-01(1.11E-09)† | 9.78E-01(7.23E-10) | 9.78E-01(6.93E-10) |
| | 100 | 9.30E-01(1.33E-05)† | 7.54E-01(7.47E-05)† | 9.78E-01(6.84E-10)† | 9.78E-01(3.68E-10) | 9.78E-01(3.32E-10) |
| F3 | 5 | 7.97E-01(4.88E-03)† | 3.72E-04(4.29E-06)† | 9.31E-01(1.43E-05)† | 9.77E-01(9.61E-07) | 9.76E-01(2.62E-05) |
| | 25 | 7.71E-01(1.57E-03)† | 4.72E-02(1.18E-03)† | 9.73E-01(7.73E-06)† | 9.77E-01(8.02E-07) | 9.77E-01(5.67E-06) |
| | 50 | 7.61E-01(2.29E-03)† | 1.65E-01(3.48E-04)† | 9.76E-01(2.70E-06)† | 9.77E-01(5.78E-07) | 9.77E-01(1.21E-06) |
| | 100 | 6.37E-01(5.57E-03)† | 1.84E-01(1.33E-07)† | 9.76E-01(2.17E-06)† | 9.77E-01(1.37E-06) | 9.77E-01(7.71E-07) |
| F4 | 5 | 9.27E-01(7.51E-06)† | 9.31E-01(4.05E-05)† | 9.53E-01(2.12E-05)† | 8.95E-01(1.21E-03)† | 9.61E-01(6.97E-04) |
| | 25 | 9.22E-01(2.22E-05)† | 9.44E-01(6.24E-05)† | 9.78E-01(1.83E-08)‡ | 8.93E-01(1.30E-03)† | 9.64E-01(7.36E-04) |
| | 50 | 9.00E-01(7.70E-06)† | 9.57E-01(1.34E-04)† | 9.78E-01(1.75E-09)‡ | 8.93E-01(1.31E-03)† | 9.64E-01(7.38E-04) |
| | 100 | 8.75E-01(6.86E-06)† | 9.66E-01(1.20E-04)‡ | 9.78E-01(3.42E-10)‡ | 8.93E-01(1.33E-03)† | 9.64E-01(7.37E-04) |
| WFG1 | 5 | 9.57E-01(1.10E-06)† | 2.97E-01(9.89E-06)† | 9.64E-01(1.23E-05)† | 9.88E-01(8.62E-07) | 9.87E-01(3.12E-06) |
| | 25 | 9.78E-01(9.84E-07)† | 3.04E-01(7.55E-06)† | 9.85E-01(3.08E-06)† | 9.89E-01(7.89E-07) | 9.89E-01(2.86E-06) |
| | 50 | 9.83E-01(4.30E-07)† | 3.04E-01(1.24E-05)† | 9.89E-01(1.76E-06) | 9.89E-01(7.27E-07) | 9.89E-01(2.28E-06) |
| | 100 | 9.85E-01(3.17E-07)† | 3.04E-01(1.01E-05)† | 9.90E-01(9.10E-07)‡ | 9.89E-01(7.02E-07) | 9.89E-01(2.07E-06) |
| WFG2 | 5 | 9.81E-01(1.75E-07)† | 2.24E-01(2.55E-09)† | 9.82E-01(1.69E-06)† | 9.91E-01(4.96E-09) | 9.91E-01(1.38E-08) |
| | 25 | 9.89E-01(2.96E-09)† | 2.24E-01(2.39E-09)† | 9.91E-01(4.74E-09)† | 9.92E-01(4.58E-09) | 9.91E-01(3.50E-09) |
| | 50 | 9.89E-01(3.37E-09)† | 2.25E-01(1.84E-09)† | 9.92E-01(3.36E-09)‡ | 9.92E-01(5.51E-09) | 9.91E-01(6.01E-09) |
| | 100 | 9.89E-01(2.39E-09)† | 2.25E-01(2.82E-09)† | 9.92E-01(3.65E-09) | 9.92E-01(7.61E-09) | 9.92E-01(1.06E-08) |
| WFG3 | 5 | 7.10E-01(1.34E-06)‡ | 2.34E-01(2.99E-08)† | 6.99E-01(7.63E-06) | 6.99E-01(6.61E-06) | 6.99E-01(7.88E-06) |
| | 25 | 7.12E-01(9.89E-08)‡ | 2.35E-01(6.17E-09)† | 6.96E-01(3.18E-06)‡ | 6.95E-01(5.37E-06) | 6.95E-01(4.06E-06) |
| | 50 | 7.12E-01(2.58E-07)‡ | 2.35E-01(6.17E-09)† | 6.95E-01(3.28E-06)‡ | 6.93E-01(4.30E-06) | 6.93E-01(3.86E-06) |
| | 100 | 7.11E-01(5.42E-07)‡ | 2.35E-01(3.43E-09)† | 6.95E-01(3.67E-06)‡ | 6.92E-01(9.57E-06) | 6.92E-01(3.92E-06) |
| WFG4 | 5 | 7.28E-01(5.68E-06)† | 2.95E-01(6.27E-05)† | 7.90E-01(5.14E-05)† | 8.62E-01(2.77E-07) | 8.62E-01(2.82E-07) |
| | 25 | 7.42E-01(2.97E-06)† | 3.42E-01(1.66E-05)† | 8.65E-01(3.60E-07)† | 8.68E-01(4.19E-08) | 8.68E-01(6.71E-08) |
| | 50 | 7.43E-01(9.00E-06)† | 3.43E-01(1.40E-05)† | 8.68E-01(5.12E-08)† | 8.69E-01(5.56E-08) | 8.69E-01(3.56E-08) |
| | 100 | 7.55E-01(8.49E-06)† | 3.40E-01(1.51E-05)† | 8.69E-01(3.86E-08)† | 8.70E-01(3.26E-08) | 8.70E-01(4.17E-08) |
| WFG5 | 5 | 7.06E-01(2.97E-06)† | 5.04E-01(2.44E-05)† | 7.62E-01(2.56E-05)† | 8.13E-01(2.08E-06) | 8.13E-01(7.64E-07) |
| | 25 | 7.13E-01(3.98E-06)† | 5.24E-01(4.43E-06)† | 8.17E-01(5.35E-07)† | 8.20E-01(1.17E-06) | 8.21E-01(1.31E-06) |
| | 50 | 7.18E-01(1.43E-07)† | 5.20E-01(5.82E-06)† | 8.22E-01(7.31E-07)† | 8.24E-01(6.90E-07) | 8.24E-01(6.95E-07) |
| | 100 | 7.25E-01(2.79E-06)† | 5.18E-01(3.59E-06)† | 8.25E-01(3.05E-07)† | 8.25E-01(1.59E-07) | 8.25E-01(1.96E-07) |
| WFG6 | 5 | 6.67E-01(7.55E-06)† | 1.26E-01(1.62E-06)† | 6.99E-01(1.21E-03)† | 7.93E-01(8.43E-04) | 7.90E-01(6.35E-04) |
| | 25 | 7.07E-01(8.87E-06)† | 1.26E-01(4.01E-06)† | 7.77E-01(1.53E-03)† | 7.97E-01(8.63E-04) | 7.95E-01(6.29E-04) |
| | 50 | 7.27E-01(1.30E-06)† | 1.27E-01(3.56E-06)† | 7.84E-01(1.46E-03) | 7.99E-01(8.50E-04) | 7.97E-01(6.06E-04) |
| | 100 | 7.40E-01(1.13E-05)† | 1.27E-01(3.26E-06)† | 7.93E-01(1.46E-03) | 8.02E-01(7.66E-04) | 8.00E-01(5.64E-04) |
| WFG7 | 5 | 7.49E-01(5.12E-06)† | 7.08E-01(1.17E-04)† | 8.26E-01(1.09E-05)† | 8.60E-01(1.92E-06) | 8.60E-01(1.21E-06) |
| | 25 | 7.89E-01(4.87E-06)† | 7.67E-01(5.22E-05)† | 8.68E-01(3.66E-08)† | 8.69E-01(6.38E-08) | 8.69E-01(4.16E-08) |
| | 50 | 7.96E-01(3.27E-07)† | 7.78E-01(2.80E-05)† | 8.69E-01(5.26E-08)† | 8.69E-01(2.76E-08) | 8.69E-01(3.45E-08) |
| | 100 | 8.03E-01(3.67E-07)† | 7.90E-01(3.94E-05)† | 8.69E-01(1.99E-08)† | 8.70E-01(2.89E-08) | 8.70E-01(1.65E-08) |
| WFG8 | 5 | 7.58E-01(1.63E-05)† | 7.08E-01(1.09E-04)† | 8.25E-01(1.22E-05)† | 8.60E-01(1.49E-06) | 8.60E-01(1.58E-06) |
| | 25 | 7.89E-01(3.71E-07)† | 7.70E-01(2.48E-05)† | 8.68E-01(5.61E-08)† | 8.69E-01(4.31E-08) | 8.69E-01(3.57E-08) |
| | 50 | 7.97E-01(2.01E-06)† | 7.80E-01(2.89E-05)† | 8.69E-01(4.73E-08)† | 8.69E-01(3.09E-08) | 8.69E-01(2.01E-08) |
| | 100 | 8.04E-01(9.83E-07)† | 7.90E-01(4.50E-05)† | 8.69E-01(3.72E-08)† | 8.69E-01(1.79E-08) | 8.69E-01(2.58E-08) |
| WFG9 | 5 | 4.47E-01(1.34E-04) | 6.24E-01(1.99E-02)‡ | 4.71E-01(1.16E-04)‡ | 4.45E-01(1.02E-03) | 4.44E-01(8.23E-04) |
| | 25 | 5.06E-01(1.13E-04)‡ | 6.87E-01(1.18E-02)‡ | 4.93E-01(1.07E-04)‡ | 4.52E-01(1.19E-03) | 4.51E-01(1.10E-03) |
| | 50 | 5.28E-01(2.11E-04)‡ | 7.09E-01(8.89E-03)‡ | 4.97E-01(1.12E-04)‡ | 4.55E-01(1.26E-03) | 4.54E-01(1.25E-03) |
| | 100 | 5.37E-01(2.36E-04)‡ | 7.18E-01(7.73E-03)‡ | 5.03E-01(3.64E-04)‡ | 4.63E-01(1.11E-03) | 4.62E-01(1.19E-03) |
| +-= | | 44/7/1 | 47/5/0 | 35/12/5 | 4/0/48 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.19: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t$ = 100), averaged across the cases of increasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 1.28E+01(1.34E+01)† | 1.09E+03(1.13E+03)† | 5.34E-02(2.33E-03)† | 4.54E-02(1.18E-02) | 3.80E-02(5.30E-03) |
| | 25 | 2.71E+01(9.16E+00)† | 1.35E+03(4.44E+03)† | 5.11E-02(3.04E-03)† | 3.22E-02(1.85E-03) | 2.12E-02(5.26E-04) |
| | 50 | 4.33E+01(1.68E+00)† | 1.45E+03(1.51E+03)† | 7.71E-02(7.15E-03)† | 2.68E-02(1.51E-03) | 2.98E-02(2.50E-03) |
| | 100 | 6.55E+01(1.77E+01)† | 1.51E+03(1.72E+03)† | 7.62E-02(5.08E-03)† | 3.07E-02(1.86E-03) | 2.51E-02(6.10E-04) |
| F2 | 5 | 2.09E-01(1.52E-05)† | 9.67E+02(5.09E+03)† | 5.54E-02(3.93E-06)† | 4.74E-02(4.43E-07) | 4.73E-02(6.95E-07) |
| | 25 | 2.76E-01(6.25E-05)† | 1.10E+03(1.26E+03)† | 4.28E-02(1.17E-06)† | 4.09E-02(2.50E-07) | 4.11E-02(4.29E-07) |
| | 50 | 3.41E-01(1.72E-05)† | 1.12E+03(6.19E+02)† | 4.04E-02(2.60E-07) | 4.01E-02(2.95E-07)‡ | 4.04E-02(4.40E-07) |
| | 100 | 3.95E-01(3.34E-05)† | 1.12E+03(5.98E+02)† | 3.93E-02(2.07E-07)‡ | 3.97E-02(2.07E-07) | 3.97E-02(3.16E-07) |
| F3 | 5 | 1.95E+01(3.77E+02)† | 1.02E+03(1.79E+03)† | 1.07E-01(5.33E-03) | 2.37E-01(1.65E-01) | 1.57E-01(5.57E-02) |
| | 25 | 4.25E+01(7.70E+02)† | 1.14E+03(2.62E+03)† | 1.26E-01(1.66E-02) | 9.49E-02(4.98E-03) | 1.28E-01(1.90E-02) |
| | 50 | 7.83E+01(6.18E+02)† | 1.15E+03(1.17E+02)† | 1.60E-01(2.49E-02)† | 1.31E-01(1.25E-02) | 1.19E-01(1.24E-02) |
| | 100 | 1.39E+02(8.43E+02)† | 1.14E+03(8.66E+01)† | 1.76E-01(1.30E-02)† | 1.23E-01(2.09E-02) | 7.90E-02(9.20E-03) |
| F4 | 5 | 3.16E-01(3.61E-06)† | 4.02E-02(1.22E-05) | 4.00E-02(2.36E-06) | 2.53E-02(8.83E-06)‡ | 3.89E-02(1.69E-05) |
| | 25 | 4.13E-01(3.88E-06)† | 3.97E-02(2.95E-05) | 4.00E-02(7.25E-07)† | 2.99E-02(1.35E-05)‡ | 3.76E-02(5.16E-06) |
| | 50 | 4.72E-01(6.07E-06)† | 3.99E-02(8.70E-06)† | 3.85E-02(3.20E-07)† | 3.02E-02(1.76E-05)‡ | 3.69E-02(4.30E-06) |
| | 100 | 5.19E-01(1.26E-06)† | 4.13E-02(3.91E-06)† | 3.86E-02(2.26E-07) | 2.99E-02(1.86E-05)‡ | 3.77E-02(5.13E-06) |
| WFG1 | 5 | 3.20E-01(3.99E-06)† | 1.14E+03(1.72E+03)† | 2.39E-01(4.01E-05)† | 2.30E-01(3.92E-05)‡ | 2.33E-01(3.22E-05) |
| | 25 | 3.34E-01(1.40E-05)† | 1.24E+03(1.16E+03)† | 2.38E-01(6.57E-05)† | 2.15E-01(3.28E-05) | 2.16E-01(2.08E-05) |
| | 50 | 3.31E-01(1.26E-05)† | 1.29E+03(1.12E+03)† | 2.25E-01(7.16E-05)† | 2.08E-01(3.03E-05) | 2.11E-01(3.19E-05) |
| | 100 | 3.38E-01(4.74E-06)† | 1.33E+03(6.25E+02)† | 2.12E-01(5.15E-05)† | 2.03E-01(2.88E-05) | 2.05E-01(3.41E-05) |
| WFG2 | 5 | 9.22E-01(3.13E-04)† | 1.42E+00(5.31E-05)† | 6.53E-01(3.23E-04)‡ | 6.96E-01(3.71E-04) | 6.93E-01(3.94E-04) |
| | 25 | 8.68E-01(1.51E-05)† | 1.42E+00(1.95E-05)† | 5.87E-01(1.43E-04)‡ | 6.07E-01(1.88E-04) | 6.09E-01(2.14E-04) |
| | 50 | 8.87E-01(1.44E-04)† | 1.42E+00(2.10E-05)† | 5.88E-01(2.14E-04)‡ | 6.01E-01(1.44E-04) | 6.06E-01(1.60E-04) |
| | 100 | 9.44E-01(1.73E-04)† | 1.42E+00(1.01E-05)† | 5.88E-01(1.37E-04)‡ | 5.94E-01(1.86E-04) | 6.01E-01(2.45E-04) |
| WFG3 | 5 | 1.90E+00(2.77E-03)† | 1.60E+00(1.33E-05)† | 1.59E+00(2.81E-03) | 1.55E+00(1.00E-02) | 1.56E+00(7.35E-03) |
| | 25 | 2.31E+00(1.87E-03)† | 1.62E+00(4.07E-05)‡ | 1.91E+00(1.97E-03)† | 1.70E+00(7.62E-03) | 1.72E+00(6.78E-03) |
| | 50 | 2.38E+00(6.68E-04)† | 1.62E+00(2.13E-05)‡ | 2.01E+00(1.13E-03)† | 1.81E+00(7.64E-03) | 1.82E+00(7.05E-03) |
| | 100 | 2.40E+00(1.98E-05)† | 1.62E+00(1.90E-05)‡ | 2.05E+00(6.54E-04)† | 1.90E+00(6.23E-03) | 1.92E+00(3.77E-03) |
| WFG4 | 5 | 5.85E-01(1.89E-05)† | 1.13E+00(6.44E-05)† | 3.10E-01(3.51E-05)‡ | 3.34E-01(3.03E-05) | 3.32E-01(2.45E-05) |
| | 25 | 6.34E-01(1.00E-06)† | 1.05E+00(6.46E-05)† | 2.80E-01(2.98E-05) | 2.82E-01(2.06E-05) | 2.82E-01(1.68E-05) |
| | 50 | 6.60E-01(1.45E-05)† | 1.03E+00(8.18E-05)† | 2.68E-01(1.79E-05)‡ | 2.73E-01(1.47E-05) | 2.72E-01(1.75E-05) |
| | 100 | 6.65E-01(5.36E-05)† | 1.01E+00(4.34E-05)† | 2.66E-01(1.30E-05) | 2.69E-01(1.19E-05) | 2.68E-01(2.19E-05) |
| WFG5 | 5 | 6.42E-01(1.55E-05)† | 3.14E-01(8.38E-05)‡ | 3.79E-01(4.77E-05)† | 3.62E-01(6.67E-05) | 3.63E-01(3.92E-05) |
| | 25 | 6.52E-01(8.84E-06)† | 2.72E-01(3.61E-04)‡ | 3.39E-01(1.83E-05)† | 3.31E-01(2.43E-05) | 3.31E-01(1.88E-05) |
| | 50 | 6.46E-01(7.87E-05)† | 2.40E-01(2.41E-04)‡ | 3.27E-01(1.03E-05)† | 3.24E-01(1.42E-05) | 3.23E-01(2.37E-05) |
| | 100 | 6.54E-01(1.37E-05)† | 2.43E-01(2.05E-04)‡ | 3.18E-01(1.49E-05) | 3.17E-01(1.29E-05) | 3.17E-01(1.69E-05) |
| WFG6 | 5 | 6.99E-01(5.95E-06)† | 1.06E+00(2.88E-04)† | 4.66E-01(8.03E-03)† | 4.12E-01(1.94E-03) | 4.15E-01(1.48E-03) |
| | 25 | 7.19E-01(5.44E-05)† | 1.06E+00(8.39E-04)† | 4.23E-01(6.03E-03)† | 3.73E-01(2.41E-03) | 3.76E-01(1.93E-03) |
| | 50 | 7.07E-01(8.23E-06)† | 1.06E+00(9.43E-04)† | 3.98E-01(5.93E-03) | 3.69E-01(2.59E-03) | 3.70E-01(2.08E-03) |
| | 100 | 6.93E-01(2.77E-05)† | 1.05E+00(8.11E-04)† | 3.83E-01(5.96E-03) | 3.65E-01(2.59E-03) | 3.67E-01(1.86E-03) |
| WFG7 | 5 | 6.47E-01(7.27E-06)† | 2.38E-01(7.94E-05)‡ | 3.15E-01(3.85E-05)† | 3.06E-01(6.21E-05)† | 3.02E-01(2.74E-05) |
| | 25 | 6.69E-01(1.15E-05)† | 2.23E-01(9.46E-05)‡ | 2.59E-01(1.37E-05)‡ | 2.63E-01(1.54E-05) | 2.61E-01(1.71E-05) |
| | 50 | 6.77E-01(1.15E-05)† | 2.26E-01(1.03E-04)‡ | 2.52E-01(1.30E-05)‡ | 2.60E-01(1.39E-05) | 2.58E-01(2.24E-05) |
| | 100 | 6.74E-01(4.46E-06)† | 2.47E-01(1.11E-04)‡ | 2.64E-01(2.04E-05)‡ | 2.71E-01(1.95E-05) | 2.71E-01(1.36E-05) |
| WFG8 | 5 | 6.48E-01(1.07E-05)† | 2.45E-01(1.02E-04)‡ | 3.15E-01(5.26E-05)† | 3.04E-01(4.61E-05) | 3.05E-01(4.12E-05) |
| | 25 | 6.61E-01(1.24E-06)† | 2.24E-01(1.23E-04)‡ | 2.59E-01(1.71E-05)‡ | 2.60E-01(1.55E-05)‡ | 2.62E-01(1.84E-05) |
| | 50 | 6.72E-01(3.36E-06)† | 2.28E-01(1.12E-04)‡ | 2.53E-01(8.86E-06)‡ | 2.57E-01(1.56E-05) | 2.57E-01(2.04E-05) |
| | 100 | 6.79E-01(1.16E-05)† | 2.47E-01(1.59E-04)‡ | 2.63E-01(1.57E-05)‡ | 2.71E-01(1.77E-05) | 2.71E-01(1.97E-05) |
| WFG9 | 5 | 1.23E+00(2.41E-04)† | 5.97E-01(9.25E-02)‡ | 1.05E+00(1.10E-03) | 1.02E+00(1.78E-03) | 1.03E+00(1.90E-03) |
| | 25 | 1.23E+00(1.50E-04)† | 4.40E-01(5.01E-02)‡ | 1.05E+00(5.11E-04)† | 1.02E+00(1.70E-03) | 1.02E+00(1.77E-03) |
| | 50 | 1.21E+00(2.46E-04)† | 3.87E-01(3.51E-02)‡ | 1.04E+00(5.36E-04) | 1.02E+00(1.44E-03) | 1.02E+00(1.54E-03) |
| | 100 | 1.20E+00(5.55E-04)† | 3.57E-01(2.89E-02)‡ | 1.02E+00(1.78E-03) | 1.01E+00(9.51E-04) | 1.01E+00(9.68E-04) |
| +-= | | 52/0/0 | 31/19/2 | 25/13/14 | 1/7/44 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.20: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 5.94E-01(1.64E-08)† | 1.11E-02(4.23E-04)† | 9.94E-01(9.15E-08)† | 9.99E-01(5.58E-08) | 9.99E-01(1.19E-09) |
| | 25 | 5.96E-01(2.49E-09)† | 1.75E-01(2.45E-03)† | 9.99E-01(1.48E-08)† | 9.99E-01(2.32E-08) | 9.99E-01(5.77E-09) |
| | 50 | 5.99E-01(2.94E-10)† | 2.38E-01(3.06E-03)† | 9.99E-01(1.33E-08)† | 9.99E-01(4.81E-09) | 9.99E-01(1.52E-09) |
| | 100 | 5.98E-01(2.84E-08)† | 2.49E-01(4.43E-03)† | 9.99E-01(1.90E-08)† | 9.99E-01(1.44E-08) | 9.99E-01(2.24E-09) |
| | 200 | 3.99E-01(1.45E-10)† | 2.61E-01(5.99E-03)† | 9.99E-01(3.00E-09)† | 9.99E-01(1.52E-08) | 9.99E-01(1.74E-08) |
| F2 | 5 | 9.51E-01(1.20E-06)† | 7.86E-01(1.89E-05)† | 9.64E-01(2.90E-06)† | 9.77E-01(5.12E-08)† | 9.77E-01(6.14E-08) |
| | 25 | 9.49E-01(6.64E-07)† | 7.61E-01(4.88E-05)† | 9.78E-01(3.62E-09)† | 9.78E-01(1.29E-09) | 9.78E-01(4.56E-09) |
| | 50 | 9.43E-01(3.70E-06)† | 7.53E-01(6.41E-06)† | 9.78E-01(1.27E-09)† | 9.78E-01(6.46E-10) | 9.78E-01(6.20E-10) |
| | 100 | 9.36E-01(1.51E-05)† | 7.54E-01(6.39E-06)† | 9.78E-01(5.08E-10)† | 9.78E-01(3.06E-10) | 9.78E-01(1.70E-10) |
| | 200 | 9.31E-01(3.02E-05)† | 7.54E-01(6.40E-06)† | 9.78E-01(3.25E-10)† | 9.78E-01(2.53E-10) | 9.78E-01(7.56E-11) |
| F3 | 5 | 5.36E-01(7.56E-04)† | 0.00E+00(0.00E+00)† | 9.31E-01(8.02E-06)† | 9.76E-01(2.79E-05) | 9.77E-01(1.91E-07) |
| | 25 | 5.59E-01(7.50E-04)† | 4.27E-02(8.43E-04)† | 9.73E-01(3.88E-06)† | 9.77E-01(2.06E-06) | 9.77E-01(8.63E-07) |
| | 50 | 5.72E-01(4.52E-04)† | 1.64E-01(2.47E-04)† | 9.77E-01(6.53E-07)† | 9.77E-01(3.88E-07) | 9.77E-01(9.36E-07) |
| | 100 | 5.78E-01(2.91E-04)† | 1.84E-01(1.43E-07)† | 9.76E-01(1.41E-06)† | 9.78E-01(5.56E-07) | 9.78E-01(3.98E-07) |
| | 200 | 3.89E-01(1.83E-03)† | 1.85E-01(8.48E-08)† | 9.77E-01(1.40E-06)† | 9.77E-01(7.59E-07) | 9.77E-01(1.41E-06) |
| F4 | 5 | 9.08E-01(3.09E-05)† | 9.34E-01(2.21E-05)† | 9.53E-01(3.74E-05) | 8.74E-01(2.48E-03)† | 9.51E-01(6.82E-04) |
| | 25 | 8.99E-01(5.16E-05)† | 9.44E-01(9.62E-05) | 9.78E-01(2.00E-07) | 8.72E-01(2.56E-03)† | 9.54E-01(7.32E-04) |
| | 50 | 8.97E-01(5.53E-06)† | 9.53E-01(1.17E-04) | 9.78E-01(4.15E-08) | 8.72E-01(2.67E-03)† | 9.54E-01(7.41E-04) |
| | 100 | 8.67E-01(1.17E-05)† | 9.67E-01(9.95E-05) | 9.78E-01(1.92E-10) | 8.72E-01(2.68E-03)† | 9.54E-01(7.39E-04) |
| | 200 | 8.53E-01(1.05E-05)† | 9.72E-01(2.08E-05) | 9.78E-01(1.77E-10) | 8.72E-01(2.70E-03)† | 9.54E-01(7.44E-04) |
| WFG1 | 5 | 9.57E-01(3.70E-07)† | 2.97E-01(2.55E-06)† | 9.64E-01(4.41E-06)† | 9.87E-01(2.19E-06) | 9.88E-01(6.18E-07) |
| | 25 | 9.78E-01(7.84E-07)† | 3.05E-01(3.85E-06)† | 9.85E-01(1.38E-06)† | 9.89E-01(1.79E-06) | 9.89E-01(4.08E-07) |
| | 50 | 9.84E-01(2.64E-07)† | 3.05E-01(4.69E-06)† | 9.89E-01(3.43E-07) | 9.89E-01(1.48E-06) | 9.89E-01(4.07E-07) |
| | 100 | 9.86E-01(2.35E-07)† | 3.05E-01(2.86E-06)† | 9.90E-01(1.76E-07)‡ | 9.89E-01(9.39E-07) | 9.89E-01(3.58E-07) |
| | 200 | 9.87E-01(7.70E-08)† | 3.05E-01(2.82E-06)† | 9.90E-01(1.44E-07)‡ | 9.89E-01(7.28E-07) | 9.89E-01(2.64E-07) |
| WFG2 | 5 | 9.81E-01(2.10E-07)† | 2.24E-01(3.33E-09)† | 9.83E-01(2.21E-06)† | 9.91E-01(5.07E-09) | 9.91E-01(5.30E-09) |
| | 25 | 9.88E-01(3.13E-08)† | 2.24E-01(1.99E-09)† | 9.92E-01(4.34E-09) | 9.92E-01(4.76E-09) | 9.92E-01(2.29E-09) |
| | 50 | 9.89E-01(3.43E-10)† | 2.25E-01(1.29E-09)† | 9.92E-01(4.72E-09) | 9.92E-01(5.09E-09) | 9.92E-01(4.99E-09) |
| | 100 | 9.89E-01(2.77E-10)† | 2.25E-01(2.35E-09)† | 9.92E-01(3.61E-09) | 9.92E-01(1.19E-08) | 9.92E-01(1.22E-08) |
| | 200 | 9.89E-01(7.99E-10)† | 2.25E-01(2.23E-09)† | 9.92E-01(8.64E-09) | 9.92E-01(1.20E-08) | 9.92E-01(1.10E-08) |
| WFG3 | 5 | 7.10E-01(1.36E-07)‡ | 2.34E-01(3.40E-08)† | 7.01E-01(5.33E-06) | 6.99E-01(5.57E-06) | 6.99E-01(2.07E-05) |
| | 25 | 7.11E-01(2.11E-08)‡ | 2.35E-01(7.54E-09)† | 6.97E-01(2.77E-06)‡ | 6.95E-01(4.89E-06) | 6.94E-01(1.15E-05) |
| | 50 | 7.13E-01(1.87E-07)‡ | 2.35E-01(4.19E-09)† | 6.96E-01(2.39E-06)‡ | 6.94E-01(5.20E-06) | 6.93E-01(5.02E-06) |
| | 100 | 7.12E-01(1.48E-07)‡ | 2.35E-01(5.22E-09)† | 6.95E-01(3.47E-06)‡ | 6.92E-01(6.08E-06) | 6.91E-01(5.14E-06) |
| | 200 | 7.13E-01(2.96E-08)‡ | 2.36E-01(2.93E-09)† | 6.94E-01(4.45E-06)‡ | 6.91E-01(1.04E-05) | 6.91E-01(3.95E-06) |
| WFG4 | 5 | 7.22E-01(2.94E-06)† | 2.94E-01(5.76E-05)† | 7.88E-01(3.59E-05)† | 8.61E-01(3.30E-07) | 8.61E-01(2.86E-07) |
| | 25 | 7.43E-01(2.29E-06)† | 3.41E-01(2.43E-05)† | 8.64E-01(5.01E-07)† | 8.68E-01(1.14E-07) | 8.68E-01(6.94E-08) |
| | 50 | 7.46E-01(7.13E-06)† | 3.43E-01(1.78E-05)† | 8.68E-01(5.22E-08)† | 8.69E-01(4.53E-08) | 8.69E-01(4.41E-08) |
| | 100 | 7.47E-01(1.93E-05)† | 3.39E-01(1.27E-05)† | 8.68E-01(4.07E-08)† | 8.70E-01(3.60E-08) | 8.69E-01(1.85E-08) |
| | 200 | 7.80E-01(9.26E-06)† | 3.39E-01(2.65E-06)† | 8.69E-01(4.30E-08)† | 8.69E-01(5.73E-08) | 8.69E-01(4.72E-08) |
| WFG5 | 5 | 7.10E-01(2.35E-06)† | 5.08E-01(5.39E-05)† | 7.63E-01(2.56E-05)† | 8.13E-01(1.37E-06) | 8.13E-01(1.84E-06) |
| | 25 | 7.32E-01(1.19E-05)† | 5.24E-01(2.91E-06)† | 8.17E-01(7.56E-07)† | 8.21E-01(8.58E-07) | 8.21E-01(1.08E-06) |
| | 50 | 7.29E-01(3.41E-06)† | 5.20E-01(4.86E-06)† | 8.22E-01(7.61E-07)† | 8.24E-01(7.62E-07) | 8.24E-01(6.03E-07) |
| | 100 | 7.31E-01(2.71E-06)† | 5.18E-01(4.27E-06)† | 8.25E-01(3.82E-07)† | 8.25E-01(2.30E-07) | 8.26E-01(3.48E-07) |
| | 200 | 7.41E-01(4.24E-07)† | 5.17E-01(3.57E-06)† | 8.26E-01(1.45E-07)† | 8.26E-01(6.05E-08) | 8.26E-01(1.66E-07) |
| WFG6 | 5 | 6.95E-01(1.64E-05)† | 1.27E-01(6.66E-06)† | 7.12E-01(1.85E-04)† | 7.87E-01(2.89E-04) | 7.88E-01(3.06E-04) |
| | 25 | 7.29E-01(5.57E-06)† | 1.27E-01(7.10E-06)† | 7.93E-01(2.47E-04) | 7.92E-01(2.97E-04) | 7.93E-01(2.98E-04) |
| | 50 | 7.36E-01(5.40E-07)† | 1.28E-01(6.82E-06)† | 7.99E-01(2.28E-04) | 7.93E-01(2.76E-04) | 7.95E-01(2.76E-04) |
| | 100 | 7.44E-01(4.90E-06)† | 1.28E-01(7.36E-06)† | 8.05E-01(1.98E-04)‡ | 7.98E-01(2.78E-04) | 7.98E-01(2.95E-04) |
| | 200 | 7.55E-01(2.03E-06)† | 1.28E-01(6.33E-06)† | 8.14E-01(1.82E-04)‡ | 8.04E-01(2.85E-04) | 8.06E-01(2.76E-04) |
| WFG7 | 5 | 7.50E-01(3.18E-06)† | 7.23E-01(9.56E-05)† | 8.25E-01(1.84E-05)† | 8.60E-01(2.22E-06) | 8.60E-01(1.61E-06) |
| | 25 | 7.91E-01(3.69E-07)† | 7.80E-01(2.52E-05)† | 8.68E-01(6.85E-08)† | 8.69E-01(6.61E-08) | 8.69E-01(5.06E-08) |
| | 50 | 7.98E-01(1.02E-06)† | 7.92E-01(1.83E-05)† | 8.69E-01(2.86E-08)† | 8.69E-01(3.87E-08) | 8.69E-01(2.49E-08) |
| | 100 | 8.04E-01(2.18E-06)† | 8.00E-01(1.30E-05)† | 8.69E-01(1.80E-08)† | 8.70E-01(1.88E-08) | 8.70E-01(2.05E-08) |
| | 200 | 8.11E-01(3.00E-07)† | 8.01E-01(1.39E-05)† | 8.70E-01(6.91E-09)† | 8.70E-01(9.41E-09) | 8.70E-01(8.96E-09) |
| WFG8 | 5 | 7.61E-01(6.98E-06)† | 7.25E-01(1.21E-04)† | 8.25E-01(3.48E-05)† | 8.60E-01(1.74E-06) | 8.60E-01(1.04E-06) |
| | 25 | 7.93E-01(2.71E-06)† | 7.81E-01(3.61E-05)† | 8.68E-01(5.04E-08)† | 8.69E-01(5.19E-08) | 8.69E-01(6.57E-08) |
| | 50 | 8.02E-01(5.35E-06)† | 7.91E-01(2.93E-05)† | 8.69E-01(4.58E-08)† | 8.69E-01(3.53E-08) | 8.69E-01(2.31E-08) |
| | 100 | 8.06E-01(2.85E-07)† | 8.00E-01(1.55E-05)† | 8.69E-01(3.56E-08)† | 8.70E-01(1.44E-08) | 8.70E-01(2.05E-08) |
| | 200 | 8.06E-01(1.37E-06)† | 8.01E-01(1.42E-05)† | 8.70E-01(1.17E-08)† | 8.70E-01(6.35E-09) | 8.70E-01(6.26E-09) |
| WFG9 | 5 | 5.16E-01(3.77E-04)‡ | 5.99E-01(1.93E-02)‡ | 4.72E-01(4.70E-05)‡ | 4.46E-01(8.13E-04) | 4.49E-01(7.02E-04) |
| | 25 | 5.27E-01(3.66E-04)‡ | 6.74E-01(1.27E-02)‡ | 4.93E-01(4.69E-05)‡ | 4.51E-01(9.85E-04) | 4.53E-01(8.91E-04) |
| | 50 | 5.34E-01(8.06E-05)‡ | 6.95E-01(1.12E-02)‡ | 4.95E-01(4.46E-05)‡ | 4.56E-01(9.86E-04) | 4.56E-01(9.70E-04) |
| | 100 | 5.80E-01(2.09E-04)‡ | 7.03E-01(9.73E-03)‡ | 4.97E-01(3.87E-05)‡ | 4.65E-01(9.89E-04) | 4.60E-01(8.75E-04) |
| | 200 | 6.21E-01(4.33E-04)‡ | 7.07E-01(8.03E-03)‡ | 5.00E-01(4.10E-05)‡ | 4.73E-01(8.99E-04) | 4.68E-01(6.09E-04) |
| +-= | | 55/5/5 | 56/5/4 | 39/13/13 | 6/0/59 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.21: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of increasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 6.40E+01(2.38E+00)† | 1.10E+03(1.68E+03)† | 4.65E-02(2.41E-03)† | 6.00E-02(4.76E-02) | 3.22E-02(5.21E-03) |
| | 25 | 7.71E+01(1.45E-01)† | 1.39E+03(4.36E+03)† | 4.40E-02(8.95E-04)† | 3.60E-02(2.62E-03) | 2.01E-02(3.12E-04) |
| | 50 | 9.37E+01(6.60E-01)† | 1.46E+03(2.51E+03)† | 6.74E-02(3.03E-03)† | 3.04E-02(2.49E-03) | 1.94E-02(1.73E-04) |
| | 100 | 1.08E+02(1.76E+00)† | 1.51E+03(2.27E+03)† | 9.87E-02(1.36E-02)† | 2.30E-02(9.40E-04) | 2.32E-02(3.29E-04) |
| | 200 | 1.37E+02(2.61E+00)† | 1.55E+03(1.95E+03)† | 5.10E-02(9.61E-04)† | 2.61E-02(7.56E-04) | 3.04E-02(9.71E-04) |
| F2 | 5 | 2.23E-01(1.98E-05)† | 9.67E+02(1.55E+03)† | 5.50E-02(4.74E-06)† | 4.80E-02(5.83E-07) | 4.78E-02(4.05E-07) |
| | 25 | 3.02E-01(2.52E-06)† | 1.11E+03(3.74E+02)† | 4.17E-02(4.57E-07)† | 4.07E-02(3.55E-07) | 4.05E-02(4.27E-07) |
| | 50 | 3.46E-01(2.37E-05)† | 1.12E+03(2.06E+02)† | 3.95E-02(2.21E-07) | 3.98E-02(3.02E-07) | 3.98E-02(2.62E-07) |
| | 100 | 3.85E-01(8.51E-05)† | 1.13E+03(2.89E+02)† | 3.88E-02(2.09E-07)‡ | 3.94E-02(1.82E-07) | 3.93E-02(1.38E-07) |
| | 200 | 4.19E-01(6.25E-05)† | 1.15E+03(2.64E+02)† | 3.88E-02(2.48E-07) | 3.88E-02(4.49E-07) | 3.88E-02(1.91E-07) |
| F3 | 5 | 1.16E+02(3.50E+01)† | 1.00E+03(2.83E+03)† | 1.19E-01(1.40E-02) | 1.93E-01(1.85E-01) | 1.05E-01(1.36E-02) |
| | 25 | 1.61E+02(7.37E+01)† | 1.13E+03(2.96E+03)† | 9.67E-02(4.24E-03) | 1.54E-01(3.56E-02) | 9.95E-02(9.07E-03) |
| | 50 | 1.86E+02(1.09E+02)† | 1.14E+03(8.50E+02)† | 1.16E-01(8.32E-03)† | 1.16E-01(2.01E-02) | 9.61E-02(9.61E-03) |
| | 100 | 2.32E+02(2.40E+02)† | 1.13E+03(5.14E+02)† | 2.37E-01(5.29E-02)† | 1.01E-01(7.17E-03) | 9.60E-02(6.99E-03) |
| | 200 | 2.71E+02(5.99E+01)† | 1.14E+03(3.39E+02)† | 1.19E-01(1.28E-02) | 8.77E-02(6.79E-03) | 1.42E-01(4.23E-02) |
| F4 | 5 | 3.48E-01(2.68E-05)† | 3.79E-02(4.49E-06) | 3.96E-02(2.40E-06)† | 2.48E-02(1.92E-05)‡ | 3.63E-02(1.99E-05) |
| | 25 | 4.17E-01(2.11E-05)† | 3.90E-02(2.00E-05)† | 3.97E-02(1.09E-06)† | 2.76E-02(2.52E-05)‡ | 3.64E-02(6.28E-06) |
| | 50 | 4.89E-01(1.16E-05)† | 3.92E-02(1.17E-05)† | 3.84E-02(2.27E-07)† | 2.78E-02(3.57E-05)‡ | 3.56E-02(5.30E-06) |
| | 100 | 5.55E-01(6.59E-05)† | 4.26E-02(1.43E-05)† | 3.87E-02(3.17E-07)† | 2.78E-02(4.02E-05)‡ | 3.74E-02(5.29E-06) |
| | 200 | 5.74E-01(4.78E-05)† | 4.49E-02(1.57E-06)† | 4.19E-02(1.99E-07)† | 2.77E-02(4.16E-05)‡ | 3.96E-02(9.45E-06) |
| WFG1 | 5 | 3.30E-01(1.40E-06)† | 1.14E+03(1.88E+03)† | 2.31E-01(3.45E-05) | 2.32E-01(4.75E-05) | 2.31E-01(3.18E-05) |
| | 25 | 3.33E-01(6.02E-06)† | 1.25E+03(9.85E+02)† | 2.32E-01(2.61E-05)† | 2.16E-01(2.45E-05) | 2.15E-01(4.16E-05) |
| | 50 | 3.33E-01(6.12E-06)† | 1.29E+03(7.45E+02)† | 2.23E-01(3.89E-05)† | 2.10E-01(3.61E-05) | 2.10E-01(3.27E-05) |
| | 100 | 3.34E-01(1.06E-05)† | 1.33E+03(5.33E+02)† | 2.10E-01(2.72E-05)† | 2.04E-01(4.04E-05) | 2.04E-01(2.48E-05) |
| | 200 | 3.37E-01(3.02E-06)† | 1.35E+03(4.74E+02)† | 1.99E-01(2.64E-05)‡ | 2.02E-01(1.88E-05) | 2.03E-01(3.28E-05) |
| WFG2 | 5 | 8.76E-01(3.52E-05)† | 1.42E+00(3.35E-05)† | 6.43E-01(1.08E-04)‡ | 6.84E-01(3.50E-04) | 6.81E-01(3.97E-04) |
| | 25 | 8.97E-01(3.86E-05)† | 1.42E+00(9.70E-06)† | 5.77E-01(8.48E-05)‡ | 6.01E-01(1.14E-04) | 6.03E-01(1.07E-04) |
| | 50 | 8.74E-01(1.08E-04)† | 1.42E+00(2.87E-05)† | 5.85E-01(1.08E-04)‡ | 5.96E-01(1.24E-04) | 5.98E-01(1.08E-04) |
| | 100 | 8.99E-01(5.02E-05)† | 1.42E+00(2.37E-05)† | 5.82E-01(6.66E-05)‡ | 5.88E-01(2.06E-04) | 5.90E-01(6.33E-05) |
| | 200 | 9.23E-01(6.55E-05)† | 1.42E+00(1.10E-05)† | 5.79E-01(3.15E-05) | 5.80E-01(7.70E-05) | 5.80E-01(2.79E-05) |
| WFG3 | 5 | 1.83E+00(6.17E-05)† | 1.60E+00(1.57E-05) | 1.56E+00(2.85E-03) | 1.53E+00(8.63E-03) | 1.56E+00(1.13E-02) |
| | 25 | 2.35E+00(3.45E-04)† | 1.61E+00(3.94E-05)‡ | 1.91E+00(1.67E-03)† | 1.72E+00(4.77E-03) | 1.73E+00(8.85E-03) |
| | 50 | 2.42E+00(1.27E-04)† | 1.62E+00(3.22E-05)‡ | 2.01E+00(1.47E-03)† | 1.84E+00(4.48E-03) | 1.84E+00(6.14E-03) |
| | 100 | 2.41E+00(1.41E-04)† | 1.62E+00(1.11E-05)‡ | 2.06E+00(4.14E-04)† | 1.93E+00(4.25E-03) | 1.92E+00(4.93E-03) |
| | 200 | 2.46E+00(4.76E-05)† | 1.62E+00(1.10E-05)‡ | 2.06E+00(3.31E-04)† | 1.98E+00(1.59E-03) | 1.98E+00(1.67E-03) |
| WFG4 | 5 | 5.86E-01(7.98E-06)† | 1.12E+00(5.01E-05)† | 3.00E-01(3.60E-05)‡ | 3.35E-01(4.35E-05) | 3.36E-01(4.51E-05) |
| | 25 | 6.29E-01(2.58E-05)† | 1.05E+00(4.30E-05)† | 2.76E-01(3.71E-05)‡ | 2.83E-01(2.25E-05) | 2.85E-01(2.19E-05) |
| | 50 | 6.54E-01(2.55E-05)† | 1.02E+00(3.51E-05)† | 2.63E-01(2.24E-05)‡ | 2.71E-01(1.33E-05) | 2.71E-01(1.49E-05) |
| | 100 | 6.74E-01(2.75E-05)† | 1.00E+00(3.56E-05)† | 2.66E-01(2.00E-05) | 2.69E-01(1.43E-05) | 2.67E-01(1.67E-05) |
| | 200 | 6.29E-01(1.36E-05)† | 1.00E+00(1.31E-05)† | 2.77E-01(1.27E-05)† | 2.69E-01(1.81E-05) | 2.69E-01(2.02E-05) |
| WFG5 | 5 | 6.46E-01(3.37E-06)† | 3.17E-01(6.39E-05)‡ | 3.78E-01(4.00E-05)† | 3.63E-01(4.79E-05) | 3.63E-01(4.69E-05) |
| | 25 | 6.39E-01(4.84E-05)† | 2.71E-01(2.64E-04)‡ | 3.38E-01(1.26E-05)† | 3.30E-01(1.51E-05) | 3.30E-01(2.26E-05) |
| | 50 | 6.49E-01(1.72E-06)† | 2.34E-01(2.78E-04)‡ | 3.26E-01(1.16E-05)† | 3.22E-01(1.39E-05) | 3.23E-01(2.57E-05) |
| | 100 | 6.42E-01(1.54E-05)† | 2.42E-01(6.53E-05)‡ | 3.18E-01(1.78E-05)† | 3.17E-01(2.33E-05) | 3.15E-01(1.51E-05) |
| | 200 | 6.32E-01(3.02E-06)† | 2.53E-01(8.91E-05)‡ | 3.18E-01(1.59E-05) | 3.18E-01(1.20E-05) | 3.18E-01(1.79E-05) |
| WFG6 | 5 | 6.64E-01(1.21E-05)† | 1.05E+00(1.03E-03)† | 4.14E-01(7.95E-04) | 4.20E-01(6.12E-04) | 4.18E-01(5.49E-04) |
| | 25 | 6.55E-01(2.70E-05)† | 1.04E+00(9.34E-04)† | 3.90E-01(8.24E-04) | 3.80E-01(8.89E-04) | 3.79E-01(7.38E-04) |
| | 50 | 6.80E-01(3.24E-05)† | 1.04E+00(1.08E-03)† | 3.68E-01(8.49E-04) | 3.75E-01(9.43E-04) | 3.73E-01(7.72E-04) |
| | 100 | 6.58E-01(1.67E-05)† | 1.04E+00(1.05E-03)† | 3.57E-01(7.27E-04)‡ | 3.70E-01(8.98E-04) | 3.69E-01(8.59E-04) |
| | 200 | 6.82E-01(5.82E-05)† | 1.03E+00(8.11E-04)† | 3.42E-01(5.75E-04)‡ | 3.59E-01(9.95E-04) | 3.57E-01(8.39E-04) |
| WFG7 | 5 | 6.40E-01(4.37E-07)† | 2.61E-01(1.14E-04)‡ | 3.00E-01(2.36E-05)‡ | 2.89E-01(2.63E-05) | 2.90E-01(5.01E-05) |
| | 25 | 6.68E-01(2.14E-06)† | 2.46E-01(5.44E-05)‡ | 2.53E-01(1.39E-05)‡ | 2.61E-01(2.28E-05) | 2.60E-01(1.46E-05) |
| | 50 | 6.75E-01(2.19E-05)† | 2.54E-01(6.30E-05) | 2.51E-01(1.66E-05)‡ | 2.58E-01(2.33E-05)‡ | 2.56E-01(2.02E-05) |
| | 100 | 6.80E-01(2.53E-05)† | 2.64E-01(6.13E-05)‡ | 2.63E-01(1.46E-05)‡ | 2.72E-01(2.01E-05) | 2.70E-01(2.03E-05) |
| | 200 | 6.43E-01(3.58E-05)† | 2.81E-01(8.05E-05)‡ | 2.95E-01(1.95E-05)‡ | 2.98E-01(1.38E-05) | 2.98E-01(1.88E-05) |
| WFG8 | 5 | 6.36E-01(2.95E-06)† | 2.62E-01(1.40E-04)‡ | 2.99E-01(5.45E-05)† | 2.87E-01(2.69E-05)‡ | 2.91E-01(5.11E-05) |
| | 25 | 6.59E-01(7.24E-07)† | 2.44E-01(8.76E-05)‡ | 2.53E-01(1.58E-05)‡ | 2.59E-01(2.67E-05)‡ | 2.62E-01(1.37E-05) |
| | 50 | 6.71E-01(8.53E-06)† | 2.53E-01(9.87E-05)‡ | 2.50E-01(1.29E-05)‡ | 2.59E-01(1.74E-05) | 2.59E-01(2.16E-05) |
| | 100 | 6.82E-01(7.94E-06)† | 2.62E-01(9.12E-05)‡ | 2.64E-01(1.81E-05)‡ | 2.72E-01(1.87E-05) | 2.73E-01(2.87E-05) |
| | 200 | 6.72E-01(6.73E-06)† | 2.79E-01(5.89E-05)‡ | 2.94E-01(2.00E-05)‡ | 3.00E-01(1.60E-05) | 3.00E-01(1.42E-05) |
| WFG9 | 5 | 1.13E+00(9.89E-04)† | 6.77E-01(9.41E-02)‡ | 1.04E+00(2.26E-04)† | 1.02E+00(9.75E-04) | 1.02E+00(1.65E-03) |
| | 25 | 1.17E+00(9.39E-04)† | 4.77E-01(5.01E-02)‡ | 1.04E+00(2.77E-04)† | 1.02E+00(9.84E-04) | 1.02E+00(1.61E-03) |
| | 50 | 1.20E+00(1.94E-04)† | 4.16E-01(4.15E-02)‡ | 1.03E+00(3.29E-04)† | 1.02E+00(8.73E-04) | 1.02E+00(1.47E-03) |
| | 100 | 1.16E+00(2.58E-04)† | 3.85E-01(3.73E-02)‡ | 1.03E+00(2.99E-04) | 1.02E+00(6.35E-04) | 1.01E+00(1.05E-03) |
| | 200 | 1.07E+00(1.41E-03)† | 3.69E-01(3.31E-02)‡ | 1.02E+00(3.13E-04) | 1.02E+00(4.31E-04) | 1.02E+00(7.34E-04) |
| +-= | | 65/0/0 | 39/23/3 | 30/20/15 | 0/8/57 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.22: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 0.00E+00(0.00E+00)† | 0.00E+00(0.00E+00)† | 6.53E-01(1.76E-02)† | 6.21E-01(2.40E-02)† | 8.25E-01(3.62E-03) |
| | 25 | 2.68E-03(1.59E-04)† | 0.00E+00(0.00E+00)† | 9.08E-01(4.67E-03)† | 8.92E-01(1.80E-02)† | 9.76E-01(4.23E-04) |
| | 50 | 6.09E-02(2.90E-03)† | 0.00E+00(0.00E+00)† | 9.68E-01(6.18E-04)† | 9.59E-01(6.56E-03)† | 9.90E-01(1.89E-05) |
| F2 | 5 | 8.86E-01(1.39E-06)† | 5.38E-01(2.97E-03)† | 9.36E-01(3.54E-08)† | 9.37E-01(2.14E-07)‡ | 9.36E-01(4.12E-07) |
| | 25 | 9.00E-01(4.54E-07)† | 5.96E-01(2.23E-03)† | 9.38E-01(6.71E-09) | 9.38E-01(1.91E-08) | 9.38E-01(1.83E-08) |
| | 50 | 9.02E-01(2.07E-07)† | 5.97E-01(2.22E-03)† | 9.39E-01(4.02E-09) | 9.39E-01(1.10E-08) | 9.39E-01(6.10E-09) |
| F3 | 5 | 1.82E-02(1.38E-03)† | 0.00E+00(0.00E+00)† | 1.35E-01(1.98E-02)† | 2.22E-01(4.87E-03)† | 4.45E-01(1.34E-02) |
| | 25 | 1.32E-03(2.63E-05)† | 0.00E+00(0.00E+00)† | 4.50E-01(6.20E-02)† | 4.30E-01(5.36E-02)† | 8.41E-01(1.82E-02) |
| | 50 | 4.25E-04(5.59E-06)† | 0.00E+00(0.00E+00)† | 6.54E-01(5.31E-02)† | 6.26E-01(5.30E-02)† | 9.07E-01(8.97E-03) |
| F4 | 5 | 8.36E-01(3.74E-06)† | 9.15E-01(6.89E-05)† | 9.29E-01(1.18E-05)‡ | 8.80E-01(1.94E-03)† | 9.28E-01(2.09E-04) |
| | 25 | 8.48E-01(2.65E-06)† | 9.33E-01(2.93E-07)† | 9.38E-01(4.14E-08)‡ | 8.92E-01(1.48E-03)† | 9.33E-01(1.87E-04) |
| | 50 | 8.51E-01(1.21E-06)† | 9.35E-01(1.13E-07)† | 9.39E-01(6.57E-09)‡ | 8.92E-01(1.51E-03)† | 9.33E-01(1.93E-04) |
| WFG1 | 5 | 8.99E-01(4.04E-06)† | 3.15E-01(2.18E-05)† | 8.72E-01(1.37E-05)† | 8.88E-01(1.69E-05)† | 9.00E-01(4.55E-05) |
| | 25 | 9.18E-01(1.40E-06)† | 3.17E-01(2.11E-05)† | 9.07E-01(3.13E-05)† | 9.18E-01(1.71E-05)† | 9.27E-01(1.36E-05) |
| | 50 | 9.31E-01(2.36E-06)† | 3.17E-01(3.48E-05)† | 9.31E-01(5.46E-06)† | 9.36E-01(5.35E-06)† | 9.39E-01(4.14E-06) |
| WFG2 | 5 | 9.18E-01(9.91E-07)† | 2.59E-01(7.02E-08)† | 9.33E-01(2.36E-06)† | 9.37E-01(9.23E-07)‡ | 9.34E-01(1.57E-06) |
| | 25 | 9.39E-01(3.40E-08)† | 2.59E-01(8.64E-09)† | 9.43E-01(7.12E-08)† | 9.43E-01(3.17E-08)† | 9.43E-01(5.22E-08) |
| | 50 | 9.40E-01(8.78E-09)† | 2.59E-01(4.62E-09)† | 9.44E-01(1.17E-08)† | 9.44E-01(5.32E-09)† | 9.44E-01(1.07E-08) |
| WFG3 | 5 | 6.66E-01(6.71E-07) | 2.70E-01(4.87E-07)† | 6.19E-01(6.06E-05)† | 6.47E-01(1.09E-04)† | 6.63E-01(1.64E-04) |
| | 25 | 7.07E-01(9.53E-07)‡ | 2.72E-01(4.27E-08)† | 6.74E-01(1.80E-05)† | 6.89E-01(3.98E-05)† | 6.95E-01(4.54E-05) |
| | 50 | 7.15E-01(7.16E-08)‡ | 2.72E-01(2.35E-08)† | 6.88E-01(1.50E-05)† | 6.99E-01(1.49E-05)† | 7.03E-01(6.87E-06) |
| WFG4 | 5 | 6.75E-01(8.64E-07)† | 3.39E-01(2.26E-05)† | 7.86E-01(9.93E-07)† | 7.87E-01(8.24E-07)† | 7.89E-01(7.27E-07) |
| | 25 | 6.98E-01(1.34E-05)† | 3.52E-01(1.86E-05)† | 7.94E-01(2.21E-07)† | 7.94E-01(2.53E-07)† | 7.95E-01(4.62E-08) |
| | 50 | 7.17E-01(3.51E-07)† | 3.50E-01(1.09E-05)† | 7.95E-01(6.66E-08)† | 7.95E-01(1.15E-07)† | 7.95E-01(2.87E-08) |
| WFG5 | 5 | 6.55E-01(3.32E-06)† | 5.47E-01(5.11E-06)† | 7.62E-01(1.10E-06)‡ | 7.58E-01(4.36E-06)‡ | 7.54E-01(3.34E-06) |
| | 25 | 6.83E-01(1.07E-05)† | 5.47E-01(3.04E-06)† | 7.70E-01(2.64E-07)‡ | 7.64E-01(2.88E-06)† | 7.63E-01(1.85E-06) |
| | 50 | 6.96E-01(4.98E-07)† | 5.45E-01(1.99E-06)† | 7.70E-01(1.51E-07)‡ | 7.66E-01(3.73E-06)† | 7.64E-01(2.25E-06) |
| WFG6 | 5 | 6.68E-01(9.91E-06)† | 1.61E-01(1.87E-05)† | 7.53E-01(1.34E-05)‡ | 7.11E-01(2.87E-04)‡ | 7.04E-01(2.15E-04) |
| | 25 | 7.10E-01(2.38E-06)† | 1.64E-01(3.71E-05)† | 7.85E-01(1.41E-06)‡ | 7.28E-01(2.89E-04)† | 7.27E-01(2.81E-04) |
| | 50 | 7.29E-01(1.21E-06)† | 1.65E-01(4.34E-05)† | 7.92E-01(3.22E-07)‡ | 7.32E-01(2.96E-04)† | 7.32E-01(3.65E-04) |
| WFG7 | 5 | 6.74E-01(3.98E-06)† | 6.67E-01(5.20E-05)† | 7.71E-01(1.98E-06)† | 7.72E-01(1.70E-06)† | 7.76E-01(2.04E-06) |
| | 25 | 7.36E-01(4.87E-06)† | 7.14E-01(4.09E-05)† | 7.92E-01(2.85E-07)† | 7.92E-01(2.39E-07)† | 7.93E-01(3.72E-07) |
| | 50 | 7.43E-01(9.64E-07)† | 7.26E-01(3.67E-05)† | 7.95E-01(1.34E-07) | 7.95E-01(5.18E-08) | 7.95E-01(1.00E-07) |
| WFG8 | 5 | 6.72E-01(1.49E-06)† | 6.68E-01(3.89E-05)† | 7.70E-01(3.05E-06)† | 7.73E-01(1.79E-06)† | 7.76E-01(3.24E-06) |
| | 25 | 7.30E-01(1.86E-06)† | 7.15E-01(3.25E-05)† | 7.92E-01(3.59E-07)† | 7.92E-01(3.19E-07)† | 7.93E-01(2.04E-07) |
| | 50 | 7.43E-01(2.13E-07)† | 7.26E-01(2.26E-05)† | 7.95E-01(8.52E-08) | 7.95E-01(5.29E-08) | 7.95E-01(8.28E-08) |
| WFG9 | 5 | 6.60E-01(4.93E-06)‡ | 6.20E-01(1.09E-02)‡ | 6.84E-01(9.88E-03)‡ | 4.73E-01(7.17E-04)† | 4.82E-01(1.89E-04) |
| | 25 | 7.31E-01(1.26E-06)‡ | 6.69E-01(1.05E-02)‡ | 7.09E-01(1.15E-02)‡ | 4.78E-01(7.50E-04)† | 4.88E-01(2.10E-04) |
| | 50 | 7.42E-01(5.25E-07)‡ | 6.78E-01(9.40E-03)‡ | 7.11E-01(1.16E-02)‡ | 4.81E-01(6.87E-04)† | 4.90E-01(2.24E-04) |
| +-= | | 32/5/1 | 36/3/0 | 23/12/4 | 30/4/5 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.23: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 50 (i.e. $\tau_t = 50$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 1.59E+02(1.98E+02)† | 1.12E+03(3.31E+02)† | 1.74E+01(4.49E+00)† | 2.01E+01(4.70E+00)† | 1.44E+01(9.25E+00) |
| | 25 | 1.37E+02(1.01E+02)† | 1.13E+03(5.93E+02)† | 6.07E+00(3.46E+00)† | 6.54E+00(3.93E+00)† | 2.00E+00(1.26E+00) |
| | 50 | 1.27E+02(7.71E+01)† | 1.15E+03(7.10E+02)† | 1.83E+00(7.53E-01)† | 2.33E+00(1.52E+00)† | 3.31E-01(5.73E-02) |
| F2 | 5 | 4.23E-01(2.96E-06)† | 7.70E+02(9.02E+01)† | 7.72E-02(1.20E-05)† | 6.39E-02(8.57E-06) | 6.50E-02(2.73E-05) |
| | 25 | 3.18E-01(1.75E-05)† | 7.72E+02(1.18E+02)† | 4.26E-02(6.85E-06)† | 3.89E-02(4.58E-06)† | 3.61E-02(6.08E-06) |
| | 50 | 3.00E-01(3.11E-07)† | 7.73E+02(1.46E+02)† | 3.19E-02(1.86E-06)† | 3.12E-02(1.62E-06)† | 2.92E-02(1.17E-06) |
| F3 | 5 | 2.72E+02(9.40E+02)† | 1.04E+03(4.05E+02)† | 4.43E+01(2.74E+01)† | 5.08E+01(6.20E+01)† | 3.71E+01(4.50E+01) |
| | 25 | 2.49E+02(8.51E+02)† | 8.99E+02(2.52E+02)† | 1.89E+01(3.08E+01)† | 1.92E+01(2.34E+01)† | 6.51E+00(9.95E+00) |
| | 50 | 2.49E+02(4.14E+02)† | 8.58E+02(1.39E+02)† | 6.16E+00(8.45E+00)† | 8.27E+00(1.42E+01)† | 1.23E+00(1.99E+00) |
| F4 | 5 | 6.52E-01(2.43E-05)† | 1.84E-01(9.07E-04)† | 1.18E-01(2.68E-04)† | 7.12E-02(5.58E-04)† | 7.15E-02(1.09E-04) |
| | 25 | 4.42E-01(8.75E-06)† | 4.71E-02(1.13E-04)† | 3.74E-02(6.19E-06)† | 2.30E-02(1.16E-05) | 2.75E-02(6.73E-06) |
| | 50 | 4.06E-01(1.72E-06)† | 3.86E-02(9.48E-05)† | 2.64E-02(8.19E-07)† | 2.00E-02(1.14E-05)† | 2.40E-02(1.79E-06) |
| WFG1 | 5 | 3.12E-01(6.79E-06)† | 9.74E+02(5.62E+02)† | 2.46E-01(4.34E-05)† | 2.32E-01(5.12E-05)† | 2.24E-01(5.34E-05) |
| | 25 | 2.67E-01(7.53E-06)† | 9.75E+02(7.41E+02)† | 2.09E-01(8.49E-05)† | 1.93E-01(7.85E-05)† | 1.76E-01(5.28E-05) |
| | 50 | 2.46E-01(7.65E-06)† | 9.79E+02(4.83E+02)† | 1.66E-01(4.80E-05)† | 1.62E-01(4.27E-05)† | 1.58E-01(4.04E-05) |
| WFG2 | 5 | 5.65E-01(2.41E-05)† | 1.21E+00(2.56E-04)† | 3.37E-01(9.18E-05) | 3.26E-01(2.12E-04) | 3.39E-01(2.54E-04) |
| | 25 | 5.20E-01(2.16E-05)† | 1.20E+00(7.99E-05)† | 3.18E-01(6.35E-05)‡ | 3.24E-01(1.17E-04) | 3.29E-01(1.49E-04) |
| | 50 | 5.08E-01(9.53E-05)† | 1.20E+00(2.32E-05)† | 3.22E-01(8.05E-05)‡ | 3.30E-01(8.81E-05) | 3.30E-01(1.18E-04) |
| WFG3 | 5 | 1.74E+00(1.40E-04)† | 1.33E+00(1.30E-04)‡ | 1.55E+00(1.13E-03)† | 1.50E+00(2.92E-03) | 1.38E+00(2.34E-03) |
| | 25 | 1.53E+00(3.21E-05)† | 1.30E+00(6.77E-06) | 1.37E+00(7.23E-04)† | 1.34E+00(2.38E-03) | 1.32E+00(1.92E-03) |
| | 50 | 1.54E+00(1.14E-04)† | 1.30E+00(9.09E-06) | 1.35E+00(4.40E-04)† | 1.32E+00(1.42E-03) | 1.31E+00(1.50E-03) |
| WFG4 | 5 | 4.62E-01(9.57E-06)† | 9.05E-01(1.04E-04)† | 1.91E-01(6.53E-06)† | 1.88E-01(8.88E-06)† | 1.79E-01(9.04E-06) |
| | 25 | 3.88E-01(1.98E-05)† | 8.45E-01(6.29E-05)† | 1.58E-01(4.99E-06)† | 1.57E-01(5.68E-06)† | 1.56E-01(6.24E-06) |
| | 50 | 3.50E-01(9.29E-06)† | 8.22E-01(4.85E-05)† | 1.55E-01(4.00E-06)† | 1.55E-01(4.22E-06)† | 1.51E-01(6.39E-06) |
| WFG5 | 5 | 4.76E-01(5.66E-06)† | 1.80E-01(3.31E-05)‡ | 2.21E-01(1.59E-05)‡ | 2.22E-01(2.83E-05)‡ | 2.33E-01(2.91E-05) |
| | 25 | 3.95E-01(1.98E-05)† | 1.52E-01(1.02E-04)‡ | 1.84E-01(6.43E-06)‡ | 1.92E-01(1.15E-05)‡ | 1.96E-01(1.45E-05) |
| | 50 | 3.67E-01(1.85E-05)† | 1.36E-01(1.39E-04)‡ | 1.78E-01(5.37E-06)‡ | 1.85E-01(8.93E-06)‡ | 1.88E-01(1.20E-05) |
| WFG6 | 5 | 5.16E-01(3.02E-06)† | 7.96E-01(1.12E-03)† | 3.02E-01(2.42E-05)‡ | 3.45E-01(9.47E-04)‡ | 3.68E-01(7.39E-04) |
| | 25 | 3.76E-01(7.83E-07)† | 7.70E-01(2.82E-03)† | 1.75E-01(7.69E-06)‡ | 2.61E-01(9.37E-04)‡ | 2.65E-01(8.11E-04) |
| | 50 | 3.39E-01(1.88E-06)† | 7.68E-01(3.22E-03)† | 1.62E-01(9.59E-06)‡ | 2.50E-01(9.32E-04) | 2.50E-01(9.65E-04) |
| WFG7 | 5 | 5.08E-01(8.93E-06)† | 2.57E-01(9.50E-05)† | 2.31E-01(9.13E-06)† | 2.26E-01(1.47E-05)† | 2.21E-01(8.79E-06) |
| | 25 | 3.91E-01(5.67E-06)† | 1.39E-01(7.10E-05)‡ | 1.63E-01(5.65E-06)† | 1.62E-01(4.80E-06)† | 1.61E-01(5.40E-06) |
| | 50 | 3.69E-01(2.21E-06)† | 1.16E-01(5.42E-05)‡ | 1.57E-01(7.55E-06)† | 1.57E-01(4.10E-06)† | 1.55E-01(4.72E-06) |
| WFG8 | 5 | 5.11E-01(1.42E-05)† | 2.55E-01(1.08E-04)† | 2.30E-01(1.74E-05)† | 2.26E-01(1.47E-05)† | 2.20E-01(1.59E-05) |
| | 25 | 3.96E-01(1.13E-06)† | 1.38E-01(8.29E-05)‡ | 1.62E-01(4.57E-06)† | 1.62E-01(5.43E-06)† | 1.62E-01(5.17E-06) |
| | 50 | 3.63E-01(1.28E-05)† | 1.18E-01(7.02E-05)‡ | 1.57E-01(3.80E-06)† | 1.57E-01(5.71E-06)† | 1.55E-01(5.42E-06) |
| WFG9 | 5 | 5.57E-01(6.27E-05)‡ | 4.20E-01(4.53E-02)‡ | 4.30E-01(3.26E-02)‡ | 7.72E-01(4.43E-04) | 7.86E-01(1.74E-04) |
| | 25 | 3.96E-01(4.38E-05)‡ | 2.79E-01(3.90E-02)‡ | 3.26E-01(4.16E-02)‡ | 7.49E-01(4.11E-04) | 7.62E-01(2.69E-04) |
| | 50 | 3.68E-01(6.03E-06)‡ | 2.49E-01(3.28E-02)‡ | 3.13E-01(4.00E-02)‡ | 7.42E-01(3.78E-04) | 7.53E-01(3.31E-04) |
| +-= | | 36/3/0 | 26/11/2 | 27/11/1 | 22/5/12 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.24: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 7.48E-03(2.07E-06)† | 0.00E+00(0.00E+00)† | 7.90E-01(1.12E-02)† | 7.86E-01(1.07E-02)† | 8.36E-01(2.08E-03) |
| | 25 | 1.53E-01(7.69E-04)† | 0.00E+00(0.00E+00)† | 9.49E-01(4.48E-03)† | 9.71E-01(1.48E-03)† | 9.88E-01(6.07E-06) |
| | 50 | 1.91E-01(4.34E-04)† | 0.00E+00(0.00E+00)† | 9.73E-01(1.67E-03)† | 9.90E-01(4.89E-05)† | 9.92E-01(4.73E-07) |
| | 100 | 3.72E-01(1.30E-03)† | 0.00E+00(0.00E+00)† | 9.91E-01(4.01E-05)† | 9.92E-01(1.46E-06)† | 9.93E-01(8.39E-08) |
| F2 | 5 | 8.87E-01(4.46E-06)† | 6.96E-01(1.39E-03)† | 9.36E-01(2.27E-07)† | 9.37E-01(6.05E-08)‡ | 9.37E-01(2.55E-07) |
| | 25 | 9.03E-01(5.52E-06)† | 7.38E-01(1.08E-03)† | 9.38E-01(1.09E-08)† | 9.38E-01(8.93E-09) | 9.38E-01(9.62E-09) |
| | 50 | 9.04E-01(2.53E-05)† | 7.39E-01(1.07E-03)† | 9.39E-01(3.31E-09)† | 9.39E-01(5.13E-09)† | 9.39E-01(3.19E-09) |
| | 100 | 9.06E-01(2.96E-06)† | 7.39E-01(1.07E-03)† | 9.39E-01(1.06E-09)† | 9.39E-01(1.73E-09)† | 9.39E-01(1.28E-09) |
| F3 | 5 | 0.00E+00(0.00E+00)† | 0.00E+00(0.00E+00)† | 2.28E-01(1.41E-02)† | 3.20E-01(1.49E-02)† | 4.96E-01(9.23E-03) |
| | 25 | 4.34E-03(5.07E-05)† | 0.00E+00(0.00E+00)† | 6.44E-01(3.60E-02)† | 6.96E-01(4.62E-02)† | 8.82E-01(2.03E-03) |
| | 50 | 5.30E-02(4.64E-03)† | 0.00E+00(0.00E+00)† | 8.03E-01(2.29E-02)† | 8.29E-01(2.25E-02)† | 9.28E-01(7.30E-05) |
| | 100 | 1.58E-01(4.90E-04)† | 0.00E+00(0.00E+00)† | 9.01E-01(4.92E-03)† | 8.98E-01(3.91E-03)† | 9.35E-01(5.27E-06) |
| F4 | 5 | 8.30E-01(1.30E-05)† | 9.16E-01(3.80E-05)† | 9.30E-01(8.05E-06)‡ | 8.96E-01(1.14E-03)† | 9.24E-01(7.12E-04) |
| | 25 | 8.37E-01(1.88E-06)† | 9.34E-01(2.86E-07)† | 9.38E-01(2.70E-08)‡ | 9.06E-01(8.68E-04)† | 9.28E-01(6.18E-04) |
| | 50 | 8.42E-01(9.80E-06)† | 9.35E-01(8.11E-08)† | 9.39E-01(4.00E-09)‡ | 9.06E-01(8.81E-04)† | 9.29E-01(6.30E-04) |
| | 100 | 8.52E-01(1.02E-06)† | 9.36E-01(2.01E-08)† | 9.39E-01(1.43E-09)‡ | 9.06E-01(8.90E-04)† | 9.29E-01(6.36E-04) |
| WFG1 | 5 | 8.97E-01(9.35E-07)† | 3.19E-01(1.32E-05)† | 8.85E-01(6.41E-06)† | 8.92E-01(1.21E-05)† | 9.04E-01(8.26E-06) |
| | 25 | 9.18E-01(1.46E-06)† | 3.21E-01(1.01E-05)† | 9.17E-01(2.00E-05)† | 9.23E-01(1.00E-05)† | 9.32E-01(2.98E-06) |
| | 50 | 9.30E-01(8.85E-07)† | 3.20E-01(1.61E-05)† | 9.38E-01(3.35E-06)† | 9.40E-01(2.54E-06)† | 9.42E-01(2.10E-06) |
| | 100 | 9.40E-01(2.13E-07)† | 3.21E-01(1.44E-05)† | 9.48E-01(6.18E-07)† | 9.48E-01(6.89E-07)† | 9.49E-01(3.29E-07) |
| WFG2 | 5 | 9.17E-01(1.25E-06)† | 2.59E-01(2.63E-08)† | 9.29E-01(6.58E-05)† | 9.34E-01(1.89E-06)† | 9.37E-01(9.84E-07) |
| | 25 | 9.40E-01(1.92E-08)† | 2.59E-01(4.28E-09)† | 9.43E-01(7.36E-08)† | 9.43E-01(4.03E-08)† | 9.44E-01(3.26E-08) |
| | 50 | 9.41E-01(2.27E-08)† | 2.59E-01(2.63E-09)† | 9.44E-01(7.26E-09)† | 9.44E-01(9.33E-09)† | 9.44E-01(3.67E-09) |
| | 100 | 9.42E-01(6.51E-09)† | 2.59E-01(2.63E-09)† | 9.44E-01(7.79E-09)† | 9.44E-01(3.03E-09)† | 9.44E-01(3.91E-09) |
| WFG3 | 5 | 6.58E-01(7.03E-06)‡ | 2.71E-01(3.76E-07)† | 6.19E-01(5.86E-05)† | 6.42E-01(1.10E-04)† | 6.48E-01(1.11E-04) |
| | 25 | 7.11E-01(1.26E-07)‡ | 2.72E-01(4.04E-08)† | 6.76E-01(2.28E-05)† | 6.87E-01(5.36E-05)† | 6.92E-01(2.35E-05) |
| | 50 | 7.14E-01(2.30E-08)‡ | 2.72E-01(2.33E-08)† | 6.91E-01(1.68E-05)† | 6.98E-01(2.15E-05)† | 7.01E-01(1.16E-05) |
| | 100 | 7.18E-01(2.19E-07)‡ | 2.72E-01(1.22E-08)† | 7.00E-01(8.10E-06)† | 7.04E-01(5.00E-06)† | 7.05E-01(4.11E-06) |
| WFG4 | 5 | 6.76E-01(3.94E-06)† | 3.48E-01(1.28E-05)† | 7.87E-01(4.41E-07)† | 7.87E-01(7.42E-07)† | 7.90E-01(2.85E-07) |
| | 25 | 7.01E-01(1.30E-05)† | 3.54E-01(2.34E-05)† | 7.94E-01(1.08E-07)† | 7.95E-01(4.25E-08)† | 7.95E-01(7.85E-08) |
| | 50 | 7.15E-01(7.93E-06)† | 3.52E-01(1.25E-05)† | 7.95E-01(4.92E-08)† | 7.95E-01(5.09E-08)† | 7.95E-01(4.40E-08) |
| | 100 | 7.26E-01(4.93E-06)† | 3.51E-01(2.07E-06)† | 7.96E-01(4.29E-08)† | 7.96E-01(3.98E-08)† | 7.96E-01(3.51E-08) |
| WFG5 | 5 | 6.49E-01(1.26E-05)† | 5.47E-01(6.22E-06)† | 7.63E-01(9.06E-07)‡ | 7.64E-01(4.53E-06)‡ | 7.62E-01(1.36E-06) |
| | 25 | 6.87E-01(1.50E-07)† | 5.47E-01(3.86E-06)† | 7.70E-01(9.53E-08)‡ | 7.68E-01(3.94E-06) | 7.69E-01(1.02E-06) |
| | 50 | 6.94E-01(6.82E-07)† | 5.45E-01(3.20E-06)† | 7.71E-01(5.13E-08)‡ | 7.70E-01(4.48E-06) | 7.70E-01(1.19E-06) |
| | 100 | 7.07E-01(2.74E-06)† | 5.43E-01(2.59E-06)† | 7.71E-01(3.44E-08)‡ | 7.70E-01(4.79E-06) | 7.71E-01(1.05E-06) |
| WFG6 | 5 | 6.78E-01(4.38E-05)† | 1.61E-01(1.95E-05)† | 7.55E-01(1.04E-05)‡ | 7.32E-01(3.27E-04)‡ | 7.04E-01(1.20E-03) |
| | 25 | 7.23E-01(5.29E-06)† | 1.65E-01(3.77E-05)† | 7.87E-01(1.40E-06)‡ | 7.54E-01(4.78E-04)‡ | 7.41E-01(6.61E-04) |
| | 50 | 7.28E-01(4.08E-07)† | 1.66E-01(4.26E-05)† | 7.93E-01(3.84E-07)‡ | 7.59E-01(5.57E-04)‡ | 7.46E-01(7.50E-04) |
| | 100 | 7.39E-01(4.02E-06) | 1.66E-01(4.51E-05)† | 7.95E-01(1.43E-07)‡ | 7.63E-01(6.34E-04)‡ | 7.49E-01(8.57E-04) |
| WFG7 | 5 | 6.82E-01(9.14E-07)† | 6.86E-01(5.50E-05)† | 7.71E-01(1.94E-06)† | 7.72E-01(1.42E-06)† | 7.77E-01(1.59E-06) |
| | 25 | 7.31E-01(2.49E-06)† | 7.26E-01(3.21E-05)† | 7.92E-01(1.55E-07)† | 7.92E-01(3.52E-07)† | 7.93E-01(1.09E-07) |
| | 50 | 7.44E-01(1.21E-06)† | 7.32E-01(3.79E-05)† | 7.95E-01(5.75E-08) | 7.95E-01(9.25E-08) | 7.95E-01(6.52E-08) |
| | 100 | 7.50E-01(1.23E-06)† | 7.34E-01(2.87E-05)† | 7.96E-01(4.73E-08) | 7.96E-01(4.30E-08) | 7.96E-01(4.31E-08) |
| WFG8 | 5 | 6.77E-01(2.43E-06)† | 6.85E-01(2.42E-05)† | 7.71E-01(4.33E-06)† | 7.72E-01(2.05E-06)† | 7.77E-01(1.47E-06) |
| | 25 | 7.35E-01(6.26E-06)† | 7.25E-01(3.19E-05)† | 7.92E-01(4.09E-07)† | 7.92E-01(2.52E-07)† | 7.93E-01(2.86E-07) |
| | 50 | 7.44E-01(1.14E-06)† | 7.34E-01(2.68E-05)† | 7.95E-01(1.07E-07) | 7.95E-01(1.42E-07) | 7.95E-01(1.18E-07) |
| | 100 | 7.51E-01(1.05E-07)† | 7.37E-01(2.08E-05)† | 7.96E-01(6.22E-08) | 7.96E-01(3.80E-08) | 7.96E-01(3.82E-08) |
| WFG9 | 5 | 6.78E-01(1.38E-05)‡ | 6.57E-01(3.48E-03)‡ | 6.48E-01(8.32E-03)‡ | 4.75E-01(3.19E-04)† | 4.88E-01(1.47E-04) |
| | 25 | 7.37E-01(1.01E-06)‡ | 7.05E-01(3.20E-03)‡ | 6.70E-01(9.76E-03)‡ | 4.80E-01(3.82E-04)† | 4.94E-01(1.86E-04) |
| | 50 | 7.43E-01(1.27E-06)‡ | 7.13E-01(2.79E-03)‡ | 6.72E-01(9.84E-03)‡ | 4.81E-01(3.84E-04)† | 4.95E-01(1.92E-04) |
| | 100 | 7.51E-01(4.79E-07)‡ | 7.19E-01(2.33E-03)‡ | 6.72E-01(9.85E-03)‡ | 4.83E-01(3.29E-04)† | 4.95E-01(2.00E-04) |
| +-= | | 43/8/1 | 48/4/0 | 32/16/4 | 38/6/8 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.25: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when the frequency of change is 100 (i.e. $\tau_t = 100$), averaged across the cases of decreasing NObj for the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 1.82E+02(1.44E+00)† | 1.14E+03(2.78E+02)† | 1.51E+01(3.42E+00)† | 1.86E+01(5.14E+00)† | 1.23E+01(3.23E+00) |
| | 25 | 1.51E+02(1.07E+00)† | 1.15E+03(2.87E+02)† | 4.25E+00(1.71E+00)† | 5.03E+00(2.15E+00)† | 1.48E+00(4.96E-01) |
| | 50 | 1.31E+02(9.61E-01)† | 1.17E+03(2.76E+02)† | 1.22E+00(4.08E-01)† | 1.41E+00(2.23E-01)† | 3.13E-01(5.17E-02) |
| | 100 | 1.20E+02(4.37E+00)† | 1.18E+03(2.85E+02)† | 2.98E-01(4.95E-02)† | 5.63E-01(5.29E-02)† | 9.69E-02(5.00E-03) |
| F2 | 5 | 4.07E-01(9.70E-06)† | 7.62E+02(5.06E+01)† | 7.43E-02(1.51E-05)† | 6.10E-02(1.69E-05)‡ | 6.57E-02(1.73E-05) |
| | 25 | 3.31E-01(3.30E-05)† | 7.60E+02(2.14E+02)† | 4.19E-02(4.72E-06)† | 3.74E-02(5.48E-06)† | 3.65E-02(3.44E-06) |
| | 50 | 3.00E-01(5.62E-05)† | 7.65E+02(2.34E+02)† | 3.15E-02(1.29E-06)† | 2.97E-02(1.39E-06)† | 2.96E-02(1.11E-06) |
| | 100 | 2.88E-01(4.25E-05)† | 7.78E+02(2.39E+02)† | 2.62E-02(5.80E-07)† | 2.56E-02(4.53E-07) | 2.57E-02(3.12E-07) |
| F3 | 5 | 3.63E+02(1.29E+02)† | 1.03E+03(4.65E+02)† | 3.68E+01(2.35E+01)† | 4.29E+01(2.70E+01)† | 3.27E+01(3.81E+01) |
| | 25 | 3.04E+02(8.94E+01)† | 8.85E+02(3.31E+02)† | 1.30E+01(1.69E+01)† | 1.42E+01(8.97E+00)† | 5.20E+00(5.50E+00) |
| | 50 | 2.73E+02(2.57E+01)† | 8.47E+02(1.71E+02)† | 4.00E+00(3.07E+00)† | 5.04E+00(3.72E+00)† | 8.83E-01(5.94E-01) |
| | 100 | 2.40E+02(2.51E+01)† | 8.12E+02(1.75E+02)† | 1.16E+00(9.68E-01)† | 1.81E+00(6.43E-01)† | 2.76E-01(6.34E-02) |
| F4 | 5 | 6.49E-01(6.36E-06)† | 1.69E-01(8.06E-04)† | 1.09E-01(1.51E-04)† | 6.68E-02(4.96E-04)‡ | 6.93E-02(2.47E-04) |
| | 25 | 4.69E-01(6.86E-07)† | 4.62E-02(9.29E-05)† | 3.48E-02(5.60E-06)† | 2.35E-02(8.28E-06)‡ | 2.67E-02(8.91E-06) |
| | 50 | 4.23E-01(1.20E-05)† | 3.89E-02(7.29E-05)† | 2.60E-02(6.29E-07)† | 2.08E-02(5.36E-06)‡ | 2.32E-02(4.02E-06) |
| | 100 | 3.93E-01(2.20E-07)† | 3.66E-02(7.14E-05)† | 2.43E-02(2.58E-07)† | 2.04E-02(7.72E-06)‡ | 2.31E-02(3.90E-06) |
| WFG1 | 5 | 3.09E-01(1.53E-05)† | 9.67E+02(5.17E+02)† | 2.18E-01(2.02E-05)† | 2.15E-01(1.77E-05)† | 2.11E-01(1.84E-05) |
| | 25 | 2.69E-01(7.54E-06)† | 9.69E+02(5.38E+02)† | 1.91E-01(5.76E-05)† | 1.81E-01(4.88E-05)† | 1.64E-01(3.18E-05) |
| | 50 | 2.42E-01(1.63E-06)† | 9.74E+02(4.97E+02)† | 1.56E-01(2.63E-05)† | 1.53E-01(2.38E-05)† | 1.49E-01(2.11E-05) |
| | 100 | 2.31E-01(3.12E-06)† | 9.75E+02(3.72E+02)† | 1.47E-01(2.11E-05)† | 1.45E-01(1.63E-05)† | 1.44E-01(2.04E-05) |
| WFG2 | 5 | 5.87E-01(7.06E-05)† | 1.22E+00(2.44E-04)† | 3.45E-01(5.58E-04)† | 3.36E-01(1.39E-04)† | 3.26E-01(1.25E-04) |
| | 25 | 4.59E-01(4.51E-05)† | 1.20E+00(3.43E-05)† | 3.27E-01(1.43E-04)† | 3.26E-01(1.27E-04)† | 3.19E-01(1.22E-04) |
| | 50 | 4.71E-01(7.77E-05)† | 1.20E+00(1.42E-05)† | 3.29E-01(1.23E-04)† | 3.28E-01(1.21E-04)† | 3.22E-01(8.16E-05) |
| | 100 | 5.37E-01(1.04E-04)† | 1.20E+00(1.73E-05)† | 3.27E-01(9.81E-05)† | 3.24E-01(1.20E-04)† | 3.22E-01(7.88E-05) |
| WFG3 | 5 | 1.74E+00(4.14E-05)† | 1.33E+00(7.82E-05)‡ | 1.56E+00(1.13E-03)† | 1.55E+00(2.67E-03)† | 1.47E+00(2.52E-03) |
| | 25 | 1.55E+00(6.19E-05)† | 1.30E+00(5.99E-06)‡ | 1.37E+00(1.25E-03)† | 1.36E+00(1.74E-03)† | 1.34E+00(1.11E-03) |
| | 50 | 1.58E+00(6.69E-05)† | 1.30E+00(7.63E-06)‡ | 1.35E+00(7.73E-04)† | 1.33E+00(1.29E-03)† | 1.32E+00(9.00E-04) |
| | 100 | 1.58E+00(7.89E-05)† | 1.30E+00(8.12E-06)‡ | 1.36E+00(4.45E-04)† | 1.32E+00(7.99E-04) | 1.33E+00(6.27E-04) |
| WFG4 | 5 | 4.64E-01(4.30E-05)† | 8.87E-01(7.37E-05)† | 1.88E-01(8.31E-06)† | 1.86E-01(9.41E-06)† | 1.78E-01(6.33E-06) |
| | 25 | 3.89E-01(5.98E-05)† | 8.33E-01(5.94E-05)† | 1.58E-01(6.82E-06)† | 1.57E-01(3.02E-06)† | 1.55E-01(4.60E-06) |
| | 50 | 3.54E-01(1.74E-05)† | 8.13E-01(3.66E-05)† | 1.55E-01(3.38E-06)† | 1.55E-01(4.72E-06)† | 1.51E-01(6.20E-06) |
| | 100 | 3.36E-01(1.02E-05)† | 8.04E-01(1.37E-05)† | 1.53E-01(4.50E-06)† | 1.53E-01(5.16E-06)† | 1.47E-01(6.03E-06) |
| WFG5 | 5 | 4.67E-01(2.04E-06)† | 1.86E-01(2.05E-05)‡ | 2.20E-01(1.05E-05)† | 2.13E-01(1.82E-05)‡ | 2.19E-01(1.32E-05) |
| | 25 | 3.87E-01(1.08E-06)† | 1.53E-01(6.82E-05)‡ | 1.84E-01(4.40E-06)‡ | 1.86E-01(5.82E-06) | 1.86E-01(2.36E-06) |
| | 50 | 3.55E-01(3.75E-06)† | 1.37E-01(1.25E-04)‡ | 1.79E-01(4.08E-06) | 1.81E-01(9.88E-06) | 1.80E-01(3.98E-06) |
| | 100 | 3.41E-01(4.48E-06)† | 1.36E-01(5.27E-05)‡ | 1.77E-01(2.01E-06) | 1.78E-01(1.10E-05) | 1.78E-01(3.11E-06) |
| WFG6 | 5 | 4.77E-01(1.10E-04)† | 7.90E-01(9.67E-04)† | 2.98E-01(3.76E-05)‡ | 3.14E-01(9.00E-04)‡ | 3.85E-01(6.59E-03) |
| | 25 | 3.65E-01(2.18E-05)† | 7.62E-01(2.63E-03)† | 1.75E-01(8.32E-06)‡ | 2.19E-01(1.23E-03)‡ | 2.42E-01(1.81E-03) |
| | 50 | 3.28E-01(1.15E-05)† | 7.60E-01(3.04E-03)† | 1.61E-01(7.09E-06)‡ | 2.06E-01(1.36E-03)‡ | 2.27E-01(2.00E-03) |
| | 100 | 3.15E-01(4.98E-06)† | 7.60E-01(3.02E-03)† | 1.57E-01(5.98E-06)‡ | 1.99E-01(1.42E-03)‡ | 2.19E-01(2.20E-03) |
| WFG7 | 5 | 5.07E-01(5.54E-06)† | 2.61E-01(1.35E-04)† | 2.29E-01(1.28E-05)† | 2.24E-01(1.39E-05)† | 2.16E-01(1.17E-05) |
| | 25 | 3.95E-01(1.54E-05)† | 1.42E-01(9.88E-05)‡ | 1.62E-01(6.05E-06)† | 1.61E-01(9.23E-06) | 1.61E-01(7.22E-06) |
| | 50 | 3.69E-01(3.59E-06)† | 1.17E-01(7.98E-05)‡ | 1.56E-01(4.86E-06)† | 1.56E-01(6.78E-06)† | 1.55E-01(4.51E-06) |
| | 100 | 3.52E-01(2.00E-06)† | 1.18E-01(4.36E-05)‡ | 1.51E-01(1.73E-05)† | 1.51E-01(1.65E-05)† | 1.50E-01(6.06E-06) |
| WFG8 | 5 | 5.01E-01(8.37E-06)† | 2.63E-01(1.01E-04)† | 2.29E-01(1.13E-05)† | 2.25E-01(1.30E-05)† | 2.16E-01(1.21E-05) |
| | 25 | 3.93E-01(3.62E-06)† | 1.42E-01(1.03E-04)‡ | 1.62E-01(6.55E-06)† | 1.61E-01(6.54E-06)† | 1.60E-01(6.05E-06) |
| | 50 | 3.69E-01(6.37E-07)† | 1.20E-01(5.26E-05)‡ | 1.57E-01(4.26E-06)† | 1.56E-01(5.63E-06)† | 1.54E-01(8.23E-06) |
| | 100 | 3.54E-01(7.44E-07)† | 1.18E-01(2.71E-05)‡ | 1.52E-01(7.19E-06)† | 1.51E-01(1.30E-05)† | 1.50E-01(6.78E-06) |
| WFG9 | 5 | 5.45E-01(3.96E-06)‡ | 3.52E-01(1.54E-02)‡ | 4.90E-01(2.77E-02)‡ | 7.76E-01(2.01E-04) | 7.82E-01(2.31E-04) |
| | 25 | 4.00E-01(8.65E-07)‡ | 2.12E-01(1.17E-02)‡ | 3.92E-01(3.49E-02)‡ | 7.58E-01(2.85E-04) | 7.58E-01(4.99E-04) |
| | 50 | 3.78E-01(1.50E-05)‡ | 1.85E-01(1.03E-02)‡ | 3.79E-01(3.40E-02)‡ | 7.52E-01(3.09E-04)† | 7.49E-01(6.11E-04) |
| | 100 | 3.57E-01(5.55E-06)‡ | 1.68E-01(9.04E-03)‡ | 3.74E-01(3.36E-02)‡ | 7.49E-01(3.36E-04)† | 7.44E-01(6.56E-04) |
| +-= | | 48/4/0 | 34/18/0 | 41/9/2 | 34/10/8 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.26: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

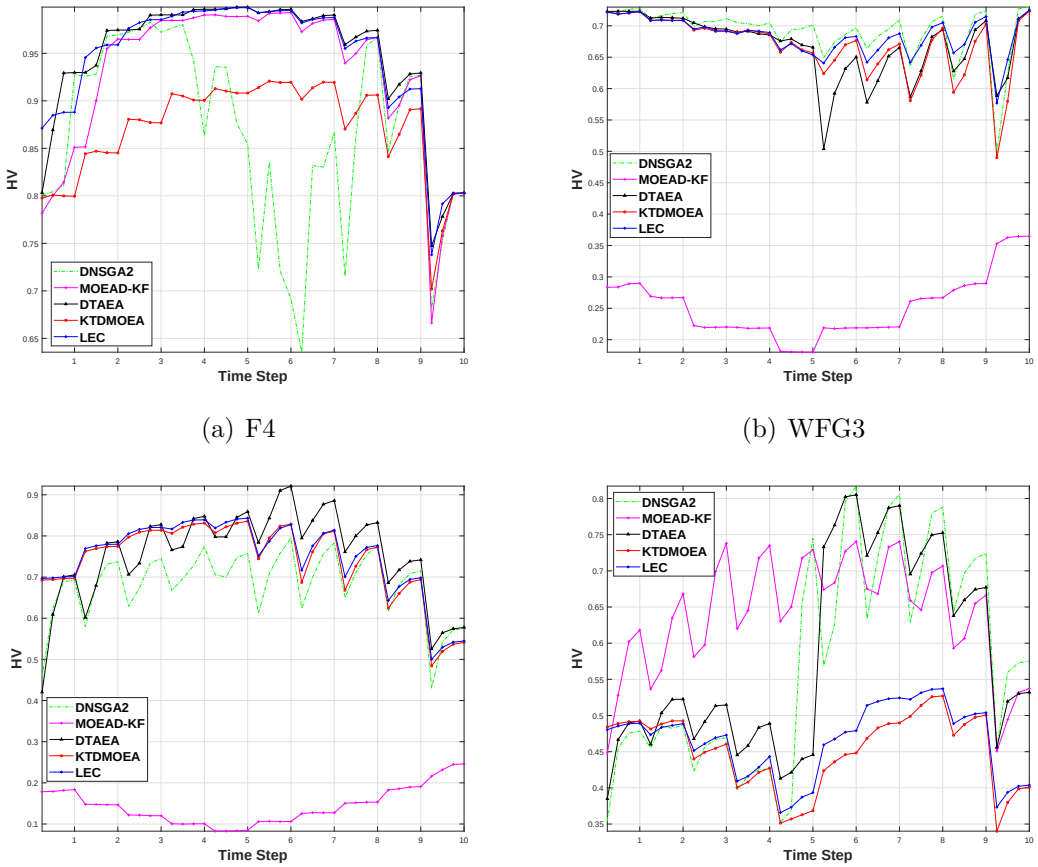
| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 3.67E-02(4.12E-05)† | 0.00E+00(0.00E+00)† | 8.80E-01(4.70E-03)‡ | 8.11E-01(7.75E-03)† | 8.40E-01(7.95E-04) |
| | 25 | 1.92E-01(9.37E-08)† | 0.00E+00(0.00E+00)† | 9.89E-01(2.11E-05)† | 9.57E-01(3.67E-03)† | 9.90E-01(4.09E-06) |
| | 50 | 1.94E-01(8.73E-08)† | 0.00E+00(0.00E+00)† | 9.92E-01(2.58E-06)† | 9.81E-01(1.31E-03)† | 9.92E-01(1.55E-07) |
| | 100 | 3.14E-01(2.80E-04)† | 8.07E-03(4.48E-04)† | 9.93E-01(7.46E-07)† | 9.90E-01(1.59E-04)† | 9.93E-01(2.71E-08) |
| | 200 | 3.93E-01(2.50E-10)† | 1.17E-01(7.27E-03)† | 9.93E-01(2.96E-08)† | 9.93E-01(1.37E-07)† | 9.93E-01(7.81E-09) |
| F2 | 5 | 8.87E-01(2.19E-06)† | 7.39E-01(7.89E-05)† | 9.37E-01(1.72E-07) | 9.37E-01(5.37E-08)‡ | 9.37E-01(1.23E-07) |
| | 25 | 8.98E-01(1.60E-05)† | 7.70E-01(1.02E-05)† | 9.38E-01(7.25E-09)† | 9.38E-01(2.02E-08)† | 9.38E-01(6.27E-09) |
| | 50 | 9.03E-01(2.98E-06)† | 7.71E-01(1.01E-05)† | 9.39E-01(3.03E-09)† | 9.39E-01(1.32E-08)† | 9.39E-01(5.00E-09) |
| | 100 | 8.96E-01(3.65E-05)† | 7.71E-01(1.01E-05)† | 9.39E-01(2.07E-09)† | 9.39E-01(1.92E-09)† | 9.39E-01(8.35E-10) |
| | 200 | 9.03E-01(1.76E-05)† | 7.71E-01(1.03E-05)† | 9.39E-01(3.22E-10)† | 9.39E-01(6.88E-10)† | 9.39E-01(5.51E-10) |
| F3 | 5 | 2.50E-02(4.65E-05)† | 0.00E+00(0.00E+00)† | 3.25E-01(1.55E-02)† | 4.05E-01(1.23E-02)† | 4.75E-01(1.03E-02) |
| | 25 | 1.55E-01(3.77E-04)† | 0.00E+00(0.00E+00)† | 7.82E-01(9.41E-03)† | 7.44E-01(1.69E-02)† | 8.88E-01(1.29E-03) |
| | 50 | 1.60E-01(1.75E-08)† | 0.00E+00(0.00E+00)† | 8.90E-01(2.75E-03)† | 8.45E-01(6.99E-03)† | 9.27E-01(8.40E-05) |
| | 100 | 2.14E-01(4.98E-04)† | 1.85E-02(1.81E-03)† | 9.26E-01(5.88E-04)† | 9.13E-01(1.95E-03)† | 9.36E-01(4.28E-06) |
| | 200 | 3.46E-01(9.19E-09)† | 8.53E-02(5.72E-03)† | 9.36E-01(1.68E-05)† | 9.28E-01(1.11E-03)† | 9.38E-01(4.40E-07) |
| F4 | 5 | 8.38E-01(2.93E-05)† | 9.17E-01(3.68E-05)† | 9.30E-01(6.29E-06)† | 8.66E-01(2.48E-03)† | 9.34E-01(4.87E-06) |
| | 25 | 8.36E-01(2.39E-05)† | 9.34E-01(2.18E-07)† | 9.38E-01(2.38E-08)† | 8.82E-01(1.78E-03)† | 9.38E-01(6.28E-08) |
| | 50 | 8.48E-01(4.79E-06)† | 9.35E-01(3.93E-08)† | 9.39E-01(4.71E-09)† | 8.82E-01(1.81E-03)† | 9.39E-01(5.60E-09) |
| | 100 | 8.58E-01(4.68E-07)† | 9.36E-01(1.16E-08)† | 9.39E-01(1.17E-09)† | 8.82E-01(1.82E-03)† | 9.39E-01(1.42E-09) |
| | 200 | 8.50E-01(3.51E-06)† | 9.36E-01(7.45E-09)† | 9.39E-01(7.10E-10)† | 8.82E-01(1.84E-03)† | 9.39E-01(3.59E-10) |
| WFG1 | 5 | 8.93E-01(3.72E-06)† | 3.20E-01(8.70E-06)† | 8.89E-01(2.48E-06)† | 8.93E-01(1.35E-05)† | 9.04E-01(1.03E-05) |
| | 25 | 9.23E-01(3.75E-06)† | 3.21E-01(1.10E-05)† | 9.19E-01(2.24E-05)† | 9.25E-01(1.67E-05)† | 9.34E-01(5.08E-06) |
| | 50 | 9.36E-01(1.45E-06)† | 3.21E-01(8.74E-06)† | 9.40E-01(2.41E-06)† | 9.41E-01(2.77E-06)† | 9.44E-01(1.16E-06) |
| | 100 | 9.44E-01(2.71E-07)† | 3.22E-01(7.71E-06)† | 9.49E-01(4.61E-07)† | 9.49E-01(7.59E-07)† | 9.50E-01(2.28E-07) |
| | 200 | 9.48E-01(7.08E-08)† | 3.22E-01(7.04E-06)† | 9.52E-01(8.15E-08) | 9.51E-01(6.12E-07)† | 9.52E-01(8.89E-08) |
| WFG2 | 5 | 9.10E-01(1.14E-06)† | 2.59E-01(2.81E-08)† | 9.33E-01(2.66E-06)† | 9.34E-01(2.29E-06)† | 9.37E-01(1.51E-06) |
| | 25 | 9.40E-01(1.64E-07)† | 2.59E-01(3.33E-09)† | 9.43E-01(4.40E-08)† | 9.43E-01(2.68E-08)† | 9.44E-01(2.56E-08) |
| | 50 | 9.41E-01(2.02E-08)† | 2.59E-01(4.35E-09)† | 9.44E-01(7.83E-09) | 9.44E-01(7.13E-09) | 9.44E-01(5.95E-09) |
| | 100 | 9.41E-01(6.64E-09)† | 2.59E-01(3.08E-09)† | 9.44E-01(7.98E-09) | 9.44E-01(3.43E-09) | 9.44E-01(5.09E-09) |
| | 200 | 9.42E-01(5.80E-09)† | 2.59E-01(1.41E-09)† | 9.44E-01(9.86E-09) | 9.44E-01(5.03E-09) | 9.45E-01(3.13E-09) |
| WFG3 | 5 | 6.61E-01(1.77E-06)‡ | 2.71E-01(2.38E-07)† | 6.21E-01(7.06E-05)† | 6.18E-01(1.75E-04)† | 6.34E-01(1.87E-04) |
| | 25 | 7.10E-01(1.87E-07)‡ | 2.72E-01(3.05E-08)† | 6.77E-01(3.21E-05)† | 6.77E-01(6.29E-05)† | 6.81E-01(6.03E-05) |
| | 50 | 7.16E-01(3.23E-07)‡ | 2.72E-01(1.85E-08)† | 6.92E-01(1.90E-05)† | 6.93E-01(2.56E-05)† | 6.97E-01(8.53E-06) |
| | 100 | 7.19E-01(3.73E-07)‡ | 2.72E-01(1.15E-08)† | 7.01E-01(4.66E-06)† | 7.03E-01(4.55E-06)† | 7.04E-01(3.47E-06) |
| | 200 | 7.18E-01(3.49E-08)‡ | 2.73E-01(5.25E-09)† | 7.06E-01(2.04E-06)† | 7.06E-01(2.14E-06) | 7.06E-01(1.74E-06) |
| WFG4 | 5 | 6.88E-01(4.41E-06)† | 3.53E-01(1.15E-05)† | 7.87E-01(7.00E-07)† | 7.87E-01(6.30E-07)† | 7.89E-01(3.10E-07) |
| | 25 | 7.09E-01(2.56E-06)† | 3.59E-01(1.36E-05)† | 7.94E-01(1.20E-07)† | 7.94E-01(5.13E-08)† | 7.95E-01(7.34E-08) |
| | 50 | 7.18E-01(3.97E-06)† | 3.54E-01(1.39E-05)† | 7.95E-01(4.02E-08)† | 7.95E-01(5.84E-08)† | 7.95E-01(3.30E-08) |
| | 100 | 7.29E-01(3.07E-06)† | 3.53E-01(3.88E-06)† | 7.96E-01(3.72E-08) | 7.96E-01(4.65E-08) | 7.96E-01(3.68E-08) |
| | 200 | 7.37E-01(2.15E-06)† | 3.54E-01(2.09E-06)† | 7.95E-01(1.81E-08) | 7.95E-01(2.80E-08) | 7.95E-01(1.21E-08) |
| WFG5 | 5 | 6.66E-01(2.04E-06)† | 5.47E-01(5.79E-06)† | 7.64E-01(6.09E-07)‡ | 7.65E-01(9.54E-07)‡ | 7.63E-01(4.05E-07) |
| | 25 | 6.92E-01(2.06E-06)† | 5.47E-01(2.58E-06)† | 7.70E-01(6.35E-08) | 7.70E-01(1.70E-07) | 7.70E-01(1.06E-07) |
| | 50 | 7.04E-01(5.75E-06)† | 5.45E-01(3.10E-06)† | 7.71E-01(3.28E-08) | 7.71E-01(9.33E-08) | 7.71E-01(8.72E-08) |
| | 100 | 7.11E-01(3.44E-06)† | 5.43E-01(2.65E-06)† | 7.71E-01(2.98E-08) | 7.71E-01(4.75E-08) | 7.71E-01(5.54E-08) |
| | 200 | 7.12E-01(6.53E-06)† | 5.44E-01(2.76E-06)† | 7.71E-01(4.11E-08) | 7.72E-01(1.98E-08) | 7.71E-01(5.11E-08) |
| WFG6 | 5 | 6.85E-01(3.91E-05)† | 1.64E-01(8.03E-06)† | 7.54E-01(9.31E-06)† | 7.47E-01(3.43E-04)† | 7.59E-01(9.93E-06) |
| | 25 | 7.28E-01(1.28E-05)† | 1.69E-01(1.47E-05)† | 7.88E-01(1.50E-06)† | 7.70E-01(4.23E-04)† | 7.89E-01(8.09E-07) |
| | 50 | 7.32E-01(1.70E-05)† | 1.69E-01(1.67E-05)† | 7.93E-01(3.55E-07) | 7.74E-01(4.52E-04)† | 7.93E-01(2.45E-07) |
| | 100 | 7.42E-01(1.50E-05)† | 1.70E-01(1.74E-05)† | 7.95E-01(1.47E-07) | 7.77E-01(4.76E-04)† | 7.95E-01(1.56E-07) |
| | 200 | 7.41E-01(4.74E-06)† | 1.70E-01(1.81E-05)† | 7.95E-01(1.66E-07) | 7.79E-01(4.92E-04)† | 7.95E-01(1.31E-07) |
| WFG7 | 5 | 6.78E-01(9.02E-06)† | 6.86E-01(4.62E-05)† | 7.72E-01(1.91E-06)† | 7.74E-01(1.04E-06)† | 7.77E-01(1.34E-06) |
| | 25 | 7.35E-01(5.66E-06)† | 7.28E-01(4.25E-05)† | 7.92E-01(1.74E-07)† | 7.93E-01(1.53E-07)† | 7.93E-01(1.69E-07) |
| | 50 | 7.46E-01(1.70E-07)† | 7.35E-01(2.81E-05)† | 7.95E-01(7.14E-08) | 7.95E-01(7.54E-08)† | 7.95E-01(7.52E-08) |
| | 100 | 7.50E-01(1.04E-06)† | 7.40E-01(2.02E-05)† | 7.96E-01(4.81E-08) | 7.96E-01(2.64E-08) | 7.96E-01(2.37E-08) |
| | 200 | 7.54E-01(2.38E-06)† | 7.44E-01(9.02E-06)† | 7.96E-01(5.00E-08) | 7.96E-01(2.05E-08) | 7.96E-01(7.83E-08) |
| WFG8 | 5 | 6.81E-01(6.22E-07)† | 6.86E-01(4.25E-05)† | 7.71E-01(2.31E-06)† | 7.73E-01(1.49E-06)† | 7.76E-01(1.66E-06) |
| | 25 | 7.34E-01(2.68E-06)† | 7.30E-01(2.09E-05)† | 7.92E-01(3.01E-07)† | 7.93E-01(2.82E-07)† | 7.93E-01(2.49E-07) |
| | 50 | 7.43E-01(1.00E-06)† | 7.38E-01(2.31E-05)† | 7.95E-01(4.58E-08) | 7.95E-01(5.03E-08) | 7.95E-01(7.64E-08) |
| | 100 | 7.53E-01(3.59E-07)† | 7.42E-01(1.73E-05)† | 7.96E-01(3.08E-08) | 7.96E-01(4.74E-08) | 7.96E-01(4.07E-08) |
| | 200 | 7.57E-01(1.16E-06)† | 7.45E-01(6.90E-06)† | 7.96E-01(3.32E-08) | 7.96E-01(3.31E-08) | 7.96E-01(4.77E-08) |
| WFG9 | 5 | 6.89E-01(3.09E-05)‡ | 6.45E-01(5.49E-03) | 5.72E-01(6.93E-03)† | 4.90E-01(6.12E-05) | 6.39E-01(1.01E-02) |
| | 25 | 7.40E-01(1.72E-05)‡ | 6.95E-01(5.30E-03) | 5.87E-01(8.21E-03)† | 4.95E-01(7.76E-05) | 6.56E-01(1.17E-02) |
| | 50 | 7.50E-01(6.68E-06)‡ | 7.04E-01(4.84E-03) | 5.89E-01(8.29E-03)† | 4.96E-01(7.91E-05) | 6.58E-01(1.17E-02) |
| | 100 | 7.57E-01(3.99E-06)‡ | 7.06E-01(4.54E-03) | 5.89E-01(8.29E-03)† | 4.96E-01(8.11E-05) | 6.58E-01(1.18E-02) |
| | 200 | 7.55E-01(5.96E-08)‡ | 7.03E-01(4.18E-03) | 5.89E-01(8.29E-03)† | 4.97E-01(6.86E-05) | 6.58E-01(1.18E-02) |
| +-= | | 55/10/0 | 60/0/5 | 43/2/20 | 43/2/20 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

Table 6.27: Mean and standard deviation values of MGD metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, averaged across the cases of decreasing NObj for the changing sequence of firstly increasing NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 1.79E+02(4.97E+00)† | 1.15E+03(3.58E+02)† | 1.39E+01(4.02E+00)† | 1.55E+01(4.91E+00)† | 1.24E+01(5.02E+00) |
| | 25 | 1.56E+02(8.53E+00)† | 1.15E+03(3.19E+02)† | 2.97E+00(9.20E-01)† | 4.24E+00(7.54E-01)† | 1.27E+00(4.25E-01) |
| | 50 | 1.39E+02(1.07E+00)† | 1.17E+03(3.08E+02)† | 7.22E-01(1.03E-01)† | 1.32E+00(2.20E-01)† | 2.28E-01(2.72E-02) |
| | 100 | 1.21E+02(1.98E+01)† | 1.18E+03(4.07E+02)† | 2.13E-01(2.47E-02)† | 5.65E-01(3.34E-02)† | 7.99E-02(5.01E-03) |
| | 200 | 1.12E+02(6.00E+00)† | 1.19E+03(2.73E+02)† | 8.66E-02(1.14E-02)† | 2.59E-01(1.61E-02)† | 6.11E-02(6.50E-03) |
| F2 | 5 | 4.20E-01(1.62E-05)† | 7.59E+02(4.61E+01)† | 7.31E-02(1.22E-05)† | 6.21E-02(1.15E-05)‡ | 7.15E-02(9.83E-06) |
| | 25 | 3.51E-01(8.04E-05)† | 7.56E+02(1.08E+02)† | 4.11E-02(3.56E-06)† | 3.70E-02(3.80E-06)‡ | 3.95E-02(1.80E-06) |
| | 50 | 3.04E-01(4.28E-06)† | 7.58E+02(1.43E+02)† | 3.11E-02(8.11E-07)† | 2.94E-02(1.13E-06)‡ | 3.00E-02(1.17E-06) |
| | 100 | 3.00E-01(9.68E-05)† | 7.71E+02(1.99E+02)† | 2.59E-02(4.17E-07)† | 2.54E-02(3.18E-07)‡ | 2.54E-02(4.11E-07) |
| | 200 | 2.90E-01(5.35E-05)† | 7.85E+02(4.32E+01)† | 2.43E-02(1.67E-07)† | 2.43E-02(9.29E-08)† | 2.42E-02(2.31E-07) |
| F3 | 5 | 3.97E+02(1.94E+01)† | 1.01E+03(3.58E+02)† | 3.44E+01(1.29E+01)† | 3.89E+01(2.26E+01)† | 2.90E+01(2.12E+01) |
| | 25 | 3.18E+02(2.30E+01)† | 8.65E+02(1.54E+02)† | 1.03E+01(6.92E+00)† | 1.14E+01(6.93E+00)† | 4.19E+00(2.76E+00) |
| | 50 | 2.88E+02(8.55E+00)† | 8.31E+02(1.13E+02)† | 2.42E+00(1.50E+00)† | 3.58E+00(3.26E+00)† | 6.22E-01(2.48E-01) |
| | 100 | 2.64E+02(1.01E+01)† | 8.01E+02(1.79E+02)† | 5.95E-01(1.31E-01)† | 1.40E+00(3.85E-01)† | 1.59E-01(2.34E-02) |
| | 200 | 2.51E+02(4.86E+00)† | 7.97E+02(1.34E+02)† | 2.06E-01(3.86E-02)† | 7.62E-01(1.27E-01)† | 1.09E-01(1.13E-02) |
| F4 | 5 | 6.26E-01(7.44E-05)† | 1.55E-01(1.06E-03)† | 1.06E-01(1.58E-04)† | 9.27E-02(8.72E-04)† | 8.89E-02(1.33E-04) |
| | 25 | 4.64E-01(7.49E-05)† | 4.62E-02(5.46E-05)† | 3.50E-02(7.05E-06)† | 2.32E-02(1.39E-05)‡ | 3.17E-02(1.20E-05) |
| | 50 | 4.14E-01(1.02E-05)† | 3.93E-02(5.17E-05)† | 2.58E-02(6.61E-07)† | 1.91E-02(1.57E-05)‡ | 2.49E-02(1.70E-06) |
| | 100 | 3.95E-01(1.48E-05)† | 3.73E-02(4.66E-05)† | 2.40E-02(2.21E-07)† | 1.87E-02(1.50E-05)‡ | 2.38E-02(3.44E-07) |
| | 200 | 3.78E-01(2.11E-06)† | 3.77E-02(4.77E-05)† | 2.46E-02(1.65E-07) | 1.89E-02(1.69E-05)‡ | 2.47E-02(2.67E-07) |
| WFG1 | 5 | 3.11E-01(9.06E-06)† | 9.57E+02(6.98E+02)† | 2.11E-01(1.33E-05)† | 2.08E-01(2.80E-05)† | 2.06E-01(1.55E-05) |
| | 25 | 2.64E-01(1.12E-05)† | 9.66E+02(1.11E+03)† | 1.87E-01(4.28E-05)† | 1.77E-01(8.31E-05)† | 1.60E-01(4.63E-05) |
| | 50 | 2.40E-01(1.14E-06)† | 9.72E+02(7.29E+02)† | 1.52E-01(2.87E-05)† | 1.50E-01(2.97E-05)† | 1.46E-01(2.37E-05) |
| | 100 | 2.22E-01(9.46E-06)† | 9.73E+02(3.22E+02)† | 1.43E-01(1.97E-05)† | 1.43E-01(3.33E-05)† | 1.41E-01(1.62E-05) |
| | 200 | 2.20E-01(2.04E-06)† | 9.76E+02(2.16E+02)† | 1.36E-01(1.35E-05)† | 1.35E-01(3.03E-05) | 1.35E-01(2.24E-05) |
| WFG2 | 5 | 5.86E-01(2.70E-05)† | 1.22E+00(3.38E-04)† | 3.34E-01(1.20E-04)† | 3.25E-01(2.19E-04)† | 3.14E-01(1.76E-04) |
| | 25 | 4.75E-01(1.49E-04)† | 1.20E+00(4.17E-05)† | 3.20E-01(1.15E-04)† | 3.18E-01(8.12E-05)† | 3.18E-01(6.60E-05) |
| | 50 | 5.00E-01(1.03E-04)† | 1.20E+00(2.17E-05)† | 3.23E-01(6.83E-05)† | 3.23E-01(9.04E-05)† | 3.21E-01(7.74E-05) |
| | 100 | 5.06E-01(4.83E-05)† | 1.20E+00(2.18E-05)† | 3.23E-01(8.99E-05)† | 3.22E-01(7.39E-05)† | 3.21E-01(9.13E-05) |
| | 200 | 5.02E-01(5.93E-05)† | 1.20E+00(1.31E-05)† | 3.17E-01(9.22E-05)† | 3.15E-01(4.86E-05)† | 3.11E-01(9.97E-05) |
| WFG3 | 5 | 1.76E+00(5.08E-04)† | 1.33E+00(8.46E-05)‡ | 1.57E+00(1.43E-03)† | 1.35E+00(2.60E-03)‡ | 1.53E+00(2.25E-03) |
| | 25 | 1.58E+00(4.60E-05)† | 1.30E+00(1.20E-05)‡ | 1.38E+00(1.05E-03)† | 1.31E+00(1.38E-03)‡ | 1.37E+00(2.24E-03) |
| | 50 | 1.57E+00(2.31E-05)† | 1.30E+00(1.25E-05)‡ | 1.36E+00(6.98E-04)† | 1.31E+00(1.13E-03)‡ | 1.34E+00(1.01E-03) |
| | 100 | 1.56E+00(6.97E-06)† | 1.30E+00(9.73E-06)‡ | 1.36E+00(2.52E-04)† | 1.32E+00(5.21E-04)‡ | 1.33E+00(7.04E-04) |
| | 200 | 1.59E+00(3.59E-07)† | 1.30E+00(1.03E-05)‡ | 1.35E+00(3.12E-04)† | 1.33E+00(3.62E-04)† | 1.32E+00(4.34E-04) |
| WFG4 | 5 | 4.27E-01(1.57E-05)† | 8.77E-01(4.59E-05)† | 1.86E-01(8.75E-06)† | 1.84E-01(8.22E-06)† | 1.76E-01(8.38E-06) |
| | 25 | 3.63E-01(2.11E-05)† | 8.29E-01(4.28E-05)† | 1.57E-01(4.36E-06)† | 1.55E-01(3.05E-06)† | 1.55E-01(6.55E-06) |
| | 50 | 3.46E-01(6.97E-06)† | 8.08E-01(2.81E-05)† | 1.53E-01(7.82E-06)† | 1.50E-01(4.19E-06)† | 1.50E-01(3.40E-06) |
| | 100 | 3.26E-01(2.49E-06)† | 8.02E-01(1.27E-05)† | 1.50E-01(5.46E-06)† | 1.47E-01(5.50E-06)† | 1.46E-01(4.30E-06) |
| | 200 | 3.19E-01(1.21E-06)† | 8.03E-01(3.64E-06)† | 1.50E-01(4.12E-06)† | 1.50E-01(4.35E-06)† | 1.48E-01(4.65E-06) |
| WFG5 | 5 | 4.45E-01(2.73E-05)† | 1.87E-01(2.77E-05)‡ | 2.17E-01(7.03E-06)† | 2.09E-01(3.82E-06)‡ | 2.15E-01(4.52E-06) |
| | 25 | 3.91E-01(8.78E-06)† | 1.56E-01(4.65E-05)‡ | 1.83E-01(5.54E-06)† | 1.84E-01(3.30E-06)† | 1.82E-01(6.32E-06) |
| | 50 | 3.53E-01(2.05E-06)† | 1.35E-01(7.38E-05)‡ | 1.78E-01(4.28E-06) | 1.81E-01(4.26E-06)† | 1.78E-01(3.23E-06) |
| | 100 | 3.42E-01(5.17E-06)† | 1.34E-01(5.31E-05)‡ | 1.76E-01(3.14E-06) | 1.78E-01(4.26E-06)† | 1.76E-01(5.19E-06) |
| | 200 | 3.32E-01(9.54E-06)† | 1.37E-01(2.04E-05)‡ | 1.75E-01(4.99E-06)‡ | 1.76E-01(4.49E-06) | 1.76E-01(4.44E-06) |
| WFG6 | 5 | 4.80E-01(4.54E-05)† | 7.73E-01(4.17E-04)† | 2.99E-01(1.94E-05)† | 2.94E-01(8.82E-04)† | 2.83E-01(3.79E-05) |
| | 25 | 3.60E-01(1.34E-05)† | 7.33E-01(1.05E-03)† | 1.74E-01(1.14E-05)† | 1.95E-01(9.94E-04)† | 1.68E-01(5.83E-06) |
| | 50 | 3.29E-01(2.21E-05)† | 7.30E-01(1.17E-03)† | 1.60E-01(7.48E-06)† | 1.83E-01(9.63E-04)† | 1.59E-01(4.42E-06) |
| | 100 | 3.16E-01(1.61E-05)† | 7.31E-01(1.15E-03)† | 1.57E-01(5.68E-06)† | 1.78E-01(9.27E-04)† | 1.56E-01(6.76E-06) |
| | 200 | 3.08E-01(1.81E-05)† | 7.32E-01(1.12E-03)† | 1.57E-01(4.36E-06) | 1.77E-01(9.17E-04)† | 1.57E-01(5.77E-06) |
| WFG7 | 5 | 4.98E-01(2.34E-05)† | 2.89E-01(7.72E-05)† | 2.19E-01(1.14E-05)† | 2.23E-01(1.53E-05)† | 2.12E-01(1.67E-05) |
| | 25 | 3.78E-01(7.20E-06)† | 1.49E-01(6.10E-05)‡ | 1.61E-01(8.61E-06)† | 1.61E-01(5.89E-06)† | 1.60E-01(6.67E-06) |
| | 50 | 3.65E-01(1.97E-06)† | 1.19E-01(8.60E-05)‡ | 1.56E-01(4.10E-06)† | 1.55E-01(5.62E-06) | 1.55E-01(5.72E-06) |
| | 100 | 3.54E-01(1.34E-05)† | 1.17E-01(7.81E-05)‡ | 1.49E-01(1.05E-05) | 1.49E-01(9.25E-06) | 1.49E-01(9.70E-06) |
| | 200 | 3.49E-01(3.46E-06)† | 1.31E-01(3.66E-05)‡ | 1.50E-01(5.45E-06)† | 1.50E-01(1.09E-05)† | 1.49E-01(1.16E-05) |
| WFG8 | 5 | 4.96E-01(1.44E-06)† | 2.89E-01(9.30E-05)† | 2.23E-01(1.59E-05)† | 2.18E-01(1.43E-05)† | 2.12E-01(1.29E-05) |
| | 25 | 3.88E-01(4.25E-06)† | 1.49E-01(3.03E-05)‡ | 1.61E-01(4.09E-06)† | 1.60E-01(5.26E-06)† | 1.59E-01(5.73E-06) |
| | 50 | 3.73E-01(1.41E-05)† | 1.18E-01(6.72E-05)‡ | 1.56E-01(4.20E-06)† | 1.55E-01(4.14E-06) | 1.55E-01(4.91E-06) |
| | 100 | 3.51E-01(1.90E-06)† | 1.17E-01(5.28E-05)‡ | 1.50E-01(7.45E-06)† | 1.50E-01(9.06E-06)† | 1.49E-01(1.17E-05) |
| | 200 | 3.46E-01(1.36E-05)† | 1.33E-01(2.90E-05)‡ | 1.50E-01(8.23E-06)† | 1.48E-01(1.16E-05) | 1.48E-01(1.25E-05) |
| WFG9 | 5 | 5.31E-01(2.74E-05) | 3.93E-01(2.50E-02) | 6.34E-01(2.40E-02)† | 7.75E-01(1.98E-04)† | 5.02E-01(3.32E-02) |
| | 25 | 3.83E-01(5.03E-05) | 2.35E-01(2.07E-02) | 5.62E-01(3.34E-02)† | 7.52E-01(3.17E-04)† | 4.20E-01(4.21E-02) |
| | 50 | 3.52E-01(2.20E-05)‡ | 2.07E-01(1.81E-02)‡ | 5.48E-01(3.28E-02)† | 7.45E-01(3.46E-04)† | 4.10E-01(4.12E-02) |
| | 100 | 3.52E-01(4.38E-06)‡ | 1.89E-01(1.62E-02)‡ | 5.43E-01(3.24E-02)† | 7.40E-01(3.55E-04)† | 4.05E-01(4.10E-02) |
| | 200 | 3.56E-01(1.67E-05)‡ | 1.78E-01(1.55E-02)‡ | 5.41E-01(3.18E-02)† | 7.40E-01(3.59E-04)† | 4.04E-01(4.03E-02) |
| +-= | | 60/3/2 | 42/21/2 | 59/1/5 | 46/13/6 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.




(a) F4 (b) WFG3

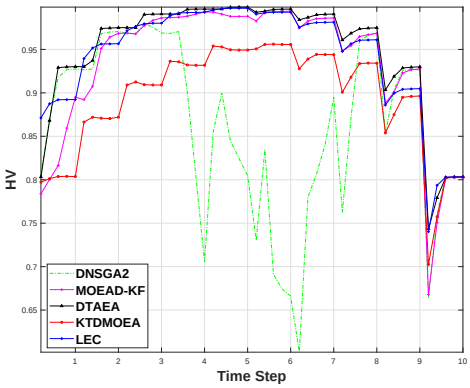
(c) WFG6 (d) WFG9

Figure 6.16: HV trajectory of sampled generations (1, 5, 25 and 50) after each change over evolution on problems F4, WFG3, WFG6 and WFG9, when $\tau_t = 50$.

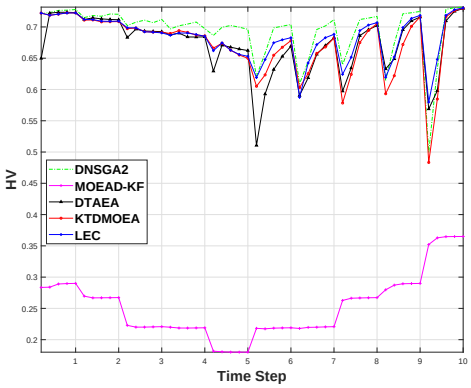
and WFG9) are drawn. HV trajectory of sampled generations (1, 5, 25, 100 and 200) after each change over evolution on these four problems for three different $\tau_t$s are presented in Figures 6.16-6.18, respectively.

For F4, it is clear from Figures 6.16(a), 6.17(a) and 6.18(a) that in the first generation right after changes, LEC gets better results than DTAEA and KTDMOEA when increasing the NObj, which further shows that our proposed LEC indeed improves diversity compared to KTDMOEA. The reason why LEC gets worse HV than DTAEA after optimization of different generations might be that the diversity provided by DA in DTAEA is much more than that of LEC.

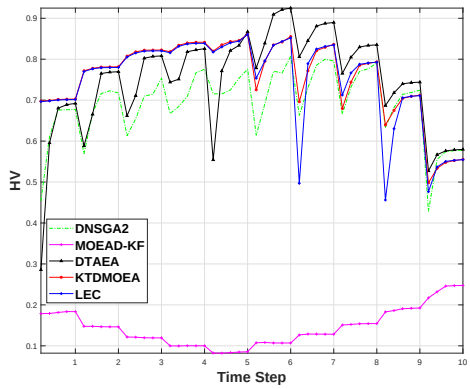
For WFG3 with degenerated PF, it is clear from Figures 6.16(b), 6.17(b) and 6.18(b) that our proposed LEC is unable to provide enough diversity right after changes. In the optimization process, LEC leverages the evenly distributed weight vectors to maintain the diversity. However, for WFG3 with degenerated PF, the evenly distributed weight vectors cover the other areas away from the degenerated PF. Therefore, solutions are unable to always locate at the true PF, thus resulting in worse diversity. It is clear from Figures 1-3 that DNSGA2 is best at all sampled



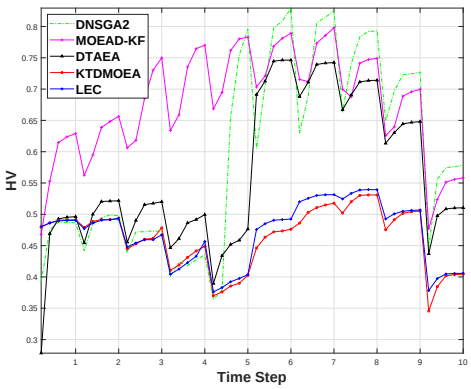

(a) F4




(b) WFG3




(c) WFG6



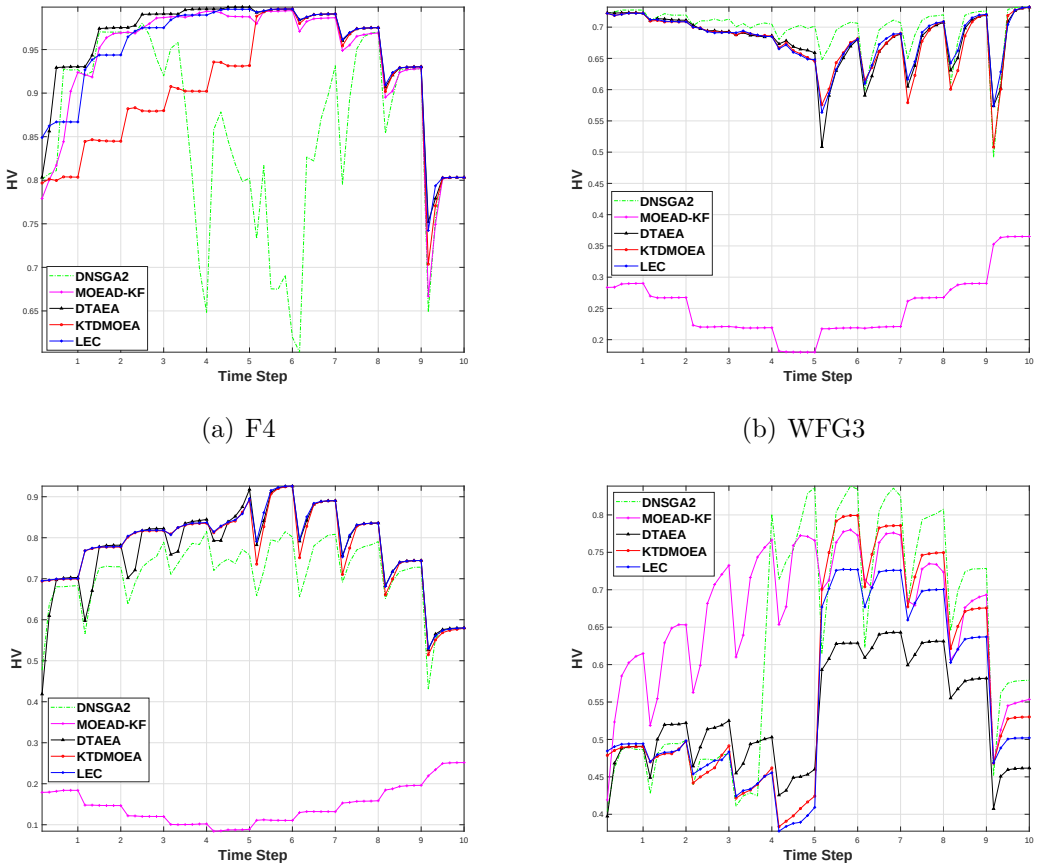

(d) WFG9

Figure 6.17: HV trajectory of sampled generations (1, 5, 25, 50 and 100) after each change over evolution on problems F4, WFG3, WFG6 and WFG9, when $\tau_t = 100$.

generations of all changes, which shows that decomposition-based algorithms might not be suitable for solving problems with degenerated PS.

For WFG6, a problem with nonseparable variables, algorithms are required to have much diversity. Therefore, it is clear from Figures 6.16(c), 6.17(c) and 6.18(c) that KTDMOEA and LEC get better HV than DTAEA while this trend inverses after some generations' optimization. When decreasing the NObj, when $\tau_t$ is small, LEC gets worse HV than DTAEA at all cases. The reason might be that when $\tau_t$ is small, the contraction directions are not such accurate, resulting in worse transferred solutions.

For WFG9, it is clear from Figures 6.16(d), 6.17(d) and 6.18(d) that our proposed LEC and KTDMOEA are the worst among all compared algorithms. The reason is analyzed as follows. For WFG9 with extremely complex problem features, its optimization requires much diversity or search operators with strong search ability. However, neither LEC or KTDMOEA has no strong search operator or no much diversity maintenance. While for other algorithms, DNSGA2 and DTAEA has enough diversity introduction right after changes. The DE operator in MOEAD-KF is able




(a) F4 (b) WFG3

(c) WFG6 (d) WFG9

Figure 6.18: HV trajectory of sampled generations (1, 5, 25, 50, 100 and 200) after each change over evolution on problems F4, WFG3, WFG6 and WFG9, when $\tau_t = 200$.

to help solutions to reach the global optimum.

### 6.4.3 Performance Comparison on Other Changes in the NObj

In the previous experiments, the NObj only increases or decreases one by one. This section aims to verify the performance of the proposed algorithm in the scenario where the NObj increases or decreases by more than one. The new changing sequence where the NObj increases or decreases by two each time is designed as follows:

$$m(t) = \begin{cases} 2, & t{=}0 \\ m(t-1)+2, & t \in [1,2] \\ m(t-1)-2, & t \in [3,4], \end{cases} \tag{6.2}$$

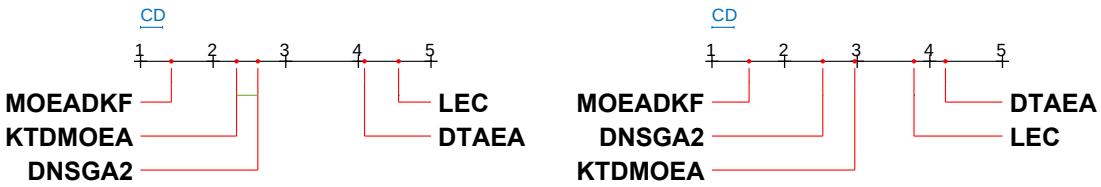
where $t$ is the identifier of the change. In this sequence of change, there are two NObj increases, i.e. NObj increasing from 2 to 4 and then from 4 to 6. Later on, there are two changes where NObj decreases from 6 to 4 and then from 4 to 2.

Experimental settings including test problems and compared algorithms are set as the same to those in Section 6.3.3. Here, both the frequency of change and *gap* are set as 200. Besides, only MHV is used to measure the quality of optimized solutions at the last generation to save spaces. The used statistical tests are the same as those in paragraphs above Section 6.4.1.

Figures 6.19 presents the Nemenyi post-tests results among MHV values of obtained solution at the first generation after changes and optimized solutions at the last generation by all compared algorithms. Friedman detects significant differences in average value for MHV of obtained solutions at the first generation after changes and the last generation after optimization with a p-value of 9.1617e-232 and 1.0503e-156, respectively. Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 6 and then decreasing it from 6 to 2, both two by two. is presented in Table 6.28.

Overall, it can be observed from Figures 6.19, our proposed LEC performs significantly best among all compared algorithms at the first generation right after changes. This result shows that our proposed LEC is able to achieve better results right after changes when NObj changes more than one.

To explore the reason why LEC gets significantly worse results after optimization when the NObj changes by two every time, HV trajectory of sampled generations over evolution on one problem F3 is drawn in Figure 6.20 to see in which stage LEC performs worse. It is clear from Figure 6.20 that when decreasing the NObj, LEC gets worse HV results than DTAEA. And the contracted solutions by LEC is worse that reconstructed solutions by DTAEA. The reason might be that even though LEC finds most promising contraction directions, it is still almost impossible to some solutions to reach the true PS after decreasing the NObj. For example, suppose the true PS of a problem with four objectives is a 3-dimension cube and the




(a) At the first generation after changes (b) At the last generation after opt

Figure 6.19: Parallel coordinates of solutions obtained at the first generation after increasing the NObj from 2 to 7 on 7-objective WFG4.

true PS of a problem with two objectives is a 1-dimension line segment in the cube. When decreasing the NObj from 4 to 2, solutions filled in the cube need to move along different directions from their original position to reach the line segment.

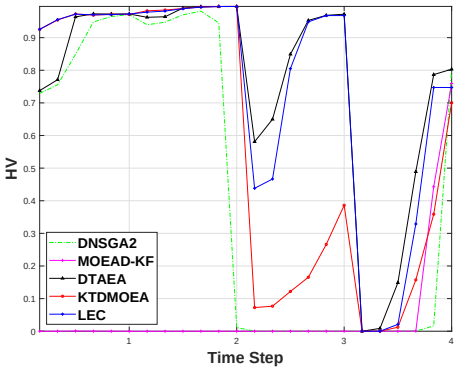



Figure 6.20: HV trajectory of sampled generations (1, 5, 25, 100 and 200) after each change over evolution on one problem F3.

### 6.4.4 Impact of Algorithm Parameters

In the process of generating solutions along the learnt contraction directions when decreasing NObj, there is a parameter $\lambda$ to set which is the number of selected solutions to generate an initialized population. Different values of this parameter will be set to verify whether different parameter settings affect the performance of LEC against other compared algorithms.

Experimental settings including test problems and compared algorithms are set as the same to those in Section 6.3.3. Here, both the frequency of change and gap are set as 200. Besides, only MHV is used to measure the quality of optimized solutions at the last generation to save spaces. The changing sequence is that the NObj decreases from 7 to 2 one by one. There are three LECs (denoted as LEC-10, LEC-20 and LEC-30), which has the value of $\lambda$ as $popsize/10$, $popsize/20$ and $popsize/30$, respectively.

In order to verify whether different parameter settings affect the performance of LEC, the Friedman and Nemenyi tests on 4 state-of-the-arts and 3 LECs are conducted to indicate the significant differences among them. The setting of the Friedman and Nemenyi tests are the same as those in paragraphs above Section 6.4.1.

Figure 5.20 presents the Nemenyi post-tests results among MHV values of optimized solutions at the last generation by 7 algorithms. Friedman detects significant differences in average MHV value with a p-value of 3.8832e-244.

It is clear that the three LECs get the best or equally best MHV values among all

Table 6.28: Mean and standard deviation values of MHV metric for optimized solutions obtained by all compared algorithms after *gap* generations' optimization on all DMOPs with a changing NObj, when $\tau_t = 200$, when firstly increasing NObj from 2 to 6 and then decreasing it from 6 to 2, both two by two.

| Prob. | *gap* | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|---|
| F1 | 5 | 4.93E-01(2.92E-09)† | 2.57E-02(6.67E-04)† | 7.13E-01(2.34E-03) | 5.73E-01(8.55E-03)† | 7.07E-01(2.65E-03) |
| | 25 | 4.97E-01(5.20E-08)† | 1.09E-01(3.05E-03)† | 8.34E-01(5.44E-03) | 6.38E-01(1.65E-02)† | 8.05E-01(3.08E-03) |
| | 50 | 6.23E-01(5.45E-04)† | 1.52E-01(4.29E-03)† | 9.44E-01(3.99E-03) | 7.83E-01(3.99E-02)† | 9.42E-01(2.36E-03) |
| | 100 | 7.06E-01(4.69E-04)† | 3.77E-01(7.21E-03)† | 9.89E-01(7.44E-05) | 8.59E-01(4.36E-02)† | 9.91E-01(4.95E-07) |
| | 200 | 9.03E-01(2.22E-03)† | 4.33E-01(4.08E-03)† | 9.92E-01(6.75E-09) | 9.16E-01(1.46E-02) | 9.92E-01(5.75E-08) |
| F2 | 5 | 9.01E-01(3.02E-07)† | 4.49E-01(8.21E-04)† | 9.16E-01(1.65E-05)† | 8.17E-01(3.47E-03)† | 9.29E-01(1.59E-06) |
| | 25 | 9.27E-01(5.31E-08)† | 4.92E-01(6.06E-04)† | 9.36E-01(6.82E-09)‡ | 9.01E-01(9.76E-04)† | 9.36E-01(1.83E-07) |
| | 50 | 9.20E-01(2.31E-06)† | 4.62E-01(2.60E-07)† | 9.37E-01(2.79E-09)† | 9.33E-01(2.77E-05)† | 9.37E-01(4.39E-09) |
| | 100 | 9.16E-01(2.31E-07)† | 4.62E-01(2.78E-07)† | 9.37E-01(9.62E-10)† | 9.37E-01(3.85E-07)† | 9.37E-01(1.29E-08) |
| | 200 | 9.13E-01(3.50E-07)† | 4.62E-01(2.67E-07)† | 9.37E-01(6.15E-10)† | 9.37E-01(6.20E-09)† | 9.37E-01(5.53E-10) |
| F3 | 5 | 4.26E-01(1.32E-06)† | 0.00E+00(0.00E+00)† | 5.98E-01(5.13E-03) | 5.04E-01(2.07E-03)† | 6.01E-01(4.23E-03) |
| | 25 | 4.55E-01(2.94E-06)† | 1.60E-04(7.96E-07)† | 7.38E-01(5.08E-03)‡ | 5.24E-01(4.10E-03)† | 6.97E-01(1.08E-03) |
| | 50 | 4.82E-01(5.90E-06)† | 1.04E-04(3.06E-07)† | 8.52E-01(4.83E-03)‡ | 5.71E-01(1.74E-02)† | 8.10E-01(4.95E-03) |
| | 100 | 4.82E-01(4.00E-04)† | 1.11E-01(5.71E-03)† | 9.31E-01(4.52E-04)‡ | 6.48E-01(3.53E-02)† | 9.20E-01(1.36E-03) |
| | 200 | 4.46E-01(2.43E-04)† | 1.90E-01(1.79E-03)† | 9.36E-01(1.35E-06) | 7.63E-01(2.23E-02)† | 9.35E-01(2.17E-06) |
| F4 | 5 | 8.30E-01(2.02E-05) | 8.68E-01(1.73E-04) | 9.09E-01(3.46E-05)‡ | 5.86E-01(1.47E-03)† | 7.29E-01(3.60E-02) |
| | 25 | 9.22E-01(6.28E-07)‡ | 9.17E-01(1.50E-04)‡ | 9.37E-01(7.02E-09)‡ | 6.53E-01(2.15E-04) | 7.50E-01(3.32E-02) |
| | 50 | 9.08E-01(1.24E-06)‡ | 9.28E-01(2.52E-05)‡ | 9.37E-01(2.85E-09)‡ | 6.53E-01(2.13E-04) | 7.50E-01(3.32E-02) |
| | 100 | 8.84E-01(6.57E-06) | 9.33E-01(4.34E-06)‡ | 9.37E-01(1.58E-09)‡ | 6.54E-01(3.03E-04) | 7.50E-01(3.33E-02) |
| | 200 | 8.71E-01(2.17E-05) | 9.34E-01(3.24E-08)‡ | 9.38E-01(1.70E-10)‡ | 6.56E-01(3.16E-04) | 7.51E-01(3.34E-02) |
| WFG1 | 5 | 8.79E-01(2.83E-07)† | 3.07E-01(1.99E-05)† | 8.80E-01(1.72E-05)† | 7.83E-01(9.60E-04)† | 9.04E-01(7.90E-06) |
| | 25 | 9.19E-01(1.52E-07)† | 3.03E-01(3.72E-05)† | 9.21E-01(7.99E-06)† | 8.49E-01(1.28E-03)† | 9.28E-01(5.14E-06) |
| | 50 | 9.30E-01(4.06E-07)† | 3.02E-01(6.14E-05)† | 9.37E-01(2.38E-06)† | 9.01E-01(7.43E-04)† | 9.38E-01(3.13E-06) |
| | 100 | 9.37E-01(1.86E-07)† | 3.03E-01(5.83E-05)† | 9.44E-01(3.59E-07) | 9.31E-01(1.02E-04)† | 9.44E-01(9.84E-07) |
| | 200 | 9.42E-01(7.15E-10)† | 3.03E-01(3.97E-05)† | 9.46E-01(1.22E-07)‡ | 9.43E-01(4.43E-06)† | 9.46E-01(1.00E-06) |
| WFG2 | 5 | 8.82E-01(3.67E-07)† | 2.65E-01(5.14E-07)† | 9.14E-01(7.79E-06)‡ | 7.04E-01(1.57E-04)† | 9.11E-01(1.92E-05) |
| | 25 | 9.33E-01(7.38E-09)† | 2.66E-01(3.99E-09)† | 9.37E-01(1.32E-07) | 7.98E-01(7.88E-04)† | 9.37E-01(1.24E-07) |
| | 50 | 9.35E-01(2.78E-09)† | 2.66E-01(3.55E-10)† | 9.38E-01(1.06E-08) | 8.72E-01(9.03E-04)† | 9.38E-01(1.84E-08) |
| | 100 | 9.36E-01(2.90E-09)† | 2.66E-01(5.04E-11)† | 9.38E-01(4.97E-09) | 9.29E-01(4.32E-04)† | 9.38E-01(7.57E-09) |
| | 200 | 9.36E-01(2.30E-09)† | 2.66E-01(5.17E-11)† | 9.38E-01(1.29E-08) | 9.38E-01(1.40E-08)† | 9.38E-01(1.53E-08) |
| WFG3 | 5 | 6.58E-01(3.73E-08) | 2.75E-01(8.40E-07)† | 6.78E-01(2.28E-04)‡ | 5.35E-01(5.26E-04)† | 6.58E-01(1.13E-04) |
| | 25 | 7.13E-01(4.44E-08)‡ | 2.79E-01(1.28E-08)† | 7.04E-01(1.57E-05)‡ | 6.20E-01(3.07E-04)† | 6.90E-01(4.98E-05) |
| | 50 | 7.16E-01(1.06E-07)‡ | 2.79E-01(6.51E-09)† | 7.08E-01(5.06E-06)‡ | 6.56E-01(4.86E-04)† | 6.96E-01(2.02E-05) |
| | 100 | 7.18E-01(3.23E-07)‡ | 2.80E-01(2.25E-09)† | 7.09E-01(1.79E-06)‡ | 6.90E-01(6.33E-05)† | 7.00E-01(2.20E-05) |
| | 200 | 7.19E-01(3.18E-08)‡ | 2.80E-01(9.04E-10)† | 7.10E-01(9.71E-07)‡ | 6.98E-01(5.94E-05)† | 7.05E-01(4.34E-05) |
| WFG4 | 5 | 6.38E-01(6.68E-06)† | 2.85E-01(6.69E-05)† | 7.03E-01(3.83E-05)† | 6.29E-01(2.26E-03)† | 7.20E-01(4.16E-06) |
| | 25 | 7.16E-01(3.06E-06)† | 3.44E-01(1.87E-05)† | 7.87E-01(9.21E-07) | 7.61E-01(4.47E-04)† | 7.83E-01(4.52E-05) |
| | 50 | 7.12E-01(8.73E-07)† | 3.54E-01(9.68E-06)† | 7.93E-01(6.45E-08)† | 7.88E-01(1.81E-05)† | 7.94E-01(1.27E-07) |
| | 100 | 7.25E-01(2.08E-06)† | 3.52E-01(2.73E-05)† | 7.95E-01(3.51E-08)† | 7.93E-01(7.75E-07)† | 7.95E-01(5.20E-08) |
| | 200 | 7.29E-01(5.61E-07)† | 3.50E-01(1.81E-06)† | 7.95E-01(3.18E-08)‡ | 7.94E-01(7.49E-08)† | 7.95E-01(8.84E-08) |
| WFG5 | 5 | 6.42E-01(5.02E-06)† | 5.00E-01(5.60E-05)† | 6.90E-01(2.16E-05)† | 5.81E-01(1.03E-04)† | 7.08E-01(1.72E-05) |
| | 25 | 6.81E-01(3.92E-06)† | 5.41E-01(8.11E-06)† | 7.52E-01(8.91E-07)‡ | 7.33E-01(4.85E-05)† | 7.37E-01(1.14E-05) |
| | 50 | 6.91E-01(1.59E-06)† | 5.41E-01(1.04E-05)† | 7.57E-01(1.22E-06)‡ | 7.50E-01(4.02E-05)† | 7.53E-01(2.07E-05) |
| | 100 | 6.97E-01(1.69E-06)† | 5.40E-01(1.06E-05)† | 7.60E-01(7.28E-07)‡ | 7.56E-01(1.28E-06)† | 7.59E-01(7.66E-07) |
| | 200 | 6.96E-01(2.42E-06)† | 5.39E-01(9.56E-06)† | 7.61E-01(7.40E-07) | 7.57E-01(3.46E-07)† | 7.61E-01(4.63E-07) |
| WFG6 | 5 | 5.89E-01(4.10E-05)† | 1.55E-01(1.73E-06)† | 6.26E-01(8.00E-05)† | 5.12E-01(7.95E-04)† | 6.94E-01(1.81E-04) |
| | 25 | 6.70E-01(5.25E-05)† | 1.58E-01(5.83E-06)† | 7.32E-01(1.29E-04) | 5.76E-01(2.09E-03)† | 7.31E-01(2.20E-04) |
| | 50 | 6.86E-01(3.47E-05)† | 1.58E-01(1.02E-05)† | 7.40E-01(1.35E-04) | 6.38E-01(3.80E-03)† | 7.38E-01(2.61E-04) |
| | 100 | 7.06E-01(1.26E-05)† | 1.58E-01(1.20E-05)† | 7.46E-01(1.51E-04) | 6.91E-01(3.92E-03)† | 7.41E-01(2.45E-04) |
| | 200 | 7.09E-01(1.66E-05)† | 1.59E-01(1.41E-05)† | 7.55E-01(2.31E-04)‡ | 6.97E-01(4.01E-03)† | 7.43E-01(2.46E-04) |
| WFG7 | 5 | 6.63E-01(4.09E-06)† | 6.04E-01(1.76E-04)† | 7.21E-01(2.04E-05)† | 6.01E-01(2.15E-04)† | 7.61E-01(6.77E-06) |
| | 25 | 7.40E-01(2.58E-06)† | 6.96E-01(4.93E-05)† | 7.91E-01(4.41E-07) | 7.81E-01(4.68E-05)† | 7.91E-01(5.37E-07) |
| | 50 | 7.50E-01(1.61E-06)† | 7.17E-01(3.87E-05)† | 7.94E-01(2.10E-07)‡ | 7.92E-01(3.07E-06)† | 7.93E-01(2.54E-07) |
| | 100 | 7.55E-01(4.05E-08)† | 7.28E-01(2.99E-05)† | 7.94E-01(2.47E-08)‡ | 7.94E-01(1.84E-07)† | 7.94E-01(7.23E-08) |
| | 200 | 7.58E-01(8.60E-07)† | 7.40E-01(8.32E-06)† | 7.95E-01(1.01E-08)† | 7.95E-01(5.71E-08)† | 7.95E-01(8.88E-09) |
| WFG8 | 5 | 6.75E-01(1.51E-06)† | 6.06E-01(2.12E-04)† | 7.21E-01(2.94E-05)† | 6.01E-01(1.91E-04)† | 7.62E-01(4.69E-06) |
| | 25 | 7.41E-01(7.39E-07)† | 6.97E-01(5.18E-05)† | 7.91E-01(4.60E-07)‡ | 7.79E-01(7.51E-05)† | 7.90E-01(3.86E-07) |
| | 50 | 7.49E-01(1.80E-06)† | 7.17E-01(2.15E-05)† | 7.94E-01(1.13E-07)‡ | 7.91E-01(3.93E-06)† | 7.93E-01(2.85E-07) |
| | 100 | 7.52E-01(4.11E-09)† | 7.29E-01(2.32E-05)† | 7.94E-01(2.90E-08)‡ | 7.94E-01(1.79E-07)† | 7.94E-01(6.66E-08) |
| | 200 | 7.56E-01(5.88E-07)† | 7.40E-01(1.21E-05)† | 7.95E-01(9.21E-09)† | 7.95E-01(5.22E-08)† | 7.95E-01(7.70E-09) |
| WFG9 | 5 | 5.33E-01(5.87E-05)‡ | 5.96E-01(1.69E-02)‡ | 4.59E-01(8.46E-06)‡ | 4.17E-01(1.15E-03)† | 4.38E-01(4.07E-04) |
| | 25 | 5.68E-01(2.92E-06)‡ | 6.72E-01(7.73E-03)‡ | 4.86E-01(2.62E-05)‡ | 5.35E-01(3.06E-03)‡ | 4.47E-01(6.20E-04) |
| | 50 | 5.71E-01(5.53E-07)‡ | 6.85E-01(5.99E-03)‡ | 4.89E-01(2.80E-05)‡ | 5.44E-01(3.38E-03)‡ | 4.50E-01(5.94E-04) |
| | 100 | 5.88E-01(1.38E-05)‡ | 6.93E-01(4.54E-03)‡ | 4.91E-01(3.71E-05)‡ | 5.53E-01(3.64E-03)‡ | 4.55E-01(5.81E-04) |
| | 200 | 6.69E-01(3.14E-06)‡ | 6.92E-01(3.74E-03)‡ | 4.92E-01(4.77E-05)‡ | 5.59E-01(3.39E-03)‡ | 4.64E-01(3.94E-04) |
| +-= | | 50/11/4 | 55/9/1 | 16/31/18 | 56/4/5 | – |

Wilcoxon Rank Sum test at a 0.05 significance level is performed between LEC and each compared algorithm. † and ‡ denote the performance of the corresponding algorithm is significantly worse than and better than that of LEC, respectively. The best mean value is highlighted with gray background.

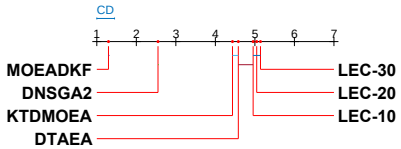



Figure 6.21: Friedman ranking among MHV of optimized solutions at the last generation by 6 state-of-the-arts and 3 LECs with different values of parameters theta (1, 2 and 4) in the changing sequence of firstly increasing the NObj from 2 to 7 and then decreasing it from 7 to 2, both one by one.

algorithms. It can be found from Figure 6.21 that there is no significant difference among the three LECs with different setting of the parameter $\lambda$. These results have verified that the performance of the proposed LEC approach is not sensitive to these setting of the parameter $\lambda$. The performance of the proposed LEC against the existing algorithms is not affected by the setting of different parameter values.

Table 6.29: Mean and standard deviation values of MHV metric for optimized solutions obtained by 4 state-of-the-arts and three LECs with different values of parameters ($\lambda = popsize/10$, $popsize/20$ and $popsize/30$) at the last generation of all changes on all DMOPs with a changing number of objectives, when firstly decreasing the number of objectives from 7 to 2 and then increasing it from 2 to 7, both one by one

| Problems | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC-10 | LEC-20 | LEC-30 |
|---|---|---|---|---|---|---|---|
| F1 | 3.93E-01(2.50E-10) | 1.17E-01(7.27E-03) | 9.93E-01(2.96E-08) | 9.93E-01(1.37E-07) | 9.93E-01(3.13E-09) | 9.93E-01(7.81E-09) | 9.93E-01(4.22E-09) |
| F2 | 9.03E-01(1.76E-05) | 7.71E-01(1.03E-05) | 9.39E-01(3.22E-10) | 9.39E-01(7.18E-10) | 9.39E-01(6.61E-10) | 9.39E-01(5.51E-10) | 9.39E-01(3.90E-10) |
| F3 | 3.46E-01(9.19E-09) | 8.53E-02(5.72E-03) | 9.36E-01(1.68E-05) | 9.36E-01(4.97E-06) | 9.37E-01(1.84E-06) | 9.38E-01(4.40E-07) | 9.37E-01(1.09E-06) |
| F4 | 8.50E-01(3.51E-06) | 9.36E-01(7.45E-09) | 9.39E-01(7.10E-10) | 9.39E-01(6.83E-10) | 9.39E-01(7.04E-10) | 9.39E-01(3.59E-10) | 9.39E-01(7.10E-10) |
| WFG1 | 9.48E-01(7.08E-08) | 3.22E-01(7.04E-06) | 9.52E-01(8.15E-08) | 9.52E-01(1.85E-07) | 9.52E-01(5.71E-08) | 9.52E-01(8.89E-08) | 9.52E-01(7.72E-08) |
| WFG2 | 9.42E-01(5.80E-09) | 2.59E-01(1.41E-09) | 9.44E-01(5.03E-09) | 9.44E-01(3.90E-09) | 9.45E-01(4.14E-09) | 9.44E-01(9.86E-09) | 9.45E-01(3.70E-09) |
| WFG3 | 7.18E-01(3.49E-08) | 2.73E-01(5.25E-09) | 7.06E-01(2.04E-06) | 7.06E-01(2.14E-06) | 7.07E-01(1.78E-06) | 7.06E-01(1.74E-06) | 7.06E-01(1.28E-06) |
| WFG4 | 7.37E-01(2.15E-06) | 3.54E-01(2.09E-06) | 7.95E-01(1.81E-08) | 7.96E-01(3.91E-08) | 7.95E-01(1.78E-08) | 7.95E-01(1.21E-08) | 7.95E-01(2.25E-08) |
| WFG5 | 7.12E-01(6.53E-06) | 5.44E-01(2.76E-06) | 7.71E-01(4.11E-08) | 7.67E-01(7.31E-06) | 7.71E-01(2.14E-08) | 7.71E-01(5.11E-08) | 7.71E-01(3.02E-08) |
| WFG6 | 7.41E-01(4.74E-06) | 1.70E-01(1.81E-05) | 7.95E-01(1.66E-07) | 7.95E-01(1.85E-07) | 7.95E-01(1.70E-07) | 7.95E-01(1.31E-07) | 7.95E-01(1.67E-07) |
| WFG7 | 7.54E-01(2.38E-06) | 7.44E-01(9.02E-06) | 7.96E-01(5.00E-08) | 7.96E-01(2.96E-08) | 7.96E-01(3.69E-08) | 7.96E-01(7.83E-08) | 7.96E-01(2.08E-08) |
| WFG8 | 7.57E-01(1.16E-06) | 7.45E-01(6.90E-06) | 7.96E-01(3.32E-08) | 7.96E-01(2.75E-08) | 7.96E-01(3.81E-08) | 7.96E-01(3.31E-08) | 7.96E-01(2.11E-08) |
| WFG9 | 7.55E-01(5.96E-08) | 7.03E-01(4.18E-03) | 5.89E-01(8.29E-03) | 7.08E-01(1.31E-02) | 6.11E-01(1.12E-02) | 6.58E-01(1.18E-02) | 7.11E-01(9.64E-03) |

### 6.4.5 Running Time Analysis

To investigate how efficient our proposal is compared to other algorithms, we record the running time (in seconds) of all compared algorithms on all benchmark problems with the changing sequence of Equation (6.1) when $\tau_t = 50$ under the same hardware configuration [2]. Also, we use Friedman and Nemenyi statistical tests [90] with a significance level 0.05 across all benchmark problems regarding the running time of five compared algorithms to indicate the significance of the running time. The running time obtained by all algorithms on one problem under one run is regarded as an observation of the test.

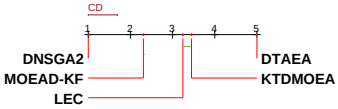



Figure 6.22: Friedman ranking among running time obtained by all compared algorithm in changing sequence of Equation (6.1). Here, larger ranks mean larger running time.

Fig. 6.22 presents the Nemenyi post-tests results among running time of all compared algorithms and Friedman detects significant differences in average running time with a p-value of 8.4050e-58. The specific mean values of running time (in seconds) obtained by all compared algorithms are presented in Table 6.30. It is clear from Fig. 6.22 that our proposed LEC achieves competitive efficiency compared to algorithms tailored for changing NObj (KTDMOEA and DTAEA). In addition, our intuition is right that algorithms with one archive (LEC and KTMOEA) cost significantly less running time than that with two archives (DTAEA). Meantime, algorithms without specific strategies to cope with changing NObj (DNSGA2 and MOEAD-KF) run faster.

## 6.5 Summary

It was studied in this Chapter that the state-of-the-art transfer approach (KTDMOEA) proposed in Chapter 5 still has some limitations when solving DMOPs with a changing NObj. Specifically, when increasing NObj, transferred solutions by KTDMOEA lacks diversity on the problem with extremely strong bias, since the heuristic PS expansion of KTDMOEA is unable to find expansion directions on this kind of problem. In addition, KTDMOEA loses convergence on problems with multi-modality and variable correlation when decreasing NObj, because solutions generated along the determined contraction directions only aim at enhancing diversity rather than convergence.

To overcome the limitations of KTDMOEA, we proposed a novel knowledge

[2]The implementation environment is as follows: 2.20-GHz Intel Core i7, 8-GB DDR4 2666MHz.

Table 6.30: Mean value of running time (in seconds) obtained by all compared algorithms when $\tau_t = 50$ in the changing sequence of Equation (6.1) under 31 runs.

| Problems | DNSGA2 | MOEAD-KF | DTAEA | KTDMOEA | LEC |
|---|---|---|---|---|---|
| F1 | 3.7093 | 20.4290 | 23.9588 | 14.3831 | 13.8774 |
| F2 | 5.8916 | 20.8500 | 37.9815 | 28.7111 | 28.1195 |
| F3 | 3.7658 | 20.7200 | 24.3278 | 14.7229 | 14.6698 |
| F4 | 6.1180 | 18.5460 | 38.3799 | 22.4218 | 27.7505 |
| WFG1 | 4.6587 | 15.0970 | 28.3667 | 19.2803 | 18.0803 |
| WFG2 | 5.1305 | 14.8890 | 32.6961 | 22.9704 | 22.6309 |
| WFG3 | 5.5108 | 15.0310 | 31.9358 | 22.7959 | 20.8459 |
| WFG4 | 6.2751 | 15.4280 | 40.9422 | 29.4419 | 30.2815 |
| WFG5 | 6.4904 | 15.2980 | 42.5587 | 32.9304 | 32.8443 |
| WFG6 | 5.9407 | 18.2600 | 36.1087 | 26.6608 | 24.2563 |
| WFG7 | 5.4063 | 14.5090 | 31.5811 | 21.6562 | 22.3603 |
| WFG8 | 5.6746 | 18.5440 | 31.6174 | 22.9454 | 25.1423 |
| WFG9 | 6.5267 | 15.9690 | 37.9773 | 22.2180 | 25.6519 |

transfer strategy, called learning to expand and contract PS (LEC), so as to enhance diversity and convergence for increasing and decreasing the NObj, respectively. LEC firstly conducts principal component analysis on PS of the problem before the change to get eigenvectors. Specifically, when increasing NObj, directions perpendicular to the obtained eigenvectors are found via solving homogeneous linear equations to be regarded as the candidate expansion directions. When decreasing NObj, these eigenvectors are directly seen as the candidate expansion directions. The reason is that solutions in the PS before the change could move towards the PS after the change along parts of these eigenvectors which have interaction to the PS after the change. Then, correct expansion and contraction directions are selected from their candidates by determining which expansion and contraction direction leads to diversity and convergence increase. Based on one solution (called base solution), directions along which generated solutions are nondominated to the base solution are regarded as leading to diversity increase. Directions along which generated solutions dominate the base solution are regarded as leading to increase in convergence. Lastly, based on these directions and the PS of the problem before the change, the PS is expanded and contracted for the new problem after changes.

Experimental investigations have demonstrated the effectiveness of the proposed LEC strategy in improving the diversity and convergence directly after changes. Moreover, for each of all frequencies of changes, the effectiveness of our proposed LEC strategy in enhancing solution quality has also been demonstrated after different generations (smaller than or equal to the frequency of changes) of outputting solutions in the optimization process. In addition, LEC performed well on different problems under different parameter settings.

# Chapter 7

# Conclusions and Future Work

This Chapter summarizes this thesis and its contributions and highlights possible future work.

## 7.1 Conclusions

In this thesis, we have proposed four research questions and proposed the corresponding answers to each of them, which will be summarized and presented in detail as follows.

In Chapter 3, a state-of-the-art knowledge transfer algorithm Tr-DMOEA [25] is taken as an example to study the first research question of "When and how to transfer knowledge when solving DMOPs with changing PSs and/or PFs position and/or shape?". Specifically, Tr-DMOEA is analysed on a series of DMO benchmark problems with different types of PS and/or PF changes and different frequencies of changes to generate transferred solutions and copied solutions without transfer. It has been found that transferred solutions have worse quality than solutions without transfer on problems with fixed PS and when environmental changes are small, since under these circumstance the previous optimal solutions have almost as good quality as the optimal solutions of the new environment. This observation gives the answer of the question "when to transfer": avoiding transfer learning on problems with fxed PS and when environmental changes are small. Moreover, a mathematical proof has been given to show that the Gaussian kernels used to map solutions in the objective space to the latent space as shown in equation (2.10) of Section 2.2.3.1 are not ideal in the existing Tr-DMOEA. The reason is that the used Gaussian kernel results in half of transferred solutions to be far away from the optimal solutions. Therefore, a linear kernel which does not have this problem is proposed to replace the Gaussian

one, answering the "how to transfer" question. Experimental studies have been conducted to verify these two answers, respectively. Then, an improved version of Tr-DMOEA has been proposed after combining these two answers via avoiding transfer when it is expected to fail and replacing the Gaussian kernel with linear one to improve the transfer. Lastly, comprehensive experimental studies have been conducted to verify the effectiveness of the improved Tr-DMOEA on different types of benchmark problems with different frequencies of changes through comparing quality of both solutions after changes and optimized solutions before the changes obtained by different DMOEAs. Experimental results have shown that the improved Tr-DMOEA obtained better solution quality than the original Tr-DMOEA on most of test problem instances under different optimization frameworks with different settings of environmental changes.

We found computational cost of Tr-DMOEA is extremely high when running its codes and program, although Tr-DMOEA has been reported to achieve good solution quality in [25]. Therefore, in Chapter 4, a time complexity analysis of each component of Tr-DMOEA and a computational study of the computational time of each component of Tr-DMOEA have been conducted to reveal what the efficiency of Tr-DMOEA is and which component of Tr-DMOEA is most time-consuming. The analyses have shown that the most time-consuming component is the 'inner' optimizer (interior point method) used for minimizing the distance of optimal solutions before the change and solutions to be found after the change in the latent space. Moreover, we have found that the inner optimizer is so expensive that it defeats the purpose of transfer optimization. Specifically, using computational cost of transfer learning to optimize randomly generated solutions leads to final solutions with significantly better convergence and diversity than those generated from transferred solutions under the same total computational budget. To verify whether the efficiency of Tr-DMOEA could be improved and whether the improved efficiency affects solution quality, we proposed to use two optimization methods SQP and active set [97] to replace the interior point method. Experimental results have been shown that Tr-DMOEAs with SQP and active set are more efficient than Tr-DMOEA with the interior point method. Moreover, it has also been experimentally demonstrated that Tr-DMOEA with the SQP method obtains best solution quality among all compared Tr-DMOEAs, which shows that the SQP method achieved a better balance between the transfer learning efficiency and effectiveness of solution quality.

Even though knowledge transfer has been successfully applied to solve DMOPs with changing shape and/or position of PSs and/or PFs, it has rarely been leveraged to tackle other types of DMOPs. In order to fill this gap, in Chapter 5, knowledge transfer has been carried out to study how to solve DMOPs with a changing NObj, another type of DMOP that also widely exist in project scheduling [9, 10], resource allocation [11, 12], etc. In order to reveal the limitations of the state-of-the-art algorithm in solving DMOPs with a changing NObj, an experimental investigation has been firstly carried out to highlight the limitations of the existing algorithm (DTAEA) [3] on DMOPs with a changing NObj where problem features are challenging, such as nonseparability, deceptiveness in the fitness landscape. It has been

found that DTAEA lacks diversity on these complex problems with challenging problem features. Facing this limitation, a novel knowledge transfer method called KTDMOEA has been proposed to enhance population diversity right after changes via PS expansion and contraction after determining the expansion and contraction directions in a heuristic way. Comprehensive experimental studies have been conducted on a series of benchmark problems with simple and complex problem features under different frequencies of changes and different NObj changing sequences through comparing KTDMOEA with several state-of-the-art DMOEAs including DTAEA. Experimental results have shown that the proposed KTDMOEA significantly outperformed the state-of-the-art DMOEAs on most test problem instances under different frequencies of changes and different NObj changing sequences, especially when the environmental changes are more frequent (i.e. the value of frequency of change is smaller). The proposed knowledge transfer strategy is also and further verified via comparing PS expansion and contraction to CA reconstruction of DTAEA under two different optimization frameworks. Moreover, parameter sensitivity was conducted on KTDMOA to reveal that its superiority over other algorithms is remained even though under different parameter settings.

As described, the proposed PS expansion/contraction in KTDMOEA was able to enhance diversity on most problem instances, yet it has been also shown to get worse solution quality right after changes on some problems. To verify this more specifically, an experimental study has been conducted in Chapter 6 on KTDMOEA. It has been shows that KTDMOEA lacks diversity on the problem with extremely strong bias right after increasing the number of objectives, and loses convergence on problems with multi-modality and variable correlation right after decreasing the number of objectives. The reason is that solutions generated from the heuristically determined expansion and contraction directions are unable to provide enough diversity or convergence on these problems, possibly because of selecting wrong directions. Facing this limitation of KTDMOEA, a novel knowledge transfer approach, called learning to expand and contract PSs (LEC) is then proposed. It firstly learns potentially good directions for expansion and contraction separately via principal component analysis, which overcomes the limitations of the heuristic in KTDMOEA when finding the expansion and contraction directions. Then, the most promising expansion and contraction directions are selected from their candidates according to whether they help diversity and convergence, respectively. Lastly, PS is expanded and contracted based on these most promising directions. Comprehensive experimental studies have been also conducted on a series of benchmark problems with simple and complex problem features under different frequencies of changes and different NObj changing sequences through comparing LEC with KTDMOEA and DTAEA for changing NObj and other two DMOEAs (DNSGA2 [7] and MOEAD-KF [43]) for changing position/shape of PSs/PFs. Experimental results have shown that the proposed LEC significantly outperforms the state-of-the-art DMOEAs on most of test problem instances under different frequencies of changes and different NObj changing sequences. It also shows that LEC indeed overcomes the limitations of KTDMOEA on problems where KTDMOEA performs worse. Moreover, parameter sensitivity was conducted on LEC to confirm its superiority over other algorithms

under different parameter settings.

## 7.2 Future Work

The future work will be conducted through the following aspects:

For the improved version of Tr-DMOEAs proposed in Chapter 3, the used kernel function to answer the question of how to transfer is a linear one, which could not deal with the data which are nonlinear in the latent space. In addition, regarding the answer of when to transfer, we just select one population after combining the copied solutions from the previous environment and transferred solutions together without explicitly figuring out whether the changes are small and the problem has fixed PSs when the transfer is expected to fail. Therefore, the first step of our future work if to find another non-linear kernel function that does not have the limitation of the Gaussian kernel. Another future work is to explore some potential metrics to measure the frequency of the environmental changes and whether one problem has fixed PSs.

For our computational studies in Chapter 4, we have found that using SQP method to optimize the 'inner' optimization problem introduced by transfer achieves best solution quality and transfer efficiency among three compared 'inner' optimizer. However, it is still unclear whether SQP makes it worth to use transfer instead of optimising from scratch. Therefore, one of our future work is to study using the computational cost of transfer with the SQP 'inner' optimizer to optimize randomly generated solutions from scratch, and then compare the solution quality of transfer and optimization from scratch.

Currently, for solving DMOPs with changing PSs and/or PFs position and/or shape, we only take Tr-DMOEA [25] as a transfer algorithm to study the research questions of "when and how to transfer knowledge" and "what is the efficiency of knowledge transfer" when solving DMOPs with changing PSs and/or PFs position and/or shape. In the future, other knowledge transfer algorithms such as [27, 81] could also be investigated to answer these research questions. Moreover, the used test benchmark functions are somehow simple. Therefore, other DMOP benchmark functions with more complex problem features and more complex change patterns like DF[107], SDP [51] could be also used to test the performance of transfer algorithms.

As expected, no algorithm would be best on all possible problems under all scenarios. For example, it has been found that our proposed KTDMOEA gets worse optimized solutions than the state-of-the-art algorithm DTAEA on problems with narrow valley. In addition, it has also been shown that KTDMOEA and LEC were

able to get better transferred solutions than DTAEA when the NObj changes are big, while they performed worse than DTAEA regarding quality of optimized solutions. Therefore, it is one of our future work to innovate more advanced knowledge transfer approaches to handle these scenarios well.

Considering that the questions of when and how to transfer have not been answered when solving DMOPs with a changing NObj, further enhancing our proposed KTDMOEA and LEC with specific methods to determine when and how to transfer in DMO with a changing NObj is proposed as future work.

Other test benchmark functions with more complex problem features than DTLZ and WFG suites for DMOPs with a changing NObj like mDTLZ [108] will be explored and used to verify the performance of our algorithms KTDMOEA proposed in Chapter 5 and LEC proposed Chapter 6, and further encourage the innovation of more advanced transfer approaches.

Currently, we are assuming PSs position of the DMOPs with a changing NObj is fixed. Moreover, the existing benchmark problems have rather simple PF shapes. Therefore, studying how to deal with DMOPs with both changing NObj and position and/or shape of PSs and/or PFs where the shape of PSs and/or PFs might be more complex is another of our future work. Facing this type of DMOPs, more advanced transfer algorithms that borrow ideas from transfer learning and machine learning algorithms can be developed.

Finally, testing our proposed algorithms on real-world problems, such as software project management [99] [9], cloud computing resources scheduling [109] [110], etc, is also one of our future work.

# Bibliography


[1] Gan Ruan, Leandro L Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. When and how to transfer knowledge in dynamic multi-objective optimization. In *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2034–2041. IEEE, 2019. 1, 1

[2] Gan Ruan, Leandro L Minku, Stefan Menzel, Bernhard Sendhoff, and Xin Yao. Computational study on effectiveness of knowledge transfer in dynamic multi-objective optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020. 2, 1

[3] Renzhi Chen, Ke Li, and Xin Yao. Dynamic multiobjectives optimization with a changing number of objectives. *IEEE Transactions on Evolutionary Computation*, 22(1):157–171, 2017. (document), 1.1, 2.1.3, 2.1.3.2, 1, 2.1.3.2.1, 2.1.3.2.2, 2.1, 2.2, 2.1.3.2.3, 2.1.3.2.3, 5, 5, 5.1, 14, 9, 8, 5.2.2, 10, 10, 5.3.1, 1, 5.3.2, 5.3.3, 5.3.4, 5.4.2.4, 6, 6.1, 8, 6.2.2.1, 6.2.2.2, 6.3.1, 6.3.2, 6.3.3, 6.3.4, 6.4.1.2, 7.1

[4] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. 1.1, 2.1.3.2.3, 3.1.2, 1, 3, 14, 8, 5.3.2

[5] Zhuhong Zhang. Multiobjective optimization immune algorithm in dynamic environments and its application to greenhouse control. *Applied Soft Computing*, 8(2):959–971, 2008. 1.1

[6] Shanmugam Palaniappan, S Zein-Sabatto, and A Sekmen. *Dynamic multi-objective optimization of war resource allocation using adaptive genetic algorithms*. IEEE, 2001. 1.1

[7] Kalyanmoy Deb, S Karthik, et al. Dynamic multi-objective optimization and decision-making using modified nsga-ii: a case study on hydro-thermal power scheduling. In *Evolutionary Multi-Criterion Optimization*, pages 803–817. Springer, 2007. 1.1, 2.1.2, 2.1.2.1, 5.3.2, 6.3.2, 7.1

[8] Kishalay Mitra, Saptarshi Majumdar, and Sasanka Raha. Multiobjective dynamic optimization of a semi-batch epoxy polymerization process. *Computers & Chemical Engineering*, 28(12):2583–2594, 2004. 1.1

[9] Xiaoning Shen, Leandro L Minku, Rami Bahsoon, and Xin Yao. Dynamic software project scheduling through a proactive-rescheduling method. *IEEE Transactions on Software Engineering*, 42(7):658–686, 2015. 1.1, 4.4, 7.1, 7.2

[10] Xiao-Ning Shen and Xin Yao. Mathematical modeling and multi-objective evolutionary algorithms applied to dynamic flexible job shop scheduling problems. *Information Sciences*, 298:198–224, 2015. 7.1

[11] Junchao Xiao, Leon J Osterweil, Qing Wang, and Mingshu Li. Dynamic resource scheduling in disruption-prone software development environments. In *International Conference on Fundamental Approaches to Software Engineering*, pages 107–122. Springer, 2010. 1.1, 4.4, 7.1

[12] Jianli Zhuo and Chaitali Chakrabarti. An efficient dynamic task scheduling algorithm for battery powered dvs systems. In *Proceedings of the 2005 Asia and South Pacific Design Automation Conference*, pages 846–849, 2005. 1.1, 7.1

[13] Mardé Helbig and Andries P Engelbrecht. Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm and Evolutionary Computation*, 14:31–47, 2014. 1.1, 2.1.2.4

[14] Shouyong Jiang, Juan Zou, Shengxiang Yang, and Xin Yao. Evolutionary dynamic multi-objective optimisation: A survey. *ACM Computing Surveys*, 55(4):1–47, 2022. 1.1, 2.1.2.4

[15] Gan Ruan, Guo Yu, Jinhua Zheng, Juan Zou, and Shengxiang Yang. The effect of diversity maintenance on prediction in dynamic multi-objective optimization. *Applied Soft Computing*, 58:631–647, 2017. 1.1, 2.1.2, 2.1.2.3, 5.3.4, 6.3.4

[16] Aimin Zhou, Yaochu Jin, and Qingfu Zhang. A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Transactions on Cybernetics*, 44(1):40–53, 2013. 2.1.2.3, 3

[17] Arrchana Muruganantham, Yang Zhao, Sen Bong Gee, Xin Qiu, and Kay Chen Tan. Dynamic multiobjective optimization using evolutionary algorithm with kalman filter. *Procedia Computer Science*, 24:66–75, 2013. 1.1, 2.1.2, 2.1.2.3

[18] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009. 1.1, 2.1.2.4, 2.2.3.1, 2.2.3.1.1, 3

[19] Sushil J Louis and Zhijie Xu. Genetic algorithms for open shop scheduling and re-scheduling. In *Proc. of the 11th ISCA Int. Conf. on Computers and their Applications*, volume 28, pages 99–102. Citeseer, 1996. 1.1, 2.2.2, 2.2.2.1.1

[20] Sushil Louis, Gary McGraw, and Richard O Wyckoff. Case-based reasoning assisted explanation of genetic algorithm results. *Journal of Experimental & Theoretical Artificial Intelligence*, 5(1):21–37, 1993.

[21] Sushil J Louis and John McDonnell. Learning with case-injected genetic algorithms. *IEEE Transactions on Evolutionary Computation*, 8(4):316–328, 2004. 1.1, 2.2.2, 2.2.2.1.1

[22] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Multifactorial evolution: toward evolutionary multitasking. *IEEE Transactions on Evolutionary Computation*, 20(3):343–357, 2015. 1.1, 2.2.2.2.1

[23] Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Insights on transfer optimization: Because experience is the best teacher. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2(1):51–64, 2017. 1.1, 2.2.1, 2.2.2

[24] Bingshui Da, Abhishek Gupta, and Yew-Soon Ong. Curbing negative influences online for seamless transfer evolutionary optimization. *IEEE Transactions on Cybernetics*, (99):1–14, 2018. 1.1, 2.2.2.1.2

[25] Min Jiang, Zhongqiang Huang, Liming Qiu, Wenzhen Huang, and Gary G Yen. Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 22(4):501–514, 2017. 1.1, 2.1.2, 2.1.2.4, 2.2.3, 2.2.3.1, 2.2.3.1.1, 2.2.3.1.2, 3, 3.1.1, 3.2.1, 4, 3, 3.4, 4, 4.1, 4.3.1, 1, 4.4, 5, 6, 7.1, 7.2

[26] Min Jiang, Zhenzhong Wang, Liming Qiu, Shihui Guo, Xing Gao, and Kay Chen Tan. A fast dynamic evolutionary multiobjective algorithm via manifold transfer learning. *IEEE Transactions on Cybernetics*, 51(7):3417–3428, 2020. 1.1, 2.2.3, 2.2.3.2, 5

[27] Min Jiang, Zhenzhong Wang, Shihui Guo, Xing Gao, and Kay Chen Tan. Individual-based transfer learning for dynamic multiobjective optimization. *IEEE Transactions on Cybernetics*, 51(10):4968–4981, 2020. 1.1, 2.1.2, 2.1.2.4, 2.2.3, 2.2.3.2, 5, 6, 7.2

[28] Xiao-Fang Liu, Zhi-Hui Zhan, Tian-Long Gu, Sam Kwong, Zhenyu Lu, Henry Been-Lirn Duh, and Jun Zhang. Neural network-based information transfer for dynamic optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(5):1557–1570, 2019. 1.1, 2.2.3

[29] Sheng-Uei Guan, Qian Chen, and Wenting Mo. Evolving dynamic multi-objective optimization problems with objective replacement. *Artificial Intelligence Review*, 23(3):267–293, 2005. 1.1, 2.1.3, 2.1.3.1

[30] Qingfu Zhang, Aimin Zhou, and Yaochu Jin. Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63, 2008. 1.1, 2.1.3.2.1, 3.1.1, 1

[31] Marco Farina, Kalyanmoy Deb, and Paolo Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 8(5):425–442, 2004. 2.1.1, 3, 4, 2.1.2, 4.2.1

[32] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. A survey of evolutionary continuous dynamic optimization over two decades—part a. *IEEE Transactions on Evolutionary Computation*, 25(4): 609–629, 2021. 2.1.2

[33] Danial Yazdani, Ran Cheng, Donya Yazdani, Jürgen Branke, Yaochu Jin, and Xin Yao. A survey of evolutionary continuous dynamic optimization over two decades—part b. *IEEE Transactions on Evolutionary Computation*, 25(4): 630–650, 2021. 2.1.2

[34] Chun-an Liu and Yuping Wang. New evolutionary algorithm for dynamic multiobjective optimization problems. In *Advances in Natural Computation*, pages 889–892. Springer, 2006. 2.1.2, 2.1.2.1

[35] Bojin Zheng. A new dynamic multi-objective optimization evolutionary algorithm. In *Third International Conference on Natural Computation (ICNC 2007)*, volume 5, pages 565–570. IEEE, 2007. 2.1.2.1

[36] Ruochen Liu, Wei Zhang, Licheng Jiao, Fang Liu, and Jingjing Ma. A sphere-dominance based preference immune-inspired algorithm for dynamic multi-objective optimization. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, pages 423–430, 2010. 2.1.2.1

[37] Carlos RB Azevedo and Aluizio FR Araújo. Generalized immigration schemes for dynamic evolutionary multiobjective optimization. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2033–2040. IEEE, 2011. 2.1.2, 2.1.2.1

[38] Chi-Keong Goh and Kay Chen Tan. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 13(1):103–127, 2009. 2.1.2, 2.1.2.1, 5.3.4, 6.3.4

[39] Yu Wang and Bin Li. Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment. In *Eleventh Conference on Congress on Evolutionary Computation*, pages 630–637, 2009. 2.1.2, 2.1.2.2

[40] Iason Hatzakis and David Wallace. Dynamic multi-objective optimization with evolutionary algorithms: a forward-looking approach. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 1201–1208. ACM, 2006. 2.1.2, 2.1.2.3

[41] Aimin Zhou, Yaochu Jin, Qingfu Zhang, Bernhard Sendhoff, and Edward Tsang. Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. In *Evolutionary Multi-Criterion Optimization*, pages 832–846. Springer, 2007. 2.1.2.3

[42] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME J. Basic Eng.*, 82(1):35–45, 1960. 2.1.2.3

[43] Arrchana Muruganantham, Kay Chen Tan, and Prahlad Vadakkepat. Evolutionary dynamic multiobjective optimization via kalman filter prediction. *IEEE Transactions on Cybernetics*, 46(12):2862–2873, 2015. 2.1.2.3, 5.3.2, 6.3.2, 7.1

[44] Zhou Peng, Jinhua Zheng, Juan Zou, and Min Liu. Novel prediction and memory strategies for dynamic multiobjective optimization. *Soft Computing*, 19:2633–2653, 2015. 2.1.2.3

[45] Yan Wu, Yaochu Jin, and Xiaoxiong Liu. A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Computing*, 19:3221–3235, 2015. 2.1.2, 2.1.2.3

[46] Arnold Neumaier and Tapio Schneider. Estimation of parameters and eigenmodes of multivariate autoregressive models. *ACM Transactions on Mathematical Software (TOMS)*, 27(1):27–57, 2001. 2.1.2.3

[47] Kay Chen Tan, Liang Feng, and Min Jiang. Evolutionary transfer optimization-a new frontier in evolutionary computation research. *IEEE Computational Intelligence Magazine*, 16(1):22–33, 2021. 2.1.2.4, 2.2, 2.2.2

[48] Shengxiang Yang and Xin Yao. *Evolutionary Computation for Dynamic Optimization Problems*, volume 490. Springer, 2013. 2.1.2.4

[49] Radhia Azzouz, Slim Bechikh, and Lamjed Ben Said. A multiple reference point-based evolutionary algorithm for dynamic multi-objective optimization with undetectable changes. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 3168–3175. IEEE, 2014. 2.1.3

[50] Liang Huang, Il Hong Suh, and Ajith Abraham. Dynamic multi-objective optimization based on membrane computing for control of time-varying unstable plants. *Information Sciences*, 181(11):2370–2391, 2011.

[51] Shouyong Jiang, Marcus Kaiser, Shengxiang Yang, Stefanos Kollias, and Natalio Krasnogor. A scalable test suite for continuous dynamic multiobjective optimization. *IEEE Transactions on Cybernetics*, 50(6):2814–2826, 2019. 2.1.3, 7.2

[52] Matthias Ehrgott. *Multicriteria optimization*, volume 491. Springer Science & Business Media, 2005. 2

[53] Yaochu Jin and Bernhard Sendhoff. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 3, pages 1910–1917. IEEE, 2003. 2.1.3.2.1

[54] Oliver Schütze, Carlos A Coello Coello, Sanaz Mostaghim, El-Ghazali Talbi, and Michael Dellnitz. Hybridizing evolutionary strategies with continuation methods for solving multi-objective problems. *Engineering Optimization*, 40 (5):383–402, 2008. 2.1.3.2.1

[55] Hu Zhang, Aimin Zhou, Shenmin Song, Qingfu Zhang, Xiao-Zhi Gao, and Jun Zhang. A self-organizing multiobjective evolutionary algorithm. *IEEE Transactions on Evolutionary Computation*, 20(5):792–806, 2016.

[56] Ke Li and Sam Kwong. A general framework for evolutionary multiobjective optimization via manifold learning. *Neurocomputing*, 146:65–74, 2014. 2.1.3.2.1

[57] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. 2.1.3.2.1, 2.1.3.2.2, 5.3.2, 5.3.3, 6.3.3

[58] Antonio J Nebro, Juan J Durillo, Francisco Luna, Bernabé Dorronsoro, and Enrique Alba. Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, 24(7):726–746, 2009.

[59] Xiaoliang Ma, Fang Liu, Yutao Qi, Xiaodong Wang, Lingling Li, Licheng Jiao, Minglei Yin, and Maoguo Gong. A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation*, 20(2):275–298, 2015. 2.1.3.2.1

[60] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*, pages 105–145. Springer, 2005. 2.1.3.2.2, 5.1, 5.3.1, 5.3.3, 6, 6.3.1, 6.3.3

[61] Michael D McKay, Richard J Beckman, and William J Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61, 2000. 2.1.3.2.3

[62] Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Transactions on Evolutionary Computation*, 19(5):694–716, 2014. 2.1.3.2.3, 5.3.3, 6.3.3

[63] Qingzheng Xu, Na Wang, Lei Wang, Wei Li, and Qian Sun. Multi-task optimization and multi-task evolutionary computation in the past five years: A brief review. *Mathematics*, 9(8):864, 2021. 2.2.1

[64] Tingyang Wei, Shibin Wang, Jinghui Zhong, Dong Liu, and Jun Zhang. A review on evolutionary multitask optimization: Trends and challenges. *IEEE Transactions on Evolutionary Computation*, 26(5):941–960, 2021. 2.2.1

[65] Bingshui Da, Abhishek Gupta, Yew-Soon Ong, and Liang Feng. Evolutionary multitasking across single and multi-objective formulations for improved problem solving. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1695–1701. IEEE, 2016. 2.2.1

[66] Dudy Lim, Yew-Soon Ong, Yaochu Jin, and Bernhard Sendhoff. Evolutionary optimization with dynamic fidelity computational models. In *Advanced*

*Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence: 4th International Conference on Intelligent Computing, ICIC 2008 Shanghai, China, September 15-18, 2008 Proceedings 4*, pages 235–242. Springer, 2008. 2.2.1

[67] Sinno Jialin Pan, Ivor W Tsang, James T Kwok, and Qiang Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010. 2.2.1, 2.2.3.1, 2.2.3.1.1, 2.2.3.1.1

[68] Liang Feng, Yew-Soon Ong, Siwei Jiang, and Abhishek Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. *IEEE Transactions on Evolutionary Computation*, 21(5):760–772, 2017. 2.2.2.1.1, 2.2.2.1.2

[69] Liang Feng, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Yew-Soon Ong, Kay-Chen Tan, and Alex Kai Qin. Evolutionary multitasking via explicit autoencoding. *IEEE Transactions on Cybernetics*, 49(9):3457–3470, 2018. 2.2.2.1.2

[70] Jiabin Lin, Hai-Lin Liu, Kay Chen Tan, and Fangqing Gu. An effective knowledge transfer approach for multiobjective multitasking optimization. *IEEE Transactions on Cybernetics*, 51(6):3238–3248, 2020. 2.2.2.1.2

[71] Zefeng Chen, Yuren Zhou, Xiaoyu He, and Jun Zhang. Learning task relationships in evolutionary multitasking for multiobjective continuous optimization. *IEEE Transactions on Cybernetics*, 52(6):5278–5289, 2020. 2.2.2.1.2

[72] Liang Feng, Yuxiao Huang, Lei Zhou, Jinghui Zhong, Abhishek Gupta, Ke Tang, and Kay Chen Tan. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. *IEEE Transactions on Cybernetics*, 51(6):3143–3156, 2020. 2.2.2.1.2

[73] Liang Feng, Wei Zhou, Weichen Liu, Yew-Soon Ong, and Kay Chen Tan. Solving dynamic multiobjective problem via autoencoding evolutionary search. *IEEE Transactions on Cybernetics*, 52(5):2649–2662, 2020. 2.2.2.1.3, 2.2.3, 2.2.3.2

[74] Abhishek Gupta, Jacek Mańdziuk, and Yew-Soon Ong. Evolutionary multitasking in bi-level optimization. *Complex & Intelligent Systems*, 1:83–95, 2015. 2.2.2.2.1

[75] Abhishek Gupta, Yew-Soon Ong, Liang Feng, and Kay Chen Tan. Multiobjective multifactorial optimization in evolutionary multitasking. *IEEE Transactions on Cybernetics*, 47(7):1652–1665, 2016. 2.2.2.2.1

[76] Kavitesh Kumar Bali, Yew-Soon Ong, Abhishek Gupta, and Puay Siew Tan. Multifactorial evolutionary algorithm with online transfer parameter estimation: Mfea-ii. *IEEE Transactions on Evolutionary Computation*, 24(1):69–83, 2019. 2.2.2.2.1

[77] Jinliang Ding, Cuie Yang, Yaochu Jin, and Tianyou Chai. Generalized multitasking for evolutionary optimization of expensive problems. *IEEE Transactions on Evolutionary Computation*, 23(1):44–58, 2017. 2.2.2.2.1

[78] Lei Zhou, Liang Feng, Kay Chen Tan, Jinghui Zhong, Zexuan Zhu, Kai Liu, and Chao Chen. Toward adaptive knowledge transfer in multifactorial evolutionary computation. *IEEE Transactions on Cybernetics*, 51(5):2563–2576, 2020. 2.2.2.2.1

[79] Liang Feng, Lei Zhou, Abhishek Gupta, Jinghui Zhong, Zexuan Zhu, Kay-Chen Tan, and Kai Qin. Solving generalized vehicle routing problem with occasional drivers via evolutionary multitasking. *IEEE Transactions on Cybernetics*, 51(6):3171–3184, 2019. 2.2.2.2.1

[80] Mei-Ying Cheng, Abhishek Gupta, Yew-Soon Ong, and Zhi-Wei Ni. Coevolutionary multitasking for concurrent global optimization: With case studies in complex engineering design. *Engineering Applications of Artificial Intelligence*, 64:13–24, 2017. 2.2.2.2.1

[81] Min Jiang, Zhenzhong Wang, Haokai Hong, and Gary G Yen. Knee point-based imbalanced transfer learning for dynamic multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):117–129, 2020. 2.2.3, 2.2.3.2, 5, 6, 7.2

[82] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in Neural Information Processing Systems*, 19, 2006. 2.2.3.1.1

[83] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004. 2.2.3.1.1, 3.2.1

[84] Sinno Jialin Pan, James T Kwok, Qiang Yang, et al. Transfer learning via dimensionality reduction. In *AAAI*, volume 8, pages 677–682, 2008. 2.2.3.1.1

[85] W Dai, Q Yang, G Xue, and Y Yu. Boosting for transfer learning. proceedings of the 24th international conference on machine learning. *Helsinki Finland: ACM*, pages 193–200, 2007. 2.2.3.2

[86] Min Jiang, Zhongqiang Huang, Liming Qiu, Wenzhen Huang, and Gary G Yen. Transfer learning-based dynamic multiobjective optimization algorithms. *IEEE Transactions on Evolutionary Computation*, 22(4):501–514, 2018. 3

[87] Mardé Helbig and A Engelbrecht. Benchmark functions for cec 2015 special session and competition on dynamic multi-objective optimization. *Dept. Comput. Sci., Univ. Pretoria, Pretoria, South Africa, Rep*, 2015. 3.1.1

[88] Margarita Reyes Sierra and Carlos A Coello Coello. Improving pso-based multi-objective optimization using crowding, mutation and?-dominance. In *International Conference on Evolutionary Multi-criterion Optimization*, pages 505–519. Springer, 2005. 3.1.1

[89] Qingya Li, Juan Zou, Shengxiang Yang, Jinhua Zheng, and Gan Ruan. A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. *Soft Computing*, 23(11):3723–3739, 2019. 3.1.1, 4.3.2.1

[90] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7(Jan):1–30, 2006. 3.1.3, 4.2.2.1, 4.3.1, 4.3.2.1, 4.3.2.2, 5.4, 6.4, 6.4.5

[91] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1-2):151–175, 2010. 3.2.1

[92] Alex Smola, Arthur Gretton, Le Song, and Bernhard Schölkopf. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory*, pages 13–31. Springer, 2007. 3.2.1

[93] Jaechang Nam, Sinno Jialin Pan, and Sunghun Kim. Transfer defect learning. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 382–391. IEEE, 2013. 3.2.2

[94] Bo Chen, Wai Lam, Ivor Tsang, and Tak-Lam Wong. Extracting discriminative concepts for domain adaptation in text mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 179–188. ACM, 2009. 3.2.2

[95] Sinno Jialin Pan, Xiaochuan Ni, Jian-Tao Sun, Qiang Yang, and Zheng Chen. Cross-domain sentiment classification via spectral feature alignment. In *Proceedings of the 19th International Conference on World Wide Web*, pages 751–760. ACM, 2010. 3.2.2

[96] CA Coello Coello and Maximino Salazar Lechuga. Mopso: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600)*, volume 2, pages 1051–1056. IEEE, 2002. 1, 3

[97] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999. 4.3.1, 7.1

[98] Yaochu Jin. Surrogate-assisted evolutionary computation: Recent advances and future challenges. *Swarm and Evolutionary Computation*, 1(2):61–70, 2011. 4.4

[99] Philippe Kruchten. *The rational unified process: an introduction*. Addison-Wesley Professional, 2004. 4.4, 7.2

[100] Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006. 5.1, 5.3.1, 6, 6.3.1

[101] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex Systems*, 9(2):115–148, 1995. 4, 3, 10

[102] Kalyanmoy Deb, Udaya Bhaskara Rao N, and Sindhya Karthik. Dynamic multi-objective optimization and decision-making using modified nsga-ii: A case study on hydro-thermal power scheduling. In *Evolutionary Multi-Criterion Optimization: 4th International Conference, EMO 2007, Matsushima, Japan, March 5-8, 2007. Proceedings 4*, pages 803–817. Springer, 2007. 5.3.2

[103] Kalyanmoy Deb. *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons, 2001. 5.3.3, 6.3.3

[104] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003. 5.3.4, 6.3.4

[105] Miqing Li and Xin Yao. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys (CSUR)*, 52(2):1–38, 2019. 5.3.4, 6.3.4

[106] Aimin Zhou. *Estimation of distribution algorithms for continuous multiobjective optimization*. PhD thesis, University of Essex, 2009. 8

[107] Shouyong Jiang, Shengxiang Yang, Xin Yao, Kay Chen Tan, Marcus Kaiser, and Natalio Krasnogor. Benchmark functions for the cec'2018 competition on dynamic multiobjective optimization. Technical report, Newcastle University, 2018. 7.2

[108] Zhenkun Wang, Yew-Soon Ong, and Hisao Ishibuchi. On scalable multiobjective test problems with hardly dominated boundaries. *IEEE Transactions on Evolutionary Computation*, 23(2):217–231, 2018. 7.2

[109] Zhi-Hui Zhan, Xiao-Fang Liu, Yue-Jiao Gong, Jun Zhang, Henry Shu-Hung Chung, and Yun Li. Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys (CSUR)*, 47(4):1–33, 2015. 7.2

[110] Zong-Gan Chen, Ke-Jing Du, Zhi-Hui Zhan, and Jun Zhang. Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, pages 708–714. IEEE, 2015. 7.2