

**Interactive Evolutionary Multi-Objective Optimization Algorithms:
Development, Improvements, Benchmarking and Analysis of
Performance**

A thesis submitted to the University of Manchester for the degree of
Doctor of Philosophy
in the Faculty of Humanities

2022

Seyed Mahdi Shavarani
Alliance Manchester Business School

Contents

List of figures	6
List of tables	10
Terms and abbreviations	12
Abstract	13
Declaration of originality	14
Copyright statement	16
Dedication	17
Acknowledgements	18
1 Introduction	19
1.1 Research Context	19
1.2 Research Aim	22
1.3 Contributions of the Thesis	23
1.4 Structure of the Thesis	24
1.5 Publications Resulting from the Thesis	26
1.5.1 Refereed Journal Papers	26
2 On Interactive Evolutionary Multi-Objective Optimization Algorithms	27
2.1 Multi-Objective Optimization	27
2.2 Interactive Evolutionary Multi-Objective Optimization Algorithms	30
2.2.1 Ad-hoc iEMOAs	33
2.2.2 Non-ad-hoc iEMOAs	33
2.3 Benchmarking Non-ad-hoc iEMOAs	36
2.3.1 Machine Decision Makers	37
2.3.2 Utility Function	38
2.3.3 Simulation of Non-idealities	40
2.3.4 Hidden and Irrelevant Objectives	41
2.4 Improving Robustness and Performance of iEMOAs	45

2.4.1	Detection and Handling Hidden and Irrelevant Objectives	45
2.4.2	An Improved Preference Learning Technique	46
3	On Benchmarking Interactive Evolutionary Multi-Objective Optimization	
	Algorithms	47
3.1	Introduction	47
3.2	State of the Art in iEMOAs	49
3.2.1	Background Concepts	49
3.2.2	Assessment of iEMOAs	50
3.3	Contributions of this chapter	52
3.3.1	Summary of selected algorithms	53
3.4	Methods	54
3.4.1	Machine Decision Maker (MDM)	54
3.4.2	Parameter tuning of the UF	59
3.5	Experimental design	61
3.5.1	Benchmark problems	61
3.5.2	Selected UFs and parameters	62
3.5.3	Parameter settings of the Machine Decision-Maker (MDM)	62
3.5.4	BCEMOA and iTDEA parameter settings	63
3.5.5	Implementations	65
3.6	Results	66
3.6.1	Experiments with no biases	66
3.6.2	Simulation of inconsistencies	70
3.6.3	Simulation of irrelevant objectives	70
3.6.4	Simulation of additively dependent objectives	71
3.7	Discussion	71
3.8	Conclusions	72
4	Detecting Hidden and Irrelevant Objectives in Interactive Multi-Objective Optimization	77
4.1	Introduction	78
4.2	Definitions	80
4.3	Background and Literature Review	83
4.4	Methods	85
4.4.1	Feature Selection	85
4.4.2	Fixed versus Variable Number of Active Objectives	88
4.4.3	BCEMOA-HD	89
4.5	Experimental Setup	90

4.5.1	Simulation of Active and Inactive Objectives	91
4.5.2	Underlying Benchmark Problems	92
4.5.3	Machine Decision Maker (MDM)	93
4.5.4	Selecting Relevant Objectives	94
4.5.5	Evaluation of the Results	94
4.6	Experimental Results	96
4.6.1	DTLZ Problems with Fixed Number of Objectives	96
4.6.2	DTLZ Problems with Variable Number of Objectives	97
4.6.3	ρ MNK Problems	98
4.6.4	99
4.7	Conclusion and Future Work	102
5	An Interactive Decision Tree-Based Evolutionary Multi-Objective Algorithm	104
5.1	Introduction	104
5.2	Literature Review	107
5.2.1	Non-ad-hoc Methods	108
5.2.2	Ad-hoc Methods	110
5.2.3	Decision Trees	111
5.2.4	Decision Tree Classification	112
5.3	Decision-Tree Based EMOA	113
5.3.1	Preference Elicitation	113
5.3.2	Preference Learning	113
5.3.3	Determination of the Score	114
5.3.4	Using DTs in EMOAs	115
5.4	Experimental Design	116
5.4.1	Machine Decision Maker (MDM)	116
5.4.2	Benchmark Problems	118
5.4.3	Evaluating Performance and the Competing Algorithms	118
5.4.4	Algorithm Parameter Settings	119
5.4.5	Implementations	120
5.5	Results & Discussion	120
5.5.1	Assessing Ranking Performance	120
5.5.2	Comparison of the Performance with Other iEMOAs	121
5.6	Conclusions	126
6	Conclusions & Future Study	129
6.1	Findings	129
6.2	Future Study	132

References	135
Appendices	161
A Supplementary Material to Chapter 3:	
On Benchmarking Interactive Evolutionary Multi-Objective Algorithms	162
B Supplementary Material to Chapter 4:	
Detecting Hidden and Irrelevant Objectives in Interactive Multi-Objective Optimization	179
C Supplementary Material to Chapter 5:	
Decision-Tree Based interactive Evolutionary Multi-Objective Optimization Algorithm	189

List of Figures

2.1	A typical MDM framework used for simulation of decision-making behaviors on experimenting with non-ad-hoc iEMOA.	38
2.2	Shape of Stewart UF with different parameters.	40
3.1	Solution representation (chromosome) used by the EA to find the Stewart UF parameters that optimize the single objective problem in Eq. (3.5). . . .	61
3.2	Results of the iEMOAs under ideal conditions with Tchebychef UF.	67
3.3	Results of the iEMOAs under ideal conditions with Stewart UF.	68
4.1	Comparison of the performance of different modes for DTLZ problems. . .	96
4.2	Comparison of the performance of different modes for DTLZ problems. . .	97
4.3	Comparison of the performance of different modes for ρ MNK problems with fixed number of objectives.	98
4.4	Comparison of the performance of different modes for ρ MNK problems with variable number of objectives.	99
4.5	Utility of the best-so-far solution within a single run after each interaction on DTLZ2 problem when using τ -HD ($\tau = 0.02$).	99
4.6	Convergence of the τ -HD ($\tau = 0.02$) towards relevant objectives ($c = \{1, 2\}$) through interactions on problem ρ MNK with 10 objectives.	100
4.7	Performance analysis of τ -HDR with different values of τ for DTLZ problems with $m = 20$. The number of interactions in all runs is 6.	101
4.8	Number of active objectives within a single run after each interaction for different values of τ in τ -HDR mode.	101
5.1	Construction of the training set from the ranked solutions over a problem with 2 objectives.	114
5.2	An instance of a decision tree trained on the information elicited in one interaction on problem DTLZ1 with $m = 2$ objectives.	116
5.3	Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA with simulation of inconsistencies in DM's decisions and Stewart UF.	124

5.4	The worst performances of DTEMOA in comparison with BCEMOA and iTDEA under ideal conditions with Stewart UF.	126
5.5	Some of the best and worst performances of the algorithms under ideal conditions with Stewart UF.	127
5.6	Location of the returned solutions with different algorithms on DTLZ7 problem.	128
A.1	Comparison of the performance of the algorithms with simulation of inconsistencies and Tchebychef utility function.	163
A.2	Comparison of the performance of the algorithms with simulation of inconsistencies and transformed Stewart's utility function.	164
A.3	Comparison of the performance of the algorithms with correlated objectives and Tchebychef utility function.	165
A.4	Comparison of the performance of the algorithms with correlated objectives and the transformed Stewart's utility function.	166
A.5	Comparison of the performance of the algorithms with simulation of a irrelevant objective and the Tchebychef utility function.	167
A.6	Comparison of the performance of the algorithms with simulation of a irrelevant objective and the transformed Stewart's utility function	168
A.7	Marginal utility functions for test 31. The values of the objectives in the MPS is depicted by an asterisk.	169
A.8	Marginal utility functions for test 32. The values of the objectives in the MPS is depicted by an asterisk.	169
A.9	Marginal utility functions for test 33. The values of the objectives in the MPS is depicted by an asterisk.	171
A.10	Marginal utility functions for test 34. The values of the objectives in the MPS is depicted by an asterisk.	171
A.11	Marginal utility functions for test 36. The values of the objectives in the MPS is depicted by an asterisk.	172
A.12	Marginal utility functions for test 37. The values of the objectives in the MPS is depicted by an asterisk.	172
A.13	Marginal utility functions for test 38. The values of the objectives in the MPS is depicted by an asterisk.	173
A.14	Marginal utility functions for test 39. The values of the objectives in the MPS is depicted by an asterisk.	173
A.15	Marginal utility functions for test 41. The values of the objectives in the MPS is depicted by an asterisk.	174

A.16	Marginal utility functions for test 42. The values of the objectives in the MPS is depicted by an asterisk.	174
A.17	Marginal utility functions for test 43. The values of the objectives in the MPS is depicted by an asterisk.	175
A.18	Marginal utility functions for test 44. The values of the objectives in the MPS is depicted by an asterisk.	175
A.19	Heatmap vs. Pf : DTLZ1, DM type 1	175
A.20	Heatmap vs. Pf : DTLZ1, DM type 2	176
A.21	Heatmap vs. Pf : DTLZ1, DM type 3	176
A.22	Heatmap vs. Pf : DTLZ1, DM type 4	176
A.23	Heatmap vs. Pf : DTLZ2, DM type 1	176
A.24	Heatmap vs. Pf : DTLZ2, DM type 2	177
A.25	Heatmap vs. Pf : DTLZ2, DM type 3	177
A.26	Heatmap vs. Pf : DTLZ2, DM type 4	177
A.27	Heatmap vs. Pf : DTLZ7, DM type 1	177
A.28	Heatmap vs. Pf : DTLZ7, DM type 2	178
A.29	Heatmap vs. Pf : DTLZ7, DM type 3	178
A.30	Heatmap vs. Pf : DTLZ7, DM type 4	178
B.1	Comparison of the performance of different modes for DTLZ problems with $m = 4$. The number of active objectives are fixed.	180
B.2	Comparison of the performance of different modes for DTLZ problems with $m = 10$. The number of active objectives are fixed.	180
B.3	Comparison of the performance of different modes for DTLZ problems with $m = 20$. The number of active objectives are fixed.	181
B.4	Comparison of the performance of different modes for DTLZ problems with $m = 4$ with variable number of objectives.	181
B.5	Comparison of the performance of different modes for DTLZ problems with $m = 10$ with variable number of objectives.	181
B.6	Comparison of the performance of different modes for DTLZ problems with $m = 20$ with variable number of objectives.	182
B.7	Comparison of the performance of different modes for ρ MNK problems with $m = 4$. Number of active objectives is fixed.	182
B.8	Comparison of the performance of different modes for ρ MNK problems with $m = 10$. Number of active objectives is fixed.	183
B.9	Comparison of the performance of different modes for ρ MNK problems with $m = 20$. Number of active objectives is fixed.	183

B.10	Comparison of the performance of different modes for ρ MNK problems with $m = 4$ with variable number of objectives.	184
B.11	Comparison of the performance of different modes for ρ MNK problems with $m = 4$ with variable number of objectives.	184
B.12	Comparison of the performance of different modes for ρ MNK problems with $m = 20$ with variable number of objectives.	185
B.13	Performance analysis of τ -HDR with different values of τ for DTLZ problems with $m = 20$	185
B.14	Number of active objectives after each interaction for different values of τ in τ -HDR mode on DTLZ test problems.	186
B.15	Location of the returned solutions with different modes compared to golden mode.	186
C.1	Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA under ideal conditions with Stewart UF.	192
C.2	Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA under ideal conditions with Tchebychef UF.	195
C.3	Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA with simulation of inconsistencies in DM's decisions and Stewart UF.	196
C.4	Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA with simulation of inconsistencies in DM's decisions and Tchebychef UF.	197

List of Tables

2.1	Description of different types of DM behaviors simulated by combinations of τ_i and λ_i adapted from (Stewart, 1996).	39
3.1	Description of different DM behaviors simulated by combinations of τ_i and λ_i for minimization problems.	56
3.2	List of UFs used in the tests.	64
3.3	The results of experiments with the simulation of inconsistencies in the decisions of the DM for BCEMOA and iTDEA.	74
3.4	The results of experiments with the simulation of irrelevant objectives for many-objective problems.	75
3.5	Results of experiments with the simulation of additively dependent objectives for BCEMOA and iTDEA.	76
4.1	A numerical instance of uni-variate feature selection.	87
5.1	Description of different DM behaviors simulated by combinations of τ_i and λ_i for minimization problems.	117
5.2	Parameter settings of the iEMOAs used in experiments.	119
5.3	Mean accuracy (and standard deviation) of preference learning in DTEMOA and BCEMOA on the DTLZ7 problem for different UFs.	122
5.4	Comparing the robustness of algorithms over experiments under ideal conditions and with the simulation of inconsistencies in the DM's decisions.	123
5.5	Comparing the performance of the algorithms over over different number of interactions and under ideal conditions.	123
5.6	Comparing the performance of algorithms over various number of objective functions.	125
5.7	Comparing the performance of algorithms over different decision-making types.	125
B.1	Comparing the accuracy of F-regression test and mutual information in feature selection.	187
B.2	Weights of active objectives used in the Tchebychef function and shape of the UF1-UF3 used in experiments.	187

B.3	Comparison of <i>Only learning</i> Mode with detection of irrelevant objectives approach corresponding to Fig 4.3.	188
B.4	Comparison of <i>Only learning</i> Mode with detection of irrelevant objectives approach in experiments depicted in Fig 4.4	188
C.1	A sample training set acquired from an experiment on DTLZ1 with 2 objectives	190
C.2	Examples used for building the tree constructed using the data in Table C.1	190
C.3	Test specifications.	191
C.4	The results of experiments under ideal conditions for BCEMOA, DTEMOA and iTDEA.	193
C.5	The results of experiments with simulation of inconsistencies in DM's decisions for BCEMOA, DTEMOA and iTDEA.	194

Terms and abbreviations

DM: Decision Maker

DT: Decision Tree

EMOA: Evolutionary Multi-Objective Algorithm

iEMOA: Interactive Evolutionary Multi-Objective Algorithm

MCDM: Multi-Criteria Decision Making

MDM: Machine Decision Maker

MOA: Multi-Objective Optimization Algorithm

MOP: Multi-objective Optimization Problem

MPS: Most Preferred Solution

PF: Pareto Front

UF: Utility Function

Abstract

Interactive multi-objective optimization algorithms exploit the preferences of a human Decision Maker (DM), elicited iteratively during the optimization process to direct the search and increase computational efficiency. Despite their undoubted promise, progress in the field has been hindered by challenges in experimenting with these methods, the main one being the fact that the results are affected by the decisions of the DM and the difficulty controlling or accounting for natural variation among DMs. One may propose using many DMs to eliminate biases in the experiments arising from DM variability. However, this would make the experiments expensive and non-replicable. The other approach is to replace the human DM with a Machine DM (MDM) in the experiments. Only a few studies have considered the simulation of decision-making behaviors in the context of interactive methods, most of which have used arbitrary utility functions to indicate the desirability of the solutions. However, it is not enough to replace the DM with a utility function assuming an ideal DM whose decisions are consistent throughout the experiments. This thesis builds on one of the only proposed MDMs that incorporate biases and proposes improvements by integrating a utility function based on psychological studies and widely used in the literature to simulate decision-making behaviors. We also expand on the types of non-idealities that are modeled, namely inconsistent decisions and preferentially dependent objectives. Moreover, we formally define and investigate the notion of irrelevant and hidden objectives and propose an approach to simulate them in the experiments. Irrelevant objectives are defined as those that are considered by the optimizer but not by the DM. Irrelevant objectives complement hidden objectives, defined as those that the DM considers but is not the optimizer. Noting that irrelevant objectives add to the complexity of the problem without improving the DM's satisfaction with the results, we propose a feature selection method that can be integrated into any interactive method to automatically filter out irrelevant objectives in an online mode and optimize only important ones. The research is further extended by proposing a new interactive method where the DM's preferences are estimated using decision trees, which naturally perform feature selection. The proposed preference learning technique targets limitations with other methods in the literature. We use our proposed MDM to compare the performance of this novel iEMOA with two state-of-the-art interactive methods. The results suggest that the superior performance of the proposed method is not degraded with the increase of the objective functions and is robust towards noise and non-idealities.

Blank Page

Declaration of originality

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institutes of learning.

Copyright statement

- i The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and they have given The University of Manchester certain rights to use such Copyright, including for administrative purposes.
- ii Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made *only* in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.
- iii The ownership of certain Copyright, patents, designs, trademarks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.
- iv Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see <http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=24420>), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see <http://www.library.manchester.ac.uk/about/regulations/>) and in The University’s policy on Presentation of Theses.

Dedication

To my parents.

This thesis is a humble tribute to your endless love and support.

Acknowledgements

I would like to express my sincere gratitude and appreciation to all those who have supported and encouraged me throughout my research.

First and foremost, I would like to thank my thesis advisors, Professor Manuel López-Ibáñez and Professor Richard Allmendinger, for their invaluable guidance, expertise, and support throughout the entire thesis process. Their insightful comments, constructive criticism, and encouragement have been instrumental in shaping my work. I would also like to express my gratitude to Professor Joshua Knowles for providing further support and supervision throughout my studies. His expertise and guidance have been invaluable in helping me to navigate through the challenges of this research project.

I would also like to thank the members of my thesis committee, Professor Jian-Bo Yang and Professor Mariano Luque Gallego, for their time, expertise, and constructive feedback, which have helped me to refine and improve my research.

I am grateful to the University of Manchester for providing me with the necessary resources and facilities to carry out my research work.

I want to thank my family and friends for their unwavering support, encouragement, and love. Their moral support and understanding have been crucial in helping me stay focused and motivated throughout this journey.

Finally, I would like to express my deepest gratitude to Professor Theodor Stewart. His willingness to generously share his insights, knowledge, and experience has made a significant contribution to the quality and rigor of my work.

Thank you all for your support and encouragement.

Chapter 1

Introduction

1.1 Research Context

Many real-life optimization problems deal with multiple conflicting objective functions, necessitating the use of Multi-Objective Optimization Algorithms (MOAs). Unlike single-objective optimization, where a single optimal solution exists, in Multi-Objective Optimization Problems (MOPs), no solution corresponds to the optimal value of all objectives. Instead, the final goal of solving a MOP is to find a set of so-called Pareto optimal solutions with desirable trade-offs for which improving the value of one objective is only possible with a sacrifice in the value of at least one other objective (Deb, 2005b, 2008). Pareto optimal solutions are not mathematically comparable, and so-called *preferences* (of an expert or concerned decision maker (DM)) should be considered to select the final solution. Thus, there are two phases to any MOP that should be considered. An optimization phase to produce Pareto optimal solutions and a decision-making phase to select the most preferred solution (MPS) among the set of generated solutions (Branke, Deb, Miettinen, & Słowiński, 2008). A general classification of MOAs is based on the stage where the decision-making phase happens, and the preferences of the Decision Maker (DM) are elicited. In this regard, the MOAs are categorized as *a priori*, *a posteriori* and *interactive* methods. In *a priori* methods, the DM is consulted before starting the optimization phase. Not knowing the possibilities and the bounds of the objective functions, the DM can be too optimistic or pessimistic and bias the search. In *a posteriori* methods, a well-distributed set of Pareto optimal solutions are generated, from which the DM should select the MPS. Evolutionary Multi-Objective Optimization Algorithms (EMOAs) are widely used as *a posteriori* method as they naturally work with a population of solutions and can generate a set of the Pareto optimal solutions in one single run (Coello Coello, 2006). Many EMOAs rely on Pareto dominance to preserve elitism. However, it has been shown that when the number of objective functions increases, Pareto dominance loses its selection pressure, and the algorithm fails to be effective (Purshouse & Fleming, 2003a; Hughes, 2005; Aguirre & Tanaka, 2007). On the other hand, even in EMOAs that overcome this problem and generally in other *a posteriori* methods,

the decision-making phase is difficult as the number of required solutions to provide a well-distributed set of Pareto optimal solutions increases exponentially with the number of objective functions (Deb, Sinha, Korhonen, & Wallenius, 2010a). Interactive EMOAs (iEMOAs) elicit and exploit DM's preferences iteratively during the optimization process to limit the search and generate only Pareto optimal solutions with **desirable** trade-offs and thus increase the computational efficiency (Belton et al., 2008; Jaszekiewicz & Branke, 2008; Miettinen, Ruiz, & Wierzbicki, 2008). The preference information is elicited in a cognitively affordable way in each interaction and is exploited to direct the search in the next optimization phase. Thus, there are significant savings in both optimization and decision-making phases regarding computational requirements and DM's cognitive effort (Afsar, Miettinen, & Ruiz, 2021). Furthermore, the increased selection pressure makes it possible for iEMOAs to reasonably estimate the actual Pareto optimal solutions and improve the quality of the final solutions (Shavarani, López-Ibáñez, & Knowles, 2021). These perceived advantages have led to the introduction of many alternative iEMOAs in recent years. However, improvements in the field can be achieved only through extensive benchmarking and quantitative assessment of the performance of these methods (López-Ibáñez & Knowles, 2015). Although researchers have provided guidelines and metrics for evaluation of EMOAs (Knowles & Corne, 2002; Knowles, Thiele, & Zitzler, 2006), experimenting on interactive EMOAs remains problematic due to the influence of the DM's decisions on final results, making experimental results subjective and difficult to replicate (Afsar, Ruiz, & Miettinen, 2021).

Having an individual DM throughout the experiments would lead to human-specific biases such as fatigue (Zujevs & Eiduks, 2008). Alternatively, one may use a large number of DMs to cancel out the biases which arise due to variations in decision-making behaviors among different DMs (Wallenius, 1975; Buchanan, 1994). However, some authors argue that such experiments may require tens of DMs to obtain reliable conclusions (Zujevs & Eiduks, 2008), which makes the experiments expensive. Therefore, most of the experiments with real DMs reported in the literature are conducted with the participation of students acting as DMs (Afsar et al., 2021). As students are not experts or responsible for the outcomes, the results of such experiments may not be reliable. Other than that, having real DMs in the experiments would make the results subjective and difficult to replicate (Chen, Xin, Chen, & Li, 2017). One possible approach is to conduct experiments by replacing human DMs with a reliable simulation of decision-making behaviors, known as Artificial decision-makers (Afsar et al., 2021) or Machine Decision Makers (MDM) (López-Ibáñez & Knowles, 2015). Only a few studies have touched the surface of this subject and directed their efforts toward the simulation of decision-making behaviors. The majority of these studies employ a Utility Function (UF) to simulate decision makers' preferences in the experiments, most of which fail to validate

the accepted assumptions for simulating decision-making behaviors such as additive independence (Keeney, 1981), trade-off interpretability and compensatory preferences (Stewart, 1996), the convexity of losses and concavity of gains (Kahneman & Tversky, 1979). Thus, it is unknown if they can provide a reliable simulation of decision-making behaviors.

It is easy to bias the results if the UF is not selected properly. For instance, an iEMOA that works well when the DM is simulated by a linear UF may completely break down when a non-linear UF is used. Such issues highlight the need for investigating the behavior of iEMOAs where preferences are simulated using more complex and realistic UFs. Besides, humans are subject to cognitive bottlenecks and prone to mistakes, and as humans, DMs are susceptible to biases and errors. Thus, it is crucial to measure the robustness of iEMOAs towards biases and non-idealities that may happen during the interactions; either inside DM's mind, modeling phase, or even inside the optimization engine. This task is not possible, except through extensive measuring and benchmarking of such methods using an MDM that can provide a realistic simulation of decision-making behaviors and the non-idealities. To this end, the following research goals should be addressed:

The first goal is to build a reliable and realistic MDM to facilitate conducting reliable experiments. There are two main components to MDMs that should be carefully examined: UF and simulation of the non-idealities. There are UFs that have their roots in psychological studies and have been widely accepted in the literature for the simulation of decision-making behaviors. However, there is no research on experimenting with iEMOAs that utilize such realistic UFs. Furthermore, To measure the robustness of the interactive methods, it is crucial to identify and simulate non-idealities that exist in real-life conditions, which can downgrade the performance of iEMOAs (Stewart, 2005). Some studies have addressed this subject and scrutinized decision-making behaviors, and potential biases that may happen inside the DM's mind or during the interactions (Tversky & Kahneman, 1974; Kahneman & Tversky, 1979; Stewart, 1996; Buchanan & Corner, 1997). However, only a few studies have considered the effect of such biases on the performance of iEMOAs. One such study is the work by Stewart (1996), in which the effect of inconsistent decisions and preferentially dependent objectives on MOAs was investigated. For the first time in the context of iEMOAs, some of the non-idealities in Stewart (1996) were simulated by López-Ibáñez and Knowles (2015). However, their study was limited to linear UFs, therefore there is still room to extend the list of simulated non-idealities by including those discussed in the literature.

After developing a reliable MDM, the second goal is to improve iEMOAs and make them robust toward simulated non-idealities. This can be achieved through extensive measuring and benchmarking of iEMOAs, systematically proposing targeted improvements to how they interact with the DM, how the solutions are filtered for presentation, how the DM's preferences

are modeled, and other elements of the algorithm.

1.2 Research Aim

We contribute to the advancement of iEMOAs by addressing the issues outlined in the previous section. There is the problem of measurement and evaluation of the performance of iEMOAs and simulation of decision-making behaviors in the experiments. We introduce a UF with its root in psychological studies which enables reliable simulation of different decision-making behaviors. Second, we discuss non-idealities that might happen inside the DM's mind, in the system-human interactions, or even in the modeling phase and propose approaches to simulate them in the experiments on iEMOAs.

It is vital to make iEMOAs robust to any potential non-idealities that might downgrade the solution process. We aim to extend the list of non-idealities that are discussed in previous studies by introducing hidden and irrelevant objectives. Hidden objectives are defined as those objectives that are important to the DM but are not being optimized. Such a scenario had been previously considered in the literature but no formal definition has been provided. *Irrelevant objectives* are defined as those objectives that are being optimized but are not significantly important to the DM. Any irrelevant objective that may exist in the model, would add to the complexity of the problem without improving the quality of the solutions from DM's perspective. On the other hand, as Battiti and Campigotto (2010) put it, hidden objectives would make the results of the optimization unsatisfactory. Thus, it would be ideal to detect and eliminate irrelevant objectives and instead optimize any important objective that has remained hidden.

Another limitation in the area of iEMOAs is how the preferences of the DM are modeled in the algorithm. Many iEMOAs assume a linear function for the preferences of the DM (Zionts & Wallenius, 1976; Jaszkievicz, 2004; Quan, Greenwood, Liu, & Hu, 2007a). Other approaches that have waived the linearity assumption of the preference model are limited whether by the number of different shapes of the UF that can be estimated or by their underlying assumptions that are often difficult to validate in real-life problems (Branke, Corrente, Greco, Słowiński, & Zielniewicz, 2016). Besides, many preference learning techniques that do not assume the linearity of the preference model, make assumptions that are not easy to validate in real-life problems. There are also studies that use machine-learning techniques to learn a complete ordering of the non-dominated solutions (Battiti & Passerini, 2010). Such learn-to-rank machine learning algorithms, are mainly in the form of an ensemble of many classifiers and are prone to errors and susceptible to considerable deviations when confronted with small biases (Cheng, Hühn, & Hüllermeier, 2009). However, it is still possible to use

machine learning in estimating some quality of the solutions or for classification purposes without being exposed to such biases. Preference learning techniques are desired to be (i) able to adapt to the complex decision-making behaviors of the DM without limitations and restricting assumptions; and (ii) robust toward any outlier and non-ideality that may happen under real-life conditions.

1.3 Contributions of the Thesis

The main contributions of this thesis to the field of iEMOAs can be summarized as follows:

We build on one of the only MDM frameworks proposed in López-Ibáñez and Knowles (2015) that investigated the effect of biases on the performance of iEMOAs. We enrich this MDM using a Sigmoid UF, which is based on psychological research by Kahneman and Tversky (1979) who established that preferences of human DMs follow an *S*-shaped curve that is concave for gains and convex for losses. The Sigmoid UF also satisfies the quasi-concavity requirement for UFs in the economics literature (Mas-Colell, Whinston, Green, et al., 1995) and is widely used in the literature for simulation of decision-making behaviors, although not in the context of iEMOAs. We use a specific type of the Sigmoid UF introduced by Stewart (1996) in his experiments on the robustness of additive UFs, hereafter called Stewart UF. A valuable property of the Stewart UF is its ability to simulate different decision-making behaviors. We further propose a systematic method to get the parameters of the Stewart UF in order to simulate a specific decision-making behavior and avoid trivial UFs.

We extend the non-idealities simulated in the original study of López-Ibáñez and Knowles (2015) by considering inconsistent decisions, preferentially dependent objective functions, hidden objectives, and irrelevant objectives. As a novel contribution of this study, we provide formal definitions of hidden and irrelevant objectives, which complement the concept of redundant objectives that already exists in the literature. An approach is proposed to integrate the simulation of these non-idealities into any existing benchmark problem. A new method is proposed based on several feature selection methods which enable a given iEMOA to dynamically detect hidden and irrelevant objectives and adjust the set of objective functions such that only those objectives that are important to the DM are optimized while the irrelevant ones are eliminated. The performance of the method is validated using problems of varying dimensionality and complexity and different UFs to represent different DMs.

Finally, we propose a novel iEMOA with a robust preference learning technique based on decision trees (DTs). The DT classifier predicts the winner solution with desirable trade-offs in pairwise comparisons. Thus, there are no restrictions on the shape of the DM's preferences. DTs can naturally handle outliers and biases and eliminate irrelevant objectives in their learn-

ing and predictions. The learning algorithms are prone to errors, and DTs are no exception. To further reduce the effect of such errors on the results, we exploit the capabilities of DTs in providing the probability of each classification. To this end, we compare each solution with all other solutions in the population and use the accumulated probability of winnings as a score to get a comprehensive idea of the quality of that solution relative to the current population. The score is used in deciding on the solutions that should be pruned and those that survive in each generation of an EMOA. Such measures make the proposed method robust towards internal and external errors. The algorithm's performance is then compared with two other iEMOAs from the literature using the MDM developed in this study.

1.4 Structure of the Thesis

This thesis is in journal format. Each chapter of this thesis, except Chapter 2, is self-contained and in the format published or submitted for publication. Thus, some content from earlier chapters might be repeated in the subsequent ones. The rest of this dissertation is organized as follows:

Chapter 2 introduces the iEMOAs and provides a background on the subject. In this regard, related definitions and assumptions are provided, different aspects of the iEMOAs are discussed, a non-exhaustive list of important studies on the measurement of iEMOAs and simulated MDMs are given, and a review of underlying assumptions of Sigmoid UFs is presented.

Chapter 3 details challenges and problems associated with experimenting with iEMOAs and investigates previous efforts on the simulation of real DMs in the experiments. The shortcomings and limitations of earlier simulations of decision-making behaviors are scrutinized. Next, a comprehensive realistic MDM is proposed based on the research by Stewart (1996, 2005) and the MDM framework proposed by López-Ibáñez and Knowles (2015). The MDM is powered by a specific type of Sigmoid UFs, that was proposed by Stewart (1996), hence called Stewart UF in this study. This UF was initially designed for maximization problems. Thus, a transformation of the Stewart UF is proposed to be used with minimization problems without violating its underlying assumptions. A method is also proposed to set the parameters of the UF to simulate a specific type of decision-making behavior. The simulation of the DM is further improved with the introduction of biases and non-idealities that may arise under real conditions. This chapter is based on the manuscript submitted to IEEE Transactions on Evolutionary Computation, which is now undergoing a second round of revisions.

In Chapter 4, irrelevant and hidden objectives are discussed in depth, formal definitions are provided, and an approach is presented for simulating these non-idealities. Thereupon,

a method is proposed that detects the existence of irrelevant and hidden objectives, removes irrelevant objectives from the set of objectives being optimized, and replaces them with the hidden ones during the optimization process to optimize only those objectives that are important to the DM. The method helps to: (a) reduce the computational cost and improve the selection pressure and convergence, which have been the main problems in many-objective problems; and (b) Adapt the optimization model to the DM's preferences and make the final results more appealing. The possible usage of the method in dimension reduction is showcased, and the cons and pros are compared to other methods that do so by eliminating redundant objectives. For the first time, here we propose an approach that can exploit the preference information elicited from the DM in iEMOAs for dimension reduction, identification of irrelevant objectives, and preserving those objectives that are important to the DM. A set of experiments is performed to evaluate the performance of the method, and the results are discussed. This chapter is based on the manuscript accepted for publication in IEEE Transactions on Evolutionary Computation, which is now undergoing a second round of revisions.

Based on the results of Chapter 4 which indicate the effectiveness of filtering the set of objectives on the quality and efficiency of the iEMOAs, Chapter 5 of this study proposes a ranking-based iEMOA with a preference-learning scheme based on decision trees that deploy feature selection naturally. Instead of learning a UF or ranking compatible with DM's preferences, the DT is used to draw rules on the trade-offs of any pair of solutions to predict the probability that a solution is preferred over the other in pairwise comparisons. The proposed preference learning method is compared with other methods from the literature, and the results are analyzed. The performance of the developed iEMOA is also compared with other iEMOAs using the MDM framework proposed in Chapter 3. The contents of this chapter are submitted to the European Journal of Operations Research.

Finally, Chapter 6 presents conclusions. Particularly, the research objectives are revisited, the study's findings are summarized, the limitations are elaborated, possible extensions of the experiments are laid out, and research directions are provided. We also briefly discuss the lessons learned and important considerations that should be noticed when selecting the UF, the parameters of the UF, different levels of non-idealities, and the test problems.

1.5 Publications Resulting from the Thesis

1.5.1 Refereed Journal Papers

1. Shavarani, S. M., López-Ibáñez, M., & Knowles, J. On Benchmarking Interactive Evolutionary Multi-Objective Algorithms. *IEEE Transactions on Evolutionary Computation*, submitted 2021, under second round of peer-review.
2. Shavarani, S. M., López-Ibáñez, M., & Allmendinger, R. Detection of Hidden Objectives and Interactive Objective Reduction. *IEEE Transactions on Evolutionary Computation*.
3. Shavarani, S. M., López-Ibáñez, M., Allmendinger, R. , & Knowles, J. An Explainable interactive Evolutionary Multi-Objective Optimization Algorithm: DTEMOA. *European Journal of Operations Research*, submitted 2022.

Refereed Conference Papers

1. Shavarani, S. M., López-Ibáñez, M., & Knowles, J. (2021, June). Realistic UFs prove difficult for state-of-the-art interactive multi-objective optimization algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 457-465).
2. Shavarani, S. M., López-Ibáñez, M., Allmendinger, R. , & Knowles, J. An Interactive Decision Tree-Based Evolutionary Multi-Objective Algorithm. Accepted in *Evolutionary Multi-Criterion Optimization (EMO2023)*.

Chapter 2

On Interactive Evolutionary Multi-Objective Optimization Algorithms

In the previous chapter, we gave a general introduction to the field of multi-objective optimization algorithms, in particular interactive Evolutionary Multi-Objective Optimization Algorithms (iEMOAs). We also provided an overview of the problems associated with measuring, assessing, and improving iEMOAs. This chapter begins with formal definitions related to multi-objective optimization problems (MOPs) and points out the difficulties in solving these problems, including the computational complexity and loss of elitism, particularly when the number of objectives increases. We then discuss iEMOAs in detail and objective reduction techniques briefly as two approaches to solving many-objective problems in Sections 2.2 and 2.3.4. We give further details on the different characteristics of iEMOAs, namely preference modeling and preference elicitation. In Sections 2.2.1 and 2.2.2, the main classes of preference model approaches are reviewed. In Section 2.3, the problems associated with benchmarking iEMOAs are laid out, the few studies that have dealt with those problems are reviewed, and their limitations are pointed out. We continue the chapter by elaborating on Machine Decision Makers (MDMs), discussing possible improvements that could lead to more realistic simulations of decision-making behaviors, and setting the scene for the next chapters of the thesis.

2.1 Multi-Objective Optimization

In Multi-Objective Optimization Problems (MOPs), we have a set of objectives that should be optimized concurrently (Purshouse, Deb, Mansor, Mostaghim, & Wang, 2014). Let's denote the feasible space by $\mathcal{X} \subseteq \mathbb{R}^n$ and a solution, also known as a decision vector, by

$\mathbf{x} \in \mathcal{X}$. Given m objective functions ($f_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}; i = 1, \dots, m$) that map each decision vector to $\mathbf{z} = \mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ in objective space, the MOP has the following general form (Miettinen, 1999):

$$\begin{aligned} & \text{minimize}_{\mathbf{z} \in Z} \quad \mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ & \text{subject to} \quad \mathbf{x} \in \mathcal{X} \end{aligned} \tag{2.1}$$

Due to the conflicting nature of the objectives in MOPs, complete ordering of the solutions based on their objective values is not possible as in single-objective optimization. Instead, Multi-Objective Optimization Algorithms (MOAs) aim to find a well-distributed set of so-called Pareto optimal solutions.

Definition 2.1.1 (Domination). A solution $\mathbf{x} \in \mathcal{X}$ dominates solution $\mathbf{y} \in \mathcal{X}$ if the two following conditions hold (Deb, 2005a):

1. $f_i(\mathbf{x}) \leq f_i(\mathbf{y}) \quad \forall i \in \{1, \dots, m\}$
2. $f_i(\mathbf{x}) < f_i(\mathbf{y}) \quad \exists i \in \{1, \dots, m\}$

Given two solutions, if none of them dominates the other, they are said to be mutually non-dominated.

Definition 2.1.2 (Pareto Optimality). A feasible solution \mathbf{x} is said to be Pareto optimal if there is no feasible solution \mathbf{y} that dominates it. Pareto optimal solutions are also known as non-dominated or efficient solutions (Deb, 2005a). For Pareto optimal solutions, improvement in one objective is only possible at the expense of deterioration in other objective values (Branke et al., 2008).

Definition 2.1.3 (Pareto optimal set). The set of all Pareto optimal solutions, which are mutually non-dominated.

Definition 2.1.4 (Pareto front). The image of all Pareto optimal solutions in the objective space is called the Pareto Front (PF).

The aim of an MOA is to find a well-distributed subset of Pareto optimal solutions or its close approximation (Knowles & Corne, 2000b; Knowles, 2006; Coello Coello, Lamont, & Van Veldhuizen, 2007). Nevertheless, the DM is interested in achieving a small subset of Pareto Optimal solutions or even the most preferred solution (MPS) thereof rather than being presented with a large set of such solutions, which is challenging to evaluate (Li, Chen, Savić, & Yao, 2019). Pareto optimal solutions are not comparable unless additional preference information is provided to break the ties between solutions on the same front and rank solutions or to select the MPS among existing options. The preference information usually comes from

a DM with enough expertise and knowledge of the field. As a result, solving a MOP involves a decision-making phase where the DM's preference information is elicited (Roos, 2001; Miettinen, 2008). MOAs can be categorized into no-preference methods, "a priori" methods, "a posteriori" methods, and interactive algorithms based on the stage of the decision-making (Horn, 1997; Van Veldhuizen & Lamont, 2000; Miettinen et al., 2008; Huber, Geiger, & Sevaux, 2015).

"No-preference" methods aim to find neutral compromise solutions without any need to preference information from the DM. Instead, some assumptions are made about what a reasonable trade-off would be (Miettinen et al., 2008). Method of Global Criterion (Yu, 1973; Zeleny, 1973) and Neutral Compromise Solution introduced by Wierzbicki (1999) fall within this category.

As the name suggests, in "a priori" methods, the preference information is elicited from the DM before starting the optimization process. This information may be in the form of aspiration levels, goals, or weights for objective functions. Thus, weighted additive methods, lexicographic ordering, goal programming, and all scalarization methods fall into this category. Linear scalarization is a method that combines multiple objective functions into a single one by assigning weights to each objective. These weights are meant to represent their relative importance, but the approach has limitations. While it enables the use of traditional single-objective techniques, it may fail to identify unsupported solutions and the impact of the weights may not always align with their intended priority. Consequently, using weight specification to guide the search is not a straightforward process, as has been demonstrated in prior research (Steuer, 1986; Roy & Mousseau, 1996; Miettinen, 2008). Scalarization functions may also be categorized as no-preferences methods if the parameters of the scalarization function are determined without input from the DM. "A priori" methods make the queries to the DM while modeling the problem, which would bring the decision-making before the optimization phase. Not knowing the possibilities, limitations, and the form of the objective space, the DM may get too optimistic or pessimistic in defining the preference information (Branke et al., 2008; Miettinen et al., 2008). However, if the solution is satisfactory, the DM has not invested much time. Otherwise, the process is repeated with different parameters until a satisfactory solution is achieved.

In "a posteriori" methods, a subset of the Pareto optimal solutions is generated from which the DM should select a satisfying solution. Evolutionary MOAs (EMOAs) fall in this category. The problem with "a posteriori" methods is that the number of solutions required for a proper representation of the PF increases exponentially with the number of objectives, making the decision-making non-trivial (Mattson, Mullur, & Messac, 2002; Purshouse & Fleming, 2003b; Singh, Isaacs, & Ray, 2011). Besides, the DM's cognitive bottleneck will

act as a barrier when evaluating high-dimensional solutions (Li et al., 2019). Furthermore, representing the whole PF would be computationally expensive, leading to a degraded performance (Li et al., 2019), and the EMOA might even fail to get close to the actual Pareto front or cover it adequately (Miettinen et al., 2008; Brockhoff & Zitzler, 2009).

Interactive algorithms are designed to address problems with “a priori” and “a posteriori” methods, particularly the lack of efficient search and selection pressure that is highlighted in many-objective problems (Fleming, Purshouse, & Lygoe, 2005). Interactive methods elicit and exploit DM’s preference information iteratively during the optimization process to identify and limit the search to interesting parts of the PF, overcoming the lack of selection pressure and the need to generate the whole PF (Deb et al., 2010a). Because of the frequency of the stages where the preference information is elicited in interactive methods, they are more tuned to the DM’s preferences than other categories, and the DM feels more in charge and involved in the optimization and the decision-making process. Due to their popularity, many interactive methods use EMOAs as their underlying optimizer. As the core of this thesis, we will scrutinize different aspects of interactive EMOAs (iEMOAs) in Section 2.2.

2.2 Interactive Evolutionary Multi-Objective Optimization Algorithms

Unlike methods such as compromise-programming (Zeleny, 1973) and Achievement Scalarization Function Approach (Wierzbicki, 1982, 1986) which provide one PF solution in each run, EMOAs have the benefit of producing an approximation of the whole PF in one single run (Knowles & Corne, 2000a). EMOAs do not need any derivative information and are flexible, applicable, and easy to implement (Branke et al., 2008). A simulation performed by Deb, Tewari, Dixit, and Dutta (2007) illustrated that an EMOA could start from a random population and converge towards points that satisfy Karush-Kuhn-Tucker conditions. Therefore, evolutionary algorithms are “convenient” in many contexts as a choice that works well without much overhead to tailor them to MOPs (Deb, 2008) and widely accepted in the literature (Corne, Jerram, Knowles, & Oates, 2001; Coello Coello, 2002, 2006; Deb, Kalyanmoy, 2007; Ishibuchi, Tsukamoto, & Nojima, 2008). However, EMOAs are not exempted from the problems associated with “a posteriori” methods and many of them, particularly those relying on the non-dominated sorting strategy proposed by Goldberg (1989), fail to provide proper performance when the number of objectives exceeds three or four and are almost useless when it exceeds ten (Khare, Yao, & Deb, 2003; Deb, Thiele, Laumanns, & Zitzler, 2005; Brockhoff & Zitzler, 2006; Li, Deb, & Yao, 2018).

Interactive EMOAs iteratively elicit and exploit the discrimination and preferences of the DM to increase selection pressure and break ties between solutions on the same front (Branke et

al., 2008; Purshouse, Jalbă, & Fleming, 2011; Tomczyk & Kadziński, 2020a). To do so, iEMOAs alternate between the decision-making and optimization phases to support the DM in reaching a desirable result without investing too much cognitive effort (Miettinen et al., 2008). In the decision-making phase, the preference information is elicited from the DM to guide the search in the subsequent optimization phase towards preferred areas of the feasible space or, in the last step, to select the MPS. During the process, the DM can learn more about his preferences and become more knowledgeable about the limits and bounds of the problem (Miettinen et al., 2008). In other words, iEMOAs help the DM justify their expectations and achieve a satisfying solution, a process referred to as psychological convergence (Miettinen, 2008). Ideally, interactive EMOAs exploit the elicited preference information to limit the search to regions of interest (Wang, Li, Zhang, Hu, & Shen, 2019), thus incurring lower computational costs (Geiger & Wenger, 2007) and reducing the cognitive effort required from the DM. Such characteristics have motivated the development of many iEMOAs in the past decades (Wang et al., 2019).

Two main characteristics of any interactive EMOA that can be used to differentiate them are the preference model and interaction/elicitation style (Xin et al., 2018; Tomczyk & Kadziński, 2020a). The preference model refers to how the DM's preferences are modeled and exploited inside the engine to direct the search. Interaction style defines the type of preference information required from the DM and the way the preference information is elicited. In what follows, we briefly consider these two aspects as they play a prominent role in characterizing any iEMOA.

Different classifications for different preference models are used in the literature. Miettinen et al. (2008) summarize all such methods in three broad categories:

1. **Trade-offs:** Trade-offs can be defined as the gain in the value of one objective function in compensation for the loss in the value of other objective function(s). In this regard, the subjective trade-off is defined as the willingness of the DM to accept deterioration in the value of some objective function(s) to achieve improvements in the values of another. In iEMOAs, the subjective trade-offs can be directly elicited from the DM or can be selected by the DM from a set of adequately represented trade-offs. The algorithm directs the search toward the preferred area of the PF by making the determined trade-offs. As a result, the optimization process leads to solutions with better quality (often characterized by the solution's utility value). UFs are a popular tool for modeling such subjective trade-offs. Examples of such methods can be found in BCEMOA (Battiti & Passerini, 2010), the Zionts-Wallenius (Z-W) method (Zionts & Wallenius, 1976), and the SPOT method (Sakawa, 1982).
2. **Reference Points:** In reference-point-based methods, the DM's desired solution is mod-

eled in terms of a reference point (in the form of aspiration or reservation levels), (Wierzbicki, 1980), and solutions closer to such an assumed reference point are prioritized (Figueira, Liefvooghe, Talbi, & Wierzbicki, 2010). Such preference information can be used to approximate the partial ordering of a population and to find solutions around the specified points. Such methods are applied in the Tchebycheff method (Steuer, 1986), Pareto Race (Korhonen & Laakso, 1986), GUESS (Buchanan & Corner, 1997) and REF-LEX method (Miettinen & Kirilov, 2005). In these methods, the initial solutions should have a good diversity to represent the PF properly (Huber et al., 2015). It should be noted that some initial solutions presented to the DM may lead to the anchoring effect (Tversky & Kahneman, 1974).

3. **Classification:** Here, the objective functions are classified into objectives that need improvement, those that can be deteriorated, and objectives whose values are already satisfactory. The optimization process then adapts to such preferences to improve the solution. According to Larichev (1992), such preference elicitation is easy for the DM. NIMBUS introduced in (Miettinen & Mäkelä, 2006; Miettinen, Mustajoki, & Stewart, 2014) and STEM (Benayoun, De Montgolfier, Tergny, & Larichev, 1971) fall into this category.

Preference elicitation and interaction style can generally take two forms (Tomczyk & Kadziński, 2020b). Direct preference elicitation requires the DM to identify some parameters of the preference model directly, which can be in the form of the reference point (aspiration level/goal) (see, e.g., PBEA (Thiele, Miettinen, Korhonen, & Molina, 2009), WASF-GA (Ruiz, Luque, Miettinen, & Saborido, 2015b)), reservation levels (González-Gallardo, Saborido, Ruiz, & Luque, 2021), and weights (see, e.g., R-NSGAII (Deb & Sundar, 2006)), among others. On the other hand, in indirect approaches, DM is required to provide some holistic judgments, which tend to be less demanding, mainly in the form of exemplary decisions. When indirect approaches are used, the DM is not required to have prior knowledge of solution space and the optimization algorithm (Jacquet-Lagrange & Siskos, 2001). Indirect queries can be in the form of pairwise comparisons (Battiti & Passerini, 2010; Branke et al., 2016; Tomczyk & Kadziński, 2019b, 2019d; Benabbou, Leroy, & Lust, 2020), selecting the best among a small subset of solutions (Fowler et al., 2010a; Köksalan & Karahan, 2010; Tomczyk & Kadziński, 2021b), accepting or rejecting a presented trade-off (Zionts & Wallenius, 1983), or ordering a subset of solutions (Sinha, Malo, & Kallio, 2018; Shavarani, López-Ibáñez, Allmendinger, & Knowles, 2023). Generally, iEMOAs that require the DM to rank a subset of solutions are known as ranking-based iEMOAs. Providing indirect information imposes less cognitive effort on the DM (DeShazo & Fermo, 2002; Branke, Greco, Słowiński, & Zielniewicz, 2010). Based on the type of information required from the DM,

iEMOAs can be divided into ad-hoc and non-ad-hoc methods. Non-ad-hoc methods are those methods where the DM can be simulated by a UF (Stewart, 1996). Accordingly, ad-hoc methods can be defined as those where the responses to the queries from the algorithm cannot be simulated by the use of a UF (Miettinen, 2001; Afsar et al., 2021). The focus of our study pivots around ranking-based non-ad-hoc iEMOAs. The next two sections provide a brief review of well-known iEMOAs classified into ad-hoc and non-ad-hoc.

2.2.1 Ad-hoc iEMOAs

Ad-hoc methods are those methods in which the queries to the DM cannot be simulated using a UF (Steuer, 1986). Due to their simplicity, many ad-hoc methods use reference points to model DM's preferences (Deb & Sundar, 2006; Ruiz et al., 2015b). It is naturally convenient for DMs to express their desires in form of reference points or aspiration levels (Luque, 2015). Usually, the reference points are used to favor solutions that are closer to them using a scalarization function (Thiele et al., 2009). The scalarization functions can take many forms. For instance, Ben Said, Bechikh, and Ghedira (2010) use Euclidean distance as a scalarization function to favor solutions closer to the expressed aspiration levels. It is also possible to formulate and optimize several scalarization functions simultaneously and leave the decision to the DM to select among a set of final solutions (Miettinen & Mäkelä, 2000, 2006).

Other ad-hoc methods ask the DM for reference-points (López-Jaimes & Coello Coello, 2014; Ruiz, Cabello, Platero, & Blázquez, 2015a), bounds on the objectives (Ruiz, Ruiz, Miettinen, Delgado-Antequera, & Ojalehto, 2019), preferred range (Sindhya, Ruiz, & Miettinen, 2011), weights (Wang, Purshouse, & Fleming, 2013; Narukawa et al., 2016a), direction of improvements (Miettinen, Podkopaev, Ruiz, & Luque, 2015), and relative importance of objectives (Shen, Guo, Chen, & Hu, 2010). A review of ad-hoc methods can be found in (Xin et al., 2018). The mechanism and the required tools for measuring ad-hoc methods are different from non-ad-hoc methods, where a UF can be used to simulate a DM. This thesis focuses on non-ad-hoc methods; thus, a detailed discussion of ad-hoc methods is out of the scope of this study.

2.2.2 Non-ad-hoc iEMOAs

Application of Multi-Attribute Utility Theory (MAUT) (Keeney & Raiffa, 1993) has been predominant in preference modeling, and utility functions (UFs) have been a popular way to represent DM's interest in a particular solution (Rachmawati & Srinivasan, 2006; Afsar et al., 2021). Non-ad-hoc methods assume the existence of an underlying UF that drives the decisions of the DM (Steuer & Gardiner, 1991). Thus, these methods attempt to estimate

such a UF in line with DM's decisions (Jaszkiewicz, 2004; Quan et al., 2007a; Battiti & Passerini, 2010). The estimated UF may be used as a scalarization function to transform the problem into a single objective problem, as a criterion to break ties between solutions on the same front, or to automate ranking or pairwise comparisons. One of the first studies in this category was done by Zionts and Wallenius (1976), who tried to use the weight space reduction technique to estimate a linear UF compatible with DM's preferences. This technique was later extended to handle pseudo-concave UFs (Zionts & Wallenius, 1983). Phelps and Köksalan (2003) use DM's preferences to estimate and optimize a linear UF compatible with DM's preferences. Similarly, in the study by Jaszkiewicz (2004), DM's pairwise comparisons are used to draw a weight vector that will constitute a linear scalarization function. After each interaction, an EA is used to find and update the weight vector. Besides estimating the DM's preferences with a linear UF, a drawback of these studies is the singularity of the estimated UF, which limits the EMOA's capability to find several solutions with interesting trade-offs (Branke et al., 2010).

Branke et al. (2010) proposed the Necessary-preference-enhanced Evolutionary Multi-objective Optimizer (NEMO-I), in which a set of linear UFs compatible with DM's pairwise comparisons is derived using GRIP. Then the learned UFs are used to rank the solutions such that solution x is necessarily at least as good as solution y if all the generated UFs imply this. On the other hand, if not all UFs confirm such a relation, it is considered that x may be as good as y . A modified version of NSGAII (Deb, Pratap, Agarwal, & Meyarivan, 2002) is then used to find all PF solutions compatible with the learned UFs.

While most non-ad-hoc methods use the learned UF to rank solutions, Quan et al. (2007a) proposed an iEMOA that learns a linear UF for determining the dominance relationships among any two solutions in pairwise comparisons by converting preferences of the DM to the constraints. Battiti and Campigotto (2010) discussed the application of online learning techniques for learning a weighted aggregation of objectives compatible with DM's pairwise comparisons. The weight vector is constructed by drawing linear constraints on the weight space.

Modeling DM's preferences linearly are only suitable for convex problem (Wang, Zhang, & Zhang, 2016). When the DM's preferences are nonlinear, a linear estimation of the UF may be misleading and not result in satisfactory solutions (Deb et al., 2010a). In the past years, preference learning techniques have been improved from learning a basic linear UF to estimating complex functions which are more compatible with DM's preferences.

Branke et al. (2016) proposed NEMO-II, a revised version of NEMO-I that can change the preference model from linear to Choquet integral if the linear model is not sufficiently compatible with DM's decisions. The new version also addresses necessary preference relations

calculation cost in NEMO1.

Similarly, the Decomposition-Based Interactive Evolutionary Algorithm for Multiple Objective Optimization (IEMO/D) (Tomczyk & Kadziński, 2019a) uses DM's pairwise comparisons to detect upper and lower limits for weight vectors and construct compatible L-norm value functions. IEMO/D extends a basic variant of MOEA/D (Zhang & Li, 2007). Greco, Matarazzo, and Słowiński (2010) apply the Dominance-based Rough Set Approach (DRSA) to the DM's preference information to draw some "if-then" rules on the values of the objective functions. The rules are later used to estimate the quality of the solutions.

Köksalan and Karahan (2010) introduced iTDEA based on their earlier multi-objective evolutionary optimization algorithm: TDEA (Karahan & Köksalan, 2010). Their algorithm relies on the *territory* parameter, which controls the number of solutions in different regions of the PF. The territory is defined as a region occupied by each individual in the objective space and controls the density of the Pareto front around a given solution. New offspring are accepted into the population if only they do not violate the territory of their closest individual in the population. In iTDEA, the DM selects the most preferred solution among some filtered solutions at each interaction. Filtering is based on the favorable weights (Köksalan & Phelps, 2007). DM's preferences are considered by defining a smaller territory for the solutions in the proximity of the selected solution.

Brain-Computer Evolutionary Multi-Objective Algorithm (BCEMOA) (Battiti & Passerini, 2010) does not make any particular assumptions on the shape of the UF and uses the Support Vector Machine (SVM) to estimate the preference model of the DM. However, the shape and the degree of the UF are limited to those specified beforehand. In each interaction, DM ranks a small subset of non-dominated solutions which are considered the training set and are labeled by the given ranks. The training set and its labels are used to train the SVM.

Similar to BCEMOA, Pedro and Takahashi (2014) proposed an Interactive Non-dominated Sorting algorithm with the Preference Model (INSPM), which is based on NSGAII and uses the neural network instead of SVM to construct the UF.

Brockhoff, Hamadi, and Kaci (2014) introduced an interactive W-HypE algorithm with a weighted hyper-volume indicator approach. Its main idea is to define a weight function on the objective space and use the contribution to the weighted hyper-volume indicator as the fitness of each solution within the EMO algorithm (For earlier studies on non-interactive hyper-volume based EMOAs see (Knowles, 2002; Knowles & Corne, 2003; Knowles, Corne, & Fleischer, 2003)). They compare the algorithm's performance with iTDEA, which requires the DM to select the best solution from a given set. In their experiments, DM was simulated by a Tchebychef function.

Although these algorithms eliminated the linearity assumption of the UFs, there are still limited by their underlying assumptions and the limited types of UFs they can estimate. Besides, many of them require objective values to be normalized and scaled, which is not an easy task, specifically if the scaling parameters should be determined a priori (Branke et al., 2010). On the other hand, experimental studies on non-ad-hoc iEMOAs (López-Ibáñez & Knowles, 2015; Shavarani et al., 2021) have revealed that the proposed methods fail to perform well when tested under real-life conditions, or their performance deteriorates with biases and non-idealities that might happen during interactions. Thus, it is vital to scrutinize different elements of iEMOAs when confronted with realistic conditions and identify the weaknesses and strengths of each algorithm. Such extensive experiments would allow researchers to systematically address the problems with iEMOAs and move toward realizing the potential of the iEMOAs

2.3 Benchmarking Non-ad-hoc iEMOAs

There have been many studies on the development of iEMOAs in recent years. However, the problem of measuring and benchmarking their performance remains troublesome. Thus, there is no concrete information on their strengths, weaknesses, and how well the proposed algorithms perform under realistic conditions, (Miettinen, 1999; Ojalehto, Podkopaev, & Miettinen, 2016; Shavarani et al., 2021). Further improvements in this field require a well-defined design of experiments wherein all external sources of noise and variation can be controlled.

Experimenting with iEMOAs is not straightforward as they are steered by the decisions of the DM and the results are subjective and affected by the preferences of that DM. Thus, it is not easy to determine if any significant observation should be attributed to the performance of the algorithm or the DM's decisions. Due to such problems, only 45 studies have been performed on the performance of iEMOAs (Afsar et al., 2021). A review of such experiments can be found in Xin et al. (2018) and Afsar et al. (2021). One approach to cancel out these biases is to run experiments with a sufficiently large number of different DMs. One study has estimated that on average 65 decision makers are required to cancel such biases (Zujevs & Eiduks, 2011). Thus, experimenting with human DMs is expensive and not affordable. As a result, the majority of the few experiments with human DMs are conducted by the participation of students or the authors of the algorithms themselves (Afsar et al., 2021). The involvement of students in the experiments would bias the results as long as they are not responsible for the outcomes. Also, having a single DM in the experiments (either the author or else) would raise human-specific issues such as learning or fatigue (Zujevs & Eiduks, 2008). Besides,

the involvement of humans in the experiments makes the results non-replicable (Chen et al., 2017). An alternative approach is to replace human DMs with realistic machine DMs. In this regard, the main concern is to identify a reliable approach to simulate decision-making behaviors in the experiments (Luque, Caballero, Molina, & Ruiz, 2007; Belton et al., 2008; Huber et al., 2015).

2.3.1 Machine Decision Makers

To enable experimenting with iEMOAs and to avoid expensive setups with human DMs, it is possible to replace human DMs with a reliable simulation of decision-making behaviors. Such a simulation is frequently called artificial or machine decision-makers (Afsar et al., 2021). There have been several studies introducing Machine Decision-Makers in the recent literature (López-Ibáñez & Knowles, 2015; Barba-González et al., 2018). As different approaches use different preference models, an MDM should be consistent with the type of information required from the DM in the interactive algorithm under study (Purshouse et al., 2014). In this regard, iEMOAs can be categorized into non-ad-hoc and ad-hoc methods based on whether the DM's responses to the queries projected in the interactions can be simulated by a UF or not, respectively (Miettinen, 1999). Experiments on ad-hoc methods require a different setup as the utilization of UF is not possible. For instance, Huber et al. (2015) simulates the preferences of the DM using a predefined MPS and simulates DM's learning by progressively narrowing down the cone which is formed based on the MPS. Other examples of MDMs for ad-hoc methods can be found in (Zujevs & Eiduks, 2011), (Afsar et al., 2021), and (Ojalehto et al., 2016). The main focus of this research is dedicated to the ranking-based non-ad-hoc approaches. Thus, the details of such experiments are not discussed here.

In experiments on non-ad-hoc methods, it is possible to simulate the DM's responses to the algorithm's queries using a UF (Steuer, 1986). The choice of the UF itself is important as any arbitrarily selected function may not adequately reflect realistic decision-making behaviors and can make the experimental results biased and unreliable (Shavarani et al., 2021). Furthermore, it is not realistic to assume ideal conditions where the DM has complete knowledge of the problem at hand, makes no mistakes or contradictory decisions, and no biases or non-idealities happen inside the DM's mind or during the machine-DM interactions (Afsar et al., 2021). Thus, as illustrated in Figure 2.1, an MDM is generally comprised of two equally important units: a UF and the simulation of non-idealities, human factors and biases. During the interactions, the iEMOA can send its queries to the MDM. The quality of the solution is evaluated using the UF. To make the simulation more realistic, the utility value given by the UF and the objective values themselves can be disturbed to simulate different types of human factors and biases such as inconsistencies in the decisions. The intensity of such biases can

be controlled within this unit.

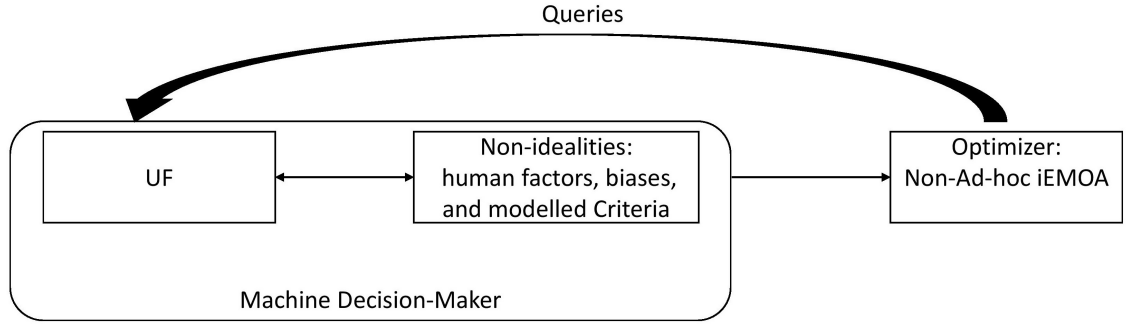


Figure 2.1. A typical MDM framework used for simulation of decision-making behaviors on experimenting with non-ad-hoc iEMOA.

One of the first MDM frameworks was suggested by López-Ibáñez and Knowles (2015). They simulated omitted (hidden) objectives, Mixed (preferentially dependent) objectives and inconsistencies in the decisions. However, their experiments were limited to linear utility functions. Later, the framework was improved using a sigmoid utility function (Shavarani et al., 2021). In what follows we will further discuss the MDM framework and its two main units.

2.3.2 Utility Function

Due to their simplicity and completeness, UFs have been frequently used in the literature to represent DM's preference model (Debreu, 1987; Huber et al., 2015). In this regard, linear functions (Reeves & Gonzalez, 1989; Fowler et al., 2010b; Köksalan & Karahan, 2010), non-linear functions (typically quadratic or polynomial functions) (Mote, Olson, & Venkataramanan, 1988; Reeves & Gonzalez, 1989; Battiti & Passerini, 2010; Köksalan & Karahan, 2010), exponential (Luque, Miettinen, Eskelinen, & Ruiz, 2009), Gaussian function (Pedro & Takahashi, 2014; Narukawa et al., 2016a) and Tchebycheff UFs (Köksalan & Karahan, 2010) have been frequently used. However, most of these UFs do not satisfy the characteristics of the utility function that are required for the simulation of decision-making behaviors (Kahneman & Tversky, 1979; Stewart, 1996), and there is no evidence that they can adequately simulate the complex behaviors of a decision-maker in the experiments (Shavarani et al., 2021). Additive Independence (Keeney, 1981), risk aversion behavior, trade-off interpretability and compensatory preferences (Stewart, 1996), the convexity of losses and concavity of gains (Kahneman & Tversky, 1979) are some of the assumptions that should be observed when simulating a DM in the experiments. For instance, linear UFs simulate constant trade-offs across all the objective values, which is not realistic (Huber et al., 2015). The research by Kahneman and Tversky (1979) revealed that Sigmoid UFs can adequately represent decision-making behaviors. The Sigmoid UFs satisfy the accepted assumption in

the economic literature, which states that decisions of a human DM follow an S-curve shape which is convex for losses and concave for gains (Mas-Colell et al., 1995). A specific Sigmoid UF was proposed by Stewart (1996) in his experiments on the robustness of additive value functions. The formula of this UF, hereafter called Stewart UF, is illustrated in Equation 2.2. In this equation, $w_i \in [0, 1]$ is the relative weight of the i^{th} objective function ($\sum_{i=1}^m w_i = 1$) and $u_i(z_i)$ is its marginal Stewart UF. Reference level (τ_i), is the value of the i^{th} objective where the function changes from concave to convex. In other words, τ_i is the threshold that separates the objective values that are perceived as “gains” or “losses” (satisfactory or unsatisfactory) from the DM’s perspective. λ_i is the value of the marginal UF at the reference level, i.e. $\lambda_i = u_i(\tau_i)$. Parameters α_i and β_i control the non-linearity of the function for the i^{th} objective. In accordance with prospect theory which claims improving an unsatisfactory criterion is more appealing than improving a satisfactory one, the curvature and slope of the curve are greater for gains than losses. This rule is observed in the Stewart UF by setting $\alpha > \beta > 0$.

$$U(\mathbf{z}) = \sum_{i=1}^m w_i u_i(z_i), \quad u_i(z_i) = \begin{cases} \frac{\lambda_i \cdot (e^{\alpha_i z_i} - 1)}{e^{\alpha_i \tau_i} - 1} & \text{if } 0 \leq z_i \leq \tau_i \\ \lambda_i + \frac{(1 - \lambda_i)(1 - e^{-\beta_i(z_i - \tau_i)})}{1 - e^{-\beta_i(1 - \tau_i)}} & \text{if } \tau_i < z_i \leq 1 \end{cases} \quad (2.2)$$

Stewart UF is able to simulate four decision-making behaviors by different combinations of τ_i and λ_i values as described in Table 2.1. Figure 2.2 illustrates the marginal Stewart UF for different values of its parameters.

Table 2.1. Description of different types of DM behaviors simulated by combinations of τ_i and λ_i adapted from (Stewart, 1996).

Type	τ_i	λ_i	Description
1	[0.1, 0.4]	[0.1, 0.4]	Limited range of compensation; plus sharp preference threshold
2	[0.1, 0.4]	[0.6, 0.9]	Limited range of compensation.
3	[0.6, 0.9]	[0.1, 0.4]	Mainly compensatory preferences, but with sharp preference threshold.
4	[0.6, 0.9]	[0.6, 0.9]	Mainly compensatory preferences.

In this research, we propose using the Stewart function in experimenting with iEMOAs. Stewart UF is designed for maximization problems. When experimenting on iEMOAs, many of the existing benchmark problems are formulated as minimization problems. Thus, the UF cannot be used for minimization problems without violating its underlying assumptions.

As a result, one should be cautious about using the Stewart UF for the minimization problems and make sure the assumption of the UF are observed. In Chapter 3, we propose a proper transformation of Stewart’s UF that can be used with minimization problems while respecting all of its assumptions.

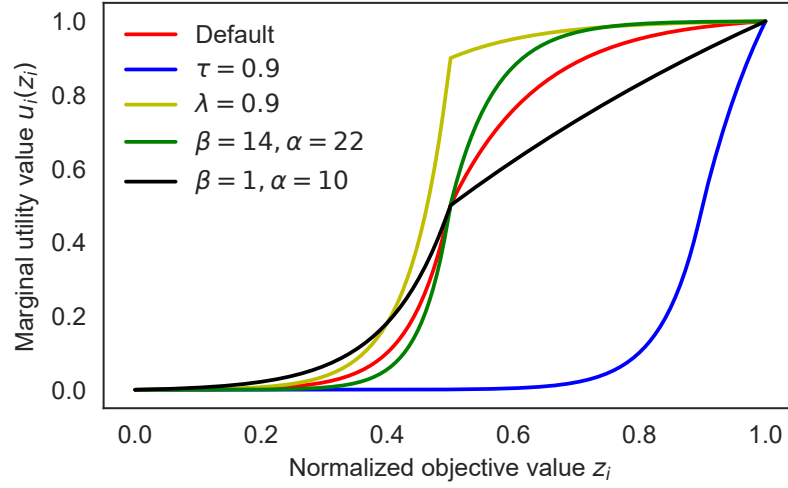


Figure 2.2. Shape of Stewart UF with different parameters. In “Default” $\tau = 0.5, \lambda = 0.5, \alpha = 16, \beta = 7$. The effect of changing each parameter from the default value on the shape of the function is also illustrated in other curves.

2.3.3 Simulation of Non-idealities

Humans are prone to mistakes. Thus, any simulation of their behaviors should also reflect such biases. A sound decision support system is supposed to account for inconsistencies in the decisions of the DM (French, 1984). Thus, it is vital to simulate such inconsistencies and non-idealities when benchmarking interactive methods and measure their robustness towards these non-idealities. It should be noted that biases may happen anywhere during human-machine interactions, within DM’s mind, or even in the modeling phase.

Stewart (2005) indicates that decisions of the DM may be affected by external and internal uncertainty. External uncertainty can be defined as the lack of proper insight into the consequences of the decision. Internal uncertainty is the lack of accuracy in the preferences, which leads to inconsistencies in the decisions. The research on iEMOAs has mainly considered the latter.

Some studies add Gaussian noise to the value returned by the UF (Campigotto & Passerini, 2010; Köksalan & Karahan, 2010) to simulate inconsistencies in the DM’s decisions. Phelps and Köksalan (2003) evaluate the performance of their proposed Interactive Evolutionary Meta-heuristic (IEM) using linear and Tchebychef UF without simulating any biases. They compare the algorithm’s performance with a Perfect Information Case where the UF is optimized directly by the optimizer as a single objective problem. Later, Jaszkievicz (2004) compare their IPMA method with IEM using Tchebychef, linear and quadratic UFs. They also evaluate the performance of the two algorithms with a perfect information case, averaged over 10 runs as a baseline. They simulate indifference threshold and inconsistencies in DM’s decisions with uniformly distributed noise. To simulate indifference, they assume the DM may not compare two solutions if the difference in their utility value falls below a given

threshold. (Kornbluth, 1985) assumes there exists a range of utility values for which the DM can not make a good evaluation.

Another bias in this context is the change in preferences, termed as *preference drift* (Pu & Chen, 2008). Such a change mainly results from the *learning* (Campigotto & Passerini, 2010) and *change in the DM's attitude*, which happens during the optimization process. Learning itself (Weber, 1987) can be described as increased knowledge of the problem during the process, which may lead to the change in the preferences (Belton & Elder, 1994). The preference change is troublesome as it causes contradictions in the preference information elicited in different interactions. Other non-idealities that have been studied in the literature include fatigue (Zujevs & Eiduks, 2008), indifference (Zionts, 1981), hesitation (Huber et al., 2015), incomplete information (Lerch & Harter, 2001), cognitive bottleneck (Tversky & Kahneman, 1974; Stewart, 2005) and anchoring (Buchanan & Corner, 1997).

Only a few studies have proposed MDMs that include simulations of the biases and non-idealities within their framework (Ojalehto et al., 2016; Shavarani et al., 2021) the first of which was the study by López-Ibáñez and Knowles (2015). Based on the non-idealities introduced by Stewart (Stewart, 1996), López-Ibáñez and Knowles (2015) proposed a Machine-Decision-Maker (MDM) framework to serve as a laboratory for analyzing the performance of interactive algorithms, analyzing BCEMOA (Battiti & Passerini, 2010) as a proof of concept. The simulated biases include inconsistencies in the decisions, mixed and omitted objective (López-Ibáñez & Knowles, 2015). However, the study used only linear UFs to represent DM's decisions.

In this study, we will simulate inconsistencies, preferentially dependent objectives, irrelevant objectives and hidden objectives. As a contribution of this thesis, we provide formal definitions and a simulation approach for the last two non-idealities. We also propose approaches to identify and deal with such non-idealities in Chapter 4, making the performance of the iEMOAs robust and improved.

2.3.4 Hidden and Irrelevant Objectives

Generally, in the modeling phase, there may be the tendency to model as many objectives as possible without noticing that they might be non-conflicting or highly correlated and hence redundant (Sinha, Saxena, Deb, & Tiwari, 2013). Each objective function adds exponentially to the complexity of the problem, both in the optimization phase and decision-making phase (Jensen, 2003; Allmendinger, Jaszkiwicz, Liefoghe, & Tammer, 2022). Thus, many researchers have discussed possible approaches to decrease the number of objectives without losing the interesting parts of the Pareto front (Brockhoff & Zitzler, 2006). Particularly, in

the context of interactive methods, it is vital to ensure the right set of objective functions is considered to avoid the curse of dimensionality and increase the accuracy of preference learning.

In this regard, objectives that can potentially be eliminated are divided into two categories:

- *Redundant*: There exists a minimal set of objectives that are enough for generating a close representation of the Pareto front. We follow the tradition in literature and call objectives *redundant* when they can be eliminated with regard to the structure of the problem and preserving the non-domination relations (Gal & Leberling, 1977; Agrell, 1997). Saxena, Ray, Deb, and Tiwari (2009) indicate that the redundancy of the objectives occurs when objectives are not conflicting or their elimination does not significantly affect the PF structure.
- *irrelevant*: In this study, objectives that are considered by the optimizer but are not relevant to the DM are termed *irrelevant* regardless of the PF and non-dominance relations. The formal definitions are provided in Chapter 4.

While the studies in the first category consider removing objectives that are highly correlated or those that do not contribute significantly to non-dominance relations and the shape of the PF (Guillén-Gosálbez, 2011a; Pozo, Ruíz-Femenia, Caballero, Guillén-Gosálbez, & Jiménez, 2012), the second category aims to do so by eliminating the objectives that are not considered by the DM when evaluating the solutions and making decisions. On the other hand, there might exist objectives that are considered by the DM but are hidden to the optimizer (Shavarani, López-Ibáñez, & Allmendinger, 2022). An example of hidden objectives is given in (Kryston et al., 2022), where a stakeholder is unwilling to reveal his preferences due to strategic reasons; thus, the relevant set of objectives remains hidden. Such discrepancy between the model and the DM's preferences will make the results unsatisfactory (Battiti & Campigotto, 2010). Thus, it is desired to design intelligent algorithms that detect irrelevant and hidden objectives, eliminating the former while replacing them with the latter. To the best of our knowledge, there is no prior study that has done so, and in Chapter 4 we propose such an approach which exploits the preference information elicited during optimization in ranking-based iEMOA to detect irrelevant and hidden objectives and updates the set of objectives that are optimized. We also consider applying the proposed method as a DM-oriented objective reduction method. As there are no prior studies on eliminating irrelevant objectives, the next section investigates some of the most important studies on objective reduction techniques that have considered eliminating redundant objectives.

Previous Research on Objective Reduction

The studies in the literature are limited to the reduction of redundant objectives without considering DM's preferences. A brief description of the main studies in redundant objective reduction is given here categorized based on their employed techniques. The reduction of irrelevant objectives is a novel contribution of this study and will be discussed in detail in Chapter 4. The redundant objective reduction techniques have been applied either “a priori” to facilitate the optimization process or “a posteriori” to assist the DM in decision-making. The proposed techniques in the literature either consider correlations among objectives or their importance in preserving the dominance relations among solutions. We acknowledge there are reduction methods that attend to the dimensions of the decision space, such as those discussed in (Allmendinger & Knowles, 2010; Chu, Zhang, Fu, Li, & Zhou, 2015; Zhao et al., 2020), as well as those that focus on constraint reduction (Saxena & Deb, 2007b, 2008). However, they fall out of the scope of this research as our focus is on the objective space. Many of these studies use the term *dimension* reduction for the same concept. However, to avoid confusion with methods that reduce the number of decision variables (Allmendinger & Knowles, 2010), we use the term “objective reduction”.

Subset selection & preserving Pareto front solutions: Gal and Leberling (1977) proposed one of the earliest *a priori* objective reduction approaches for linear programming models while preserving the set of Pareto optimal solutions. Although the method was later extended by Agrell (1997) to include a non-linear model considering the correlation among objectives with a probabilistic test of redundancy, they both make exhaustive assumptions about the problem structure that is impossible to validate for real-life problems. Brockhoff and Zitzler (2007a) introduce two minimum objective subset selection (MOSS) methods: δ MOSS finds a minimal set of objectives with a maximum error of δ and k -EMOSS finds a subset with fixed size k having the minimum error. The error here is a measure of change in the set of PF solutions. In (Brockhoff & Zitzler, 2007b), the authors use their minimal set approach online in the simple indicator-based evolutionary algorithm (SIBEA) (Zitzler, Brockhoff, & Thiele, 2007). As MOSS problems are computationally expensive, Guillén-Gosálbez (2011b) proposed a linear programming model for k -EMOSS problem, which was demonstrated to have acceptable performance. López Jaimes, Coello Coello, and Chakraborty (2008) use an unsupervised feature selection technique (Mitra, Murthy, & Pal, 2002) for preserving the most conflicting objectives. Later, this technique was incorporated into NSGA-II (Deb et al., 2002) to construct an online reduction method (López Jaimes, Coello Coello, & Urías Barrientos, 2009). Singh et al. (2011) also focus on the Pareto corner points to identify which objectives are sufficient to reproduce the PF. Some drawbacks of their approach include not capturing the whole PF, losing objectives that are contributing somewhere on the PF away from corner

points, and possibly overestimating dimensions.

Recombining non-conflicting objectives: Some studies try to identify non-conflicting objectives that can be aggregated into a single objective. Freitas, Fleming, and Guimarães (2013) used the harmonic level, introduced in (Purshouse & Fleming, 2003a, 2007), to identify objectives that can be merged into a new compound scalarized objective with minimal effect on the PF. As opposed to conflict, there is “harmony” among objectives when improvement of either of them does not lead to deterioration of the other (Giagkiozis, Purshouse, & Fleming, 2013). Like other objective reduction methods, the DM preferences are not considered in the reduction process in (Freitas et al., 2013). However, the DM can decide on the final number of objectives “a priori”. Similarly, de Freitas, Fleming, and Guimarães (2015) use aggregation trees to identify harmonious objectives.

PCA & Correlation among objectives: One of the first objective reduction methods based on linear PCA was proposed by Deb and Saxena (2006) that was implemented within the NSGA-II framework. Linear PCA relies on the correlation among objectives to detect conflicting objectives, thus it does not guarantee that the dominance relation is preserved (Brockhoff & Zitzler, 2009). Furthermore, linear PCA does not provide good performance for non-linear data, and it might fail to identify all non-conflicting objectives (Lygoe, Cary, & Fleming, 2010). To address these issues, two more methods were introduced in (Saxena & Deb, 2007a), which used correntropy PCA (Xu, Pokharel, Paiva, & Príncipe, 2006) and maximum variance unfolding as was previously used in (Weinberger, Sha, & Saul, 2004; Weinberger & Saul, 2006). PCA was also used in (Goel et al., 2007), where the method was validated on a problem with 4 objectives only. PCA-based methods are applied to non-dominated solutions and thus their performance relies heavily on the proper approximation of the PF (Singh et al., 2011).

Visualization Some studies suggest the identification of redundant objectives through visualization (Mattson & Messac, 2003; Obayashi & Sasaki, 2003; Koppen & Yoshida, 2007; Fieldsend & Everson, 2013). However, most of these studies aim to facilitate decision-making and intuitive visualization of the solutions rather than the optimization process itself and are limited in the number of objectives they can handle. Thus, they use objective reduction methods such as PCA to enable the visualization and hence bring new insights to the DM.

Drawbacks of Existing Methods

The problems with existing objective reduction algorithms as outlined above can be summarized as follows:

- **Dependency on non-dominated solutions:** Many existing dimension reduction techniques rely on a properly represented PF which itself is troublesome in the first place when the number of objectives is high.
- **Computationally expensive:** As discussed, most of the introduced methods, including PCA-based and subset selection methods, are computationally expensive and/or can't be applied to many-objective problems.
- **In their study, Costa and Oliveira (2010) have demonstrated that objectives that are deemed redundant by PCA may be “informative”, i.e., contain trade-off information that would be lost if omitted.**
- **Offline reduction:** Most of the existing techniques consider dimension reduction in an offline mode, whether a priori to facilitate the optimization process or a posteriori as a decision-making aid.
- **Not exploiting preference information:** None of the existing methods consider the preferences of the DM in their dimension reduction techniques. There are methods that ask the DM about some parameters, for instance cardinality of the reduced set, but none use the preferences for reduction purposes.

2.4 Improving Robustness and Performance of iEMOAs

2.4.1 Detection and Handling Hidden and Irrelevant Objectives

In interactive methods, it is possible to exploit and analyze the preferences of the DM that are elicited during the interactions to detect the objectives that the DM does not consider in evaluating the presented solutions. For the first time in the literature, In Chapter 4 we introduce an efficient approach based on uni-variate feature selection and recursive feature elimination to identify irrelevant and hidden objectives. The proposed method is computationally efficient, has reasonable accuracy, and effectively reduces the number of objectives and objective evaluations. The main goal of feature selection is The importance of feature selection and dimensionality reduction in overcoming the curse of dimensionality is frequently highlighted in the machine learning literature (Jain & Zongker, 1997; Wang, Tang, & Liu, 2017). Simply put, the curse of dimensionality states that the learning accuracy is significantly decreased with a higher number of features. The main goal of feature selection is to identify the minimal set of features that results in a model that has the same (or at least close) accuracy to that trained on the full set of features, assuming that there are objectives that can be eliminated without having much of an impact on the learned model. This task is even

more vital in iEMOAs where the size of the training data used for preference learning and modeling is also limited by the queries to the DM and his cognitive bottleneck. The objective reduction also helps visualization and model interpretability which are desired characteristics of any iEMOA.

2.4.2 An Improved Preference Learning Technique

As outlined in Section 2.2.2, most non-ad-hoc methods assume a linear UF when modeling the DM's preferences (Zionts & Wallenius, 1976; Phelps & Köksalan, 2003). Linear models are only suitable for convex problems. Otherwise, the algorithm is biased towards areas of the PF that might not include the MPS, or a satisfactory solution (Deb et al., 2010a; Wang et al., 2016). Others that overcome this limitation are either limited by the number of UF types that can be estimated or by extensive assumptions that are not easily verified in real-life problems, if possible at all (Zionts & Wallenius, 1983; Deb et al., 2002). The other problem is that they transform the EMOA into a single objective problem where the benefit of the EMOA in producing well-distributed solutions with interesting trade-offs is diminished (Branke et al., 2010). The machine learning techniques used in the literature for estimated ranking of a finite number of solutions in the population are mostly in the form of an ensemble of many estimators, each trained to predict an individual rank, and such an approach is known to be highly sensitive to small biases (Cheng et al., 2009). Another issue with many iEMOAs in the literature is the requirement to normalize and scale objective functions, a task that is not easy to implement in real-world problems. To overcome these problems, we propose a novel iEMOA in Chapter 5 with a preference learning technique that uses a Decision Tree (DT) classifier to predict the winner solutions in pairwise comparisons; i.e the solution is classified as preferred or otherwise. Furthermore, DTs naturally employ feature selection when learning to classify and simplify the learned model (Xia, Zhang, Li, & Yang, 2008; Yu, Wan, & Lee, 2011). Furthermore, DTs are explainable, easily comprehensible, and can be understood by the DM (Cheng et al., 2009). The latter is an important characteristic as it helps the DM to understand the process and trust in final results (Allen, Moussa, & Liu, 2019). Thus, there has been an emphasis on the explainability of the learning methods in interactive methods and particularly the development of explainable learning methods such as DTs for preference learning is encouraged (Misitano, Afsar, Larraga, & Miettinen, 2022).

Chapter 3

On Benchmarking Interactive Evolutionary Multi-Objective Optimization Algorithms

Shavarani, S. M., López-Ibáñez, M., & Knowles, J.

A summary of this chapter is published in Proceedings of the Genetic and Evolutionary Computation Conference

Submitted to IEEE Transactions on Evolutionary Computation (under the second round of peer review)

We carry out a detailed performance assessment of two modern interactive evolutionary multi-objective algorithms (EMOAs) using a machine decision maker that enables us to repeat experiments and isolate and study some specific behaviours modeled after human decision makers (DMs). Using the same set of benchmark test problems as in the original papers on these interactive EMOAs (in up to 10 objectives), we bring to light interesting effects when we use a machine DM based on sigmoidal utility functions that have support from the psychology literature (replacing the simpler utility functions used in the original papers). Our machine DM enables us to go further and simulate human biases and inconsistencies as well. Our results from this study, the most comprehensive assessment of (two) interactive methods so far conducted, suggest that current well-known algorithms have shortcomings that need addressing. This further demonstrates the value of work on improved benchmarking of interactive EMOAs.

3.1 Introduction

Evolutionary multi-objective optimization algorithms (EMOAs) have proved to be a leading choice for solving (theoretical and applied) multi-objective problems for several decades

now. However, as with other methods, their performance may decline when the number of objectives increases (Ishibuchi et al., 2008), suggesting a need for further developments. In one approach to address the scaling issue, the preferences of the Decision-Maker (DM) can be incorporated into the EMOA as an extra criteria to preserve elitism, focus the search, and limit it only to the interesting parts of the Pareto front (PF). Such algorithms that use the preferences of the DM are termed preference-based EMOAs (Branke & Deb, 2005), and they are of perhaps the highest degree of interest to the field today. Depending on the stage of the preference elicitation, preference-based algorithms can be divided into “a priori”, “a posteriori” and interactive methods (Miettinen et al., 2008). To overcome problems with “a priori” and “a posteriori” methods, which elicit preference information respectively before and after the optimization process, interactive methods interact with the DM during the optimization process which leads to attractive features of these methods over “a priori” and “a posteriori” methods that can be briefly summarized as reduced computational costs by limiting the search to interesting parts of the PF, low cognitive effort required from the DM and psychological convergence (Belton et al., 2008).

Despite their merits and potential, improvements in the field have been hindered by difficulties in experimenting on them due to the pivotal role of the DM in determining the final results. Experiments cannot be performed with just one single DM because of human-to-human variation in aspects such as fatigue and learning. Having many DMs would also make the experiments unaffordable. Most studies in the literature that involve human DMs have students acting as DMs, despite they may not be knowledgeable about the problem or not take the experiment seriously (Afsar et al., 2021). On the other hand, studies that simulate DMs do so by utilizing arbitrary utility functions (UFs) and oversimplified and idealized assumptions about DMs’ preference model and interactions that make the experiments far from realistic. Under such circumstances, it is hard to say how well the algorithms perform in real-life problems and even it is unclear whether they are better than “a priori” or “a posteriori” methods.

This study makes use of and improves the Machine Decision Maker (MDM) framework proposed by López-Ibáñez and Knowles (2015) to simulate a human DM in experiments with interactive EMOAs (iEMOAs). Although other UFs can be used in the MDM framework (López-Ibáñez & Knowles, 2015), in this study the true preferences of the MDM are simulated by a Sigmoid UF proposed by Stewart (1996) that is inspired by psychological studies (Kahneman & Tversky, 1979) and was first used for benchmarking iEMOAs by Shavarani et al. (2021). This work supersedes previous studies (López-Ibáñez & Knowles, 2015; Shavarani et al., 2021) and implements several important biases and non-idealities in the literature to make the MDM more realistic. A transformation of the Stewart UF is proposed that can be

used for minimization problems preserving all its realistic assumptions. Furthermore, experiments are expanded to include many-objective problems. The proposed MDM thus enables conducting more realistic experiments, which we argue is a requirement for systematic improvement of iEMOAs.

The rest of the chapter is organized as follows: A brief review of the most relevant studies is provided in Section 3.2. Contributions of the chapter are summarized in Section 3.3. Methods and the machine decision maker characteristics are described in Section 3.4. Experimental design is laid out in Section 3.5. The results are discussed in Section 3.6. Section 3.7 provides further insights concerning the experiments performed here, and finally conclusions are drawn in Section 3.8.

3.2 State of the Art in iEMOAs

3.2.1 Background Concepts

In multi-objective optimization problems (MOPs) there exists a set of m conflicting objectives f_1, \dots, f_m that have to be optimized simultaneously, having a general form as follows:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X} \end{aligned}$$

where \mathbf{f} maps a decision vector \mathbf{x} in solution space $\mathcal{X} \subseteq \mathbb{R}^n$ to an objective vector \mathbf{z} in objective space $Z \subseteq \mathbb{R}^m$. MOPs are often termed many-objective problems when $m > 3$.

Definition 3.2.1 (Domination). A solution $\mathbf{x} \in \mathcal{X}$ dominates solution $\mathbf{y} \in \mathcal{X}$ if $\mathbf{f}(\mathbf{x})$ is not worse than $\mathbf{f}(\mathbf{y})$ in any objective and $\mathbf{f}(\mathbf{x})$ is strictly better than $\mathbf{f}(\mathbf{y})$ in at least one objective.

Definition 3.2.2 (Non-domination). A feasible solution $\mathbf{x} \in \mathcal{X}$ is said to be non-dominated if there is no solution $\mathbf{y} \in \mathcal{X}$ that dominates it. Non-dominated solutions are also known as Pareto optimal solutions.

Definition 3.2.3 (Pareto front). The projection of the Pareto solutions on the objective space is known as the Pareto front.

Due to the conflicting nature of objectives, any improvement in one objective is only possible at the cost of deterioration of at least one other objective. Thus, complete ordering of the solutions based on their objective values is not possible in MOPs. As a result, the aim of solving a MOP is finding a satisfying non-dominated solution from the Pareto front (PF). In this regard, solving a MOP consists of two phases (Miettinen et al., 2008): (1) Optimization

phase: finding a close approximation of the PF and (2) Decision making phase: finding the DM's most-preferred solution (MPS).

The number of solutions required for covering the PF increases exponentially with the number of objectives. This brings difficulties to both phases. In the first phase, challenges associated with many-objective problems (Korn, Pagel, & Faloutsos, 2001; Purshouse & Fleming, 2007; Li, Li, Tang, & Yao, 2015), make the optimization process computationally expensive. In the second phase, evaluating a large number of alternatives is not cognitively affordable for the DM and there is a high chance of missing the MPS.

To address this issue, Multi-Criteria Decision Making (MCDM) techniques are employed to support the DM in the process of decision making. In interactive methods, the two phases are interlaced until reaching a satisfying solution. The decision making phase happens in interactions with the DM scheduled in different stages of the optimization process. In each interaction the preference information is elicited from the DM. This information is used to estimate the preference model of the DM, which is exploited in the next optimization phase to guide the search towards the interesting part of the PF. Xin et al. (2018) identify four discrete parts of any interactive method that discriminate different iEMOAs: interaction pattern, the preference information, the preference model and the search engine.

With regard to preference models, iEMOAs are divided into two broad categories of ad-hoc and non-ad-hoc methods (Steuer, 1986). In non-ad-hoc methods, it is assumed that the decisions of the DM are driven by an underlying utility function (Steuer & Gardiner, 1991)¹. Thus, it is possible to use an appropriate UF to model decision making behaviors of the DM in the experiments (Miettinen, 1999; Afsar et al., 2021). However, adoption of an arbitrary UF that does not provide for a good simulation of decision making behaviors can highly bias the results of the experiments. Algorithms whose performance was proved to be satisfying in the experiments with arbitrary UFs may be far away from expectations and ineffective under real-life conditions (Shavarani et al., 2021).

3.2.2 Assessment of iEMOAs

In order to compare different algorithms, it is vital to have proper test problems, performance indicators and a well-defined controlled framework. There are performance indicators available for EMOAs such as hypervolume and number of non-dominated solutions (Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fonseca, 2003; Zitzler, Knowles, & Thiele, 2008; Bezerra, López-Ibáñez, & Stützle, 2018). However, in the case of iEMOAs, it is important to

¹Ad-hoc methods do not assume that an underlying utility function exists, however, in practice, it may be impossible to determine whether a DM is driven by a UF unknown to them (as assumed by the MDM framework) or not.

discriminate if any significant difference is due to the decisions of the DMs, interaction style or the underlying EMOA.

The design of experiments on iEMOAs is difficult due to the fact that results are highly affected by the decisions of the DM and that all DMs are different (Wallenius, 1975; Buchanan, 1994). To cancel out the noise arising from differences in different DMs, a sufficient number of DMs should be involved in the experiments (Afsar et al., 2021). Zujevs and Eiduks (2011) determined that an average of 65 participants were used by several behavioral studies in decision-making, which indicates the typical complexity and financial cost of such studies. Another important drawback of experiments with human DMs is that the results are subjective and difficult to replicate (Chen et al., 2017). To address this problem, a convenient approach in the literature when experimenting on non-ad-hoc methods is to simulate the desirability of solutions to the DM using a UF, thus removing the need of a human DM who interacts with the method. However, the reliability of the results will be under question when the selected UF is not a realistic simulation of a human DM. Existing studies on the measurement of iEMOAs use arbitrary UFs such as linear functions (Reeves & Gonzalez, 1989; Battiti & Passerini, 2010), non-linear functions (typically quadratic or polynomial functions) (Reeves & Gonzalez, 1989; Battiti & Passerini, 2010; Köksalan & Karahan, 2010), Gaussian function (Pedro & Takahashi, 2013; Narukawa et al., 2016a) and Tchebychef UFs (Köksalan & Karahan, 2010), and there is no evidence that they model the actual behavior of human DMs. In contrast there do exist UFs that are supported by psychological studies and are capable of simulating different decision-making behaviors, such as Sigmoid UFs. Sigmoid UFs are based on the prospect theory by Kahneman and Tversky (1979) which states that preferences of human DMs follow such *S*-shaped curves, being concave for gains and convex for losses. Following the results of prospect theory, various Sigmoid UFs have been proposed and used in the literature (Stewart, 2005; Gong, Zhang, & Chiclana, 2018). Sigmoid UFs also satisfy the accepted requirement of UFs being quasi-concave in the economics literature (Mas-Colell et al., 1995).

In this study we make use of a specific Sigmoid UF, hereafter called Stewart UF, which was proposed by Stewart (1996) in his studies on robustness of additive utility functions. Later, Shavarani et al. (2021) used the Stewart UF in experiments on iEMOAs and showed that the performance of iEMOAs diminishes when the Stewart UF is used, which in turn emphasizes the necessity of structured experiments to systematic progress in the field.

Although a Sigmoid UF may mimic some of the decision making behaviors of a human DM, it is still not realistic to assume an ideal DM who is consistent and error free in his decisions (Tversky & Kahneman, 1974). Consequently, in order to have a reliable analysis of interactive methods, their performance and robustness should be evaluated when con-

fronted with such non-idealities (Stewart, 2005). There are very few works in the literature that have stepped towards evaluation of interactive EMOAs under realistic simulation of DMs and even these studies are mostly limited to the simulation of mistakes using a random noise (Campigotto & Passerini, 2010; Köksalan & Karahan, 2010). Some studies have suggested a framework to account for a simulated DM and associated non-idealities. Zujevs and Eiduks (2011) proposed a DM simulation for experimenting on interactive multi-objective optimization methods based on the minimization of solutions to an ideal point, but no realistic biases or behaviors were simulated. López-Ibáñez and Knowles (2015) devised a *machine DM* (MDM) framework for experimenting on ranking-based iEMOAs. They adapted some of the non-idealities proposed in (Stewart, 1996, 1999, 2005), which are among the most extensive simulations of psychological biases and non-idealities that might happen in interactions and decisions of the DM. These include additive dependence, anchoring, loss aversion, unmodelled criteria and inconsistencies in the decisions. The original MDM study (López-Ibáñez & Knowles, 2015) used a linear UF with random weights to simulate the DM's underlying preferences.

Ojalehto et al. (2016) replace the human DM with an optimizer that guides the search according to ad-hoc preferences, while simulating two behaviors seen in human DMs; sudden preference changes and ending the process prematurely. Chen et al. (2017) proposed a *virtual decision maker library*, which can adapt to the required preference model to express different types of preference information such as weights, reference points and tradeoffs. The virtual DM is able to simulate several decision making personalities as well as dynamically changing preferences. The authors use additive quadratic functions in their experiments.

As Afsar et al. (2021) indicate, most research on interactive methods does not account for required elements such as non-idealities and realistic UFs in their studies. The main goal of this study is to demonstrate simulations of human behaviors and cognitive biases to carry out more realistic evaluations of interactive EMOAs, when it is not affordable to run such extensive experiments with human DMs, in order to highlight weaknesses and undesirable behaviors of such algorithms that are not identified by the typical ad-hoc evaluations under idealized simulations of DMs and UFs.

3.3 Contributions of this chapter

We build on previous studies and expand the MDM framework proposed by (López-Ibáñez & Knowles, 2015) with Stewart's UF and non-idealities proposed in (Stewart, 1996). The non-idealities selected from (Stewart, 1996) are additively dependent objectives and inconsistencies in DM decisions. We also propose simulation of irrelevant objectives as detailed

in Section 3.4.1.

In its original form, Stewart’s UF is proposed for maximization problems and using it in minimization problems would result in violations of its underlying assumptions. Here, we develop a Stewart UF for minimization problems while making sure that all the assumptions of the Stewart UF, including those with their roots in prospect theory, are observed.

The parameters of the proposed Stewart UF can be tuned to simulate various decision-making behaviors, such as high or low compensatory preferences. For some parameter combinations of Stewart’s UF, the utility value of the whole PF solutions may be squeezed in a narrow interval, which would make any significant observation troublesome. Also, it may be desired to have the MPS away from the corner points or on a specific part of the PF. Here, we have developed a mathematical model and a solution method, which can be used to find the parameters of the proposed UF not only to have the mentioned desired characteristics but also to target a particular decision-making behavior or a degree of non-linearity.

To illustrate the differences between traditional UFs and the Stewart UF, the experiments are repeated as well with a Tchebychev function and the results are compared. As declared in (Afsar et al., 2021), most of the studies on the performance of interactive methods evaluate a single interactive method. Here, two interactive algorithms, namely BCEMOA (Battiti & Passerini, 2010) and iTDEA (Köksalan & Karahan, 2010), are selected and their performance is analysed and compared. The two algorithms differ in internal mechanisms, including the preference elicitation, modelling and incorporation, which makes their comparison of interest. The performance of the algorithms is also evaluated in many-objective problems.

3.3.1 Summary of selected algorithms

We have chosen two ranking-based interactive EMOAs that are well-established and highly cited.

Brain Computer Evolutionary Multiobjective Optimization Algorithm (BCEMOA) (Battiti & Passerini, 2010) is an iEMOA based on NSGA-II (Deb et al., 2002). In each interaction, the DM is expected to rank a small subset of non-dominated solutions according to his preferences. The set of solutions along with their ranks are added to the training set. Training set is then used to train a support-vector machine (SVM) model (Bishop, 2006b) in order to predict the relative rank of future objective vectors. The predicted value for each solution replaces its crowding distance in subsequent generations. Further interactions are expected to increase the size of the training set and the accuracy of the learned model, making it more consistent with DM’s decisions. After evolving the population, and sorting the population

based on non-domination and the value of the learned utility function, the algorithm returns the best solution.

Interactive Territory Defining Evolutionary Algorithm (iTDEA) (Köksalan & Karahan, 2010), which is built upon an earlier study of TDEA (Karahane & Köksalan, 2010), maintains two populations; a fixed-size regular population that contains both dominated and non-dominated individuals, and an archive population with a variable size that only includes non-dominated solutions. In each generation, a single offspring is generated. If it is dominated by the members of the regular population, it is discarded, otherwise it replaces a dominated individual or a random one if the new solution does not dominate any other solution. The offspring enters the archive under two conditions. First, it should be non-dominated with respect to the archive population and second it should not violate the territory of existing individuals in the archive. Territory is defined as a region occupied by each individual in the objective space and it controls the density of the Pareto front in each region. For the territory violation check, the offspring is compared to its closest individual in the archive. At each interaction of iTDEA, the DM is asked to select the most preferred solution among several filtered solutions. The solutions in the proximity of the best are given smaller territory leading to larger number of solutions in these regions (Karahane & Köksalan, 2010). At the end, the algorithm returns a small subset of non-dominated solutions from which the DM selects the most preferred one. The size of this subset is identical to those presented in interactions.

3.4 Methods

3.4.1 Machine Decision Maker (MDM)

This study makes use of the MDM framework introduced in (López-Ibáñez & Knowles, 2015). An MDM is composed of: (1) a UF that simulates the true preferences of the MDM, i.e., the preferences of the MDM without being altered by any biases and non-idealities, and (2) transformations that alter its input objective vector or its output utility values, thus biasing the MDM's true preferences to simulate cognitive biases and other non-ideal decision-making behaviors.

Utility Function

The true preference model underlying the MDM's decisions is simulated by a UF:

$$U(\mathbf{z}): \mathbb{R}^m \rightarrow [0, 1] \quad (3.1)$$

The Stewart UF: Here, we make use of a specific Sigmoid UF that was proposed in the field of MCDM by Stewart (1996). The Stewart UF is a weighted sum of marginal functions $u_i(z_i): [0, 1] \rightarrow [0, 1]$, each of them has four parameters per objective i that control the shape of the function:

τ_i : Reference level, i.e. the point where losses are separated from gains with a steep inflation rate (loss can be interpreted as those values of the objective that are not satisfying to the MDM).

λ_i : The value of the function at the reference level.

α_i : The non-linearity of the function over losses.

β_i : The non-linearity of the function over the gains.

It has been shown that the Stewart UF is significantly harder than linear UFs and produce different behaviors than quadratic functions, even without simulating any other biases (Shavarani et al., 2021).

As outlined in (Stewart, 1996), there are some assumptions underlying the Stewart UF:

- Input objective values are maximized and within the interval $[0, 1]$.
- The UF is defined to be maximized; i.e. the higher the utility the more desirable the solution.
- The slope and curvature for gains ($\tau_i < z_i \leq 1$) are smaller than the slope for losses ($0 \leq z_i \leq \tau_i$) in order to observe an assumption of prospect theory which proposes improving a satisfying solution is less appealing than improving an unsatisfying one. In the formulation this is applied by considering $\alpha_i > \beta_i > 0$ where α_i pertains to losses and β_i pertains to gains.
- The function is concave for gains and convex for losses.

In contrast to the above assumptions, objective values should be minimized in many multi-objective optimization benchmark problems. In addition, many works in interactive EMOAs assume that the UF values should be minimized (Battiti & Passerini, 2010; Köksalan & Karahan, 2010). To be consistent with the iEMOA literature, we consider here a transformation of the Stewart UF for minimization.

The Stewart UF for minimization: The Stewart UF is designed to be maximized for maximization problems. The function is monotonically increasing, which means higher values

Table 3.1. Description of different DM behaviors simulated by combinations of τ_i and λ_i for minimization problems. The table is based on its original counterpart proposed in (Stewart, 1996).

Type	τ_i	λ_i	Description
1	[0.6, 0.9]	[0.1, 0.4]	Mainly compensatory preferences.
2	[0.6, 0.9]	[0.6, 0.9]	Mainly compensatory preferences, but with sharp preference threshold.
3	[0.1, 0.4]	[0.1, 0.4]	Limited range of compensation, all higher values nearly equally undesirable.
4	[0.1, 0.4]	[0.6, 0.9]	Limited range of compensation, plus sharp preference threshold.

of the objective functions will yield higher values of the UF. Each marginal $u_i(z_i)$ is a piece-wise linear function with different behaviors below and above τ_i . In order to give higher utility to solutions with smaller objective values z_i , we need to map the values to the $[0, 1]$ interval such that the minimum z_i will correspond to 1 and the maximum to 0. In order to maintain the assumptions about the concavity for gains and convexity for losses, as well as the meaning of α_i and β_i , we also need to transform the reference levels (τ_i) as well as the two extremes of the piece-wise intervals (0 will become 1 and 1 will become 0). The effect of these transformations will be to swap the two sides of the piece-wise function and reflect the shape of the UF along the x -axis. The resulting UF becomes:

$$U'(\mathbf{z}) = \sum_{i=1}^m w_i u_i(z_i) \quad (3.2)$$

$$u_i(z_i) = \begin{cases} \lambda_i + \frac{(1 - \lambda_i)(1 - e^{-\beta_i(\tau_i - z_i)})}{1 - e^{-\beta_i \tau_i}} & \text{if } 0 \leq z_i \leq \tau_i \\ \frac{\lambda_i \cdot (e^{\alpha_i(1 - z_i)} - 1)}{e^{\alpha_i(1 - \tau_i)} - 1} & \text{if } \tau_i < z_i \leq 1 \end{cases}$$

where the objective values z_i must be minimized and the parameters of the UF (τ_i , λ_i , α_i and β_i) have exactly the same meaning as described above, i.e., no transformation is needed. Following the iEMOA literature, we also consider here that lower utility values are preferred by transforming the utility values as $U(\mathbf{z}) = 1 - U'(\mathbf{z})$. In the rest of this chapter, we will refer to this $U(\mathbf{z})$ simply as the “Stewart UF” for conciseness.

Different decision making behaviors can be simulated by different parameter combinations of the Stewart UF as laid out in Table 3.1, which only differs from the original (Stewart, 1996) in that low values of τ_i are now associated to gains and high values to losses. Figures visualizing the effect of different parameters on the shape of the marginal UFs are provided in Appendix A, together with heatmap plots showing the PF on top of the Stewart UF for problems with 2 objectives.

A difficulty of using the Stewart UF is how to set its parameters in a way that is useful for benchmarking iEMOAs. In Section 3.4.2, we propose an algorithm for setting its parameters

given a benchmark problem.

Tchebychef UF: Aside from the realistic Stewart UF, here the Tchebychef UF is also used in the experiments to have a baseline, providing for an “easier” UF, that allows us to assess whether any observed effects are due to the Stewart UF. The Tchebychef UF is formulated as:

$$U(\mathbf{z} = \mathbf{f}(\mathbf{x})) = \max_{i=1\dots m} w_i |z_i - z_i^{\text{ideal}}| \quad (3.3)$$

where w_i is the weight of each objective function and $\mathbf{z}^{\text{ideal}}$ is the ideal or Utopian point.

Inconsistencies in the DM’s decisions

DMs are prone to mistakes and assuming an ideal DM who would provide error-free preference information is unrealistic. In reality, errors may arise due to DM’s confusion, fatigue, and lack of comprehensive information. This case may happen specifically when the presented solutions have approximately the same desirability to the DM. The DM may get confused and make the wrong decision. For example, when comparing solutions \mathbf{x} , \mathbf{y} and \mathbf{z} , the DM may make non-transitive decisions; i.e. preferring solution \mathbf{x} over solution \mathbf{y} and solution \mathbf{y} over solution \mathbf{z} , yet expressing a preference for \mathbf{z} over \mathbf{x} . To simulate these types of inconsistencies, a normally distributed random noise is generated with mean zero and variance σ and added to the utility (Stewart, 1996; Battiti & Passerini, 2010; López-Ibáñez & Knowles, 2015; Shavarani et al., 2021).

Irrelevant objectives

A specific DM interacting with the iEMOA may not be present in the problem modelling phase or her desires may not be reflected under specific circumstances. Thus, there might be non-concurrence between the modelled objectives and those that are really important to the DM. Potential objectives that are not being optimized but are relevant to the DM are said to be “hidden” to the mathematical optimization model (Battiti & Campigotto, 2010). Objectives that are considered by the optimizer, yet are irrelevant to the DM are called “irrelevant”. Changing the values of such objectives would not have any effect on the utility of those solutions to the DM. To illustrate these different cases, let us consider an iEMOA that optimizes a custom-manufactured car design according to the DM’s preferences, with three possible optimization objectives: cost, aesthetic appeal and mileage. If the iEMOA does optimize and show to the DM all three objectives, yet the DM does not care about aesthetic appeal when interacting, then the latter is an irrelevant objective. Irrelevant objectives are simulated by choosing $q \leq m$ objectives as irrelevant from the total m being optimized. In each

experiment, the objectives deemed relevant are selected randomly based on the weights w_i associated to each objective. When calculating the utility of a solution, the z_i values of the q irrelevant objectives are set to their reference level τ_i , which means there is no perceived gain or loss when the irrelevant objective value changes, hence they are irrelevant to the MDM. For Tchebychef UF, irrelevant objectives are set to 0. When $q = 0$, there are no irrelevant objectives present.

Additively-dependent objectives

Additive UFs typically assume that the marginal UFs satisfy *additive independence*, which means that the change in value of $U(\mathbf{z})$ for a constant change in value of $u_i(z_i)$, assuming all other z_j , $j \in \{1, \dots, m\} \wedge j \neq i$ are unchanged, does not depend on the fixed values of $u_j(z_j)$ (Stewart, 1996). In other words, given two objective vectors $\mathbf{z}, \mathbf{z}' \in Z$ that only differ in one objective value, i.e., $z_j = z'_j \forall j \in \{1, \dots, m\} \wedge j \neq i$, and $\exists i, z_i \neq z'_i$, then $U(\mathbf{z}) - U(\mathbf{z}') = u_i(z_i) - u_i(z'_i)$.

Stewart (1996) points out that checking additively independence is quite difficult in practice. If the objectives used as inputs to the marginal UFs are actually functions of two or more “true” objectives, then the resulting UF may not be strictly additively independent (Keeney, 1981) because a value change in one true objective may imply changes in the value of several marginal UFs.

Stewart (1996) proposes a “mixing” of objectives that ensures that objectives become additively dependent even if the true objectives are not. Let $C = \{c_1, c_2, \dots, c_{m-q}\}$ be the index set of relevant objectives in random order, then each pair of the relevant objectives $(z_{c_k}, z_{c_{k+1}})$, $k = 1, 3, 5, \dots, k < |C|$, are modified as follows:

$$\tilde{z}_{c_k} = (1 - \gamma)z_{c_k} + \gamma z_{c_{k+1}} \quad \tilde{z}_{c_{k+1}} = (1 - \gamma)z_{c_{k+1}} + \gamma z_{c_k} \quad (3.4)$$

where $\gamma \in [0, 1]$ is a parameter of the MDM that controls the intensity of the dependency. The mixing of objectives happens only for relevant objectives. Thus, irrelevant objectives, if any, are ignored. It should be noted that the additive dependence happens in the “mind” of the MDM, whereas the true set of objectives without any mixing are optimized by the algorithm, i.e. mixing happens only in the interactions and has no effect on the optimization process. More formally, the optimization algorithm optimizes the true set of objective values \mathbf{z} , whereas the utility value is calculated based on the modified objective vector $\tilde{\mathbf{z}}$ to reflect the dependence among objectives inside the MDM’s “mind”, which is not known to the optimization algorithm.

The degree of each non-ideality is controlled by their corresponding parameter in the MDM. By setting the different values of the parameters, that is σ , γ and q , various situations and non-idealities can be modelled with different intensities and the performance of the algorithms can be evaluated when different non-idealities exist.

The iEMOAs considered here interact with the above MDM by asking it to rank (or choose the best among) a set of objective vectors. For each pair compared, the MDM calculates the value of $\tilde{U}(\tilde{\mathbf{z}})$, which is obtained by applying the above non-idealities to $U(\tilde{\mathbf{z}})$ (Eqs. 3.2 and 3.3), and uses it to rank the pair. The iEMOA never has direct access to the value of $\tilde{U}(\tilde{\mathbf{z}})$ nor $U(\tilde{\mathbf{z}})$.

3.4.2 Parameter tuning of the UF

One downside of the Stewart UF compared to other less realistic UFs is the large number of parameters that need to be set before using it to benchmark an iEMOA. In particular, an experimenter needs to set α_i , β_i , τ_i , λ_i and w_i , for each $i = 1, \dots, m$, where m is the number of objectives. When setting those parameters, an experimenter may wish to enforce certain properties of the resulting UF, e.g., a minimum difference between the best and worst possible values of the UF for any solution in the PF, as well as avoid that the PF solution with the best UF value (\mathbf{z}^{MPS}) is close to the extremes or the middle of the PF, since iEMOAs that have an implicit bias towards such regions would have an advantage if the \mathbf{z}^{MPS} is located there. Finally, the experimenter may wish to more precisely control the approximate location for the \mathbf{z}^{MPS} . Manually setting the parameters of the Stewart UF to satisfy those wishes would be cumbersome. Therefore, we propose here a procedure that the experimenter can follow based on optimizing the following single-objective mathematical model.

$$\begin{aligned}
\min \quad & \sum_{i=1}^m |\check{\mathbf{z}}_i - \mathbf{z}_i^{\text{MPS}}| \\
\text{s.t.:} \quad & \mathbf{z}^{\text{MPS}} = \arg \min_{\mathbf{z} \in PF} U(\mathbf{z}; \alpha, \beta, \tau, \lambda, \mathbf{w}) \\
& \mathbf{z}^{\text{LPS}} = \arg \max_{\mathbf{z} \in PF} U(\mathbf{z}; \alpha, \beta, \tau, \lambda, \mathbf{w}) \\
& U(\mathbf{z}^{\text{LPS}}) - U(\mathbf{z}^{\text{MPS}}) \geq \Delta \\
& \alpha_i > \beta_i \quad \forall i = 1, \dots, m \\
& \sum_{i=1}^m w_i = 1 \\
& \alpha_i, \beta_i \in \mathbb{Z}^+; \tau_i, \lambda_i, w_i \in [0, 1] \quad \forall i = 1, \dots, m
\end{aligned} \tag{3.5}$$

In the model above, the experimenter specifies a desired location in objective space ($\check{\mathbf{z}}$) for the most-preferred solution. The decision variables are the parameters of the Stewart UF (α_i ,

$\beta_i, \tau_i, \lambda_i, w_i$). Given these parameter values, the model calculates \mathbf{z}^{MPS} , i.e., the solution in the PF with the best utility value (the most-preferred solution) and \mathbf{z}^{LPS} , i.e., the solution in the PF with the worst utility value (the least-preferred). The objective function minimizes the distance between $\check{\mathbf{z}}$ and \mathbf{z}^{MPS} , so that \mathbf{z}^{MPS} will be as close as possible to the solution specified by the experimenter, while respecting the constraints of the Stewart UF. Here we have chosen the Manhattan distance to reduce the computational effort of solving the model, but other distance metrics could be considered by the experimenter. We added an additional constraint that ensures the difference between $U(\mathbf{z}^{\text{LPS}})$ and $U(\mathbf{z}^{\text{MPS}})$ is larger than a threshold Δ to avoid uninformative UFs where all solutions in the PF have roughly the same utility value.

The step-by-step process of solving the model in (3.5) can be described as follows:

1. Generation of PF , which should be a reasonably good approximation of the Pareto front in order to have a chance that the resulting UF adequately captures the range of possibly UF values. In the case of benchmarking problems, this approximation can be found either analytically or by running an EMOA for sufficiently large time.
2. Selecting $\check{\mathbf{z}}$ from the PF. In this study, we chose the $\check{\mathbf{z}}$ points by selecting points far way from the center of the Pareto frontier and satisfying $\check{\mathbf{z}} > \frac{1}{4m}$, such that we also avoid the extremes of the PF.
3. Choosing a MDM type from Table 3.1, which will further constraint the valid ranges of the Stewart UF parameters.
4. Solving the model in (3.5) to identify valid parameters of the Stewart UF.

There are several ways to solve the above mathematical model. For simplicity and ease of implementation, we use a single-objective evolutionary algorithm (EA) that starts from a random population of 200 solutions, each solution in the population is a chromosome that holds the parameter values of the Stewart UF. Since there are 5 parameters for each marginal Stewart UF, when finding the parameters of an UF for a problem with m objectives, the length of the chromosome is $5m$ (Fig. 3.1). In our experiments, we have used $\Delta = 0.1$ and, if this constraint is violated, we penalize the objective function in (3.5) with a value of $1/(U(\mathbf{z}^{\text{LPS}}) - U(\mathbf{z}^{\text{MPS}}))$. We stop the EA after 200 generations, which we found empirically to be enough due to the smoothness of the Stewart UF, and the best solution found by the EA are the parameter values used for the Stewart UF.

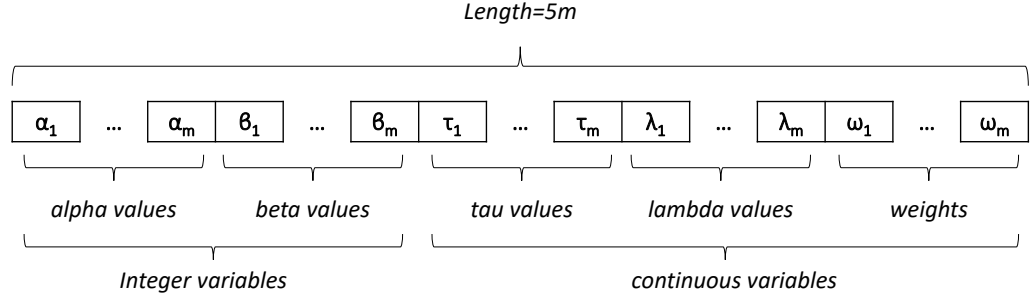


Figure 3.1. Solution representation (chromosome) used by the EA to find the Stewart UF parameters that optimize the single objective problem in Eq. (3.5).

3.5 Experimental design

3.5.1 Benchmark problems

Following the original works (Battiti & Passerini, 2010; Köksalan & Karahan, 2010), we use benchmark problems DTLZ1, DTLZ2 and DTLZ7 (Deb et al., 2005) with 2, 4 and 10 objectives. The problems with two objectives make it possible to investigate the PF, the position of the \mathbf{z}^{MPS} , \mathbf{z}^{LPS} and the contours of the UF while the high-dimensional cases would help us to scrutinize the performance of the algorithms in many-objective problems. The number of decision variables (n) is set to $m + 4$ for DTLZ1, $m + 9$ for DTLZ2 and $m + 19$ for DTLZ7 as suggested in the original study (Deb et al., 2005), m being the number of objectives.

Each problem scrutinizes different aspects of algorithm performance:

- DTLZ1: Convergence to the Pareto Front. The objective space contains 11^{k-1} Pareto optimal fronts where $k = n - m + 1$, and each of them can attract the evolutionary algorithm.
- DTLZ2: Convergence to the true PF and the capabilities of the algorithm in adapting to many-objective problems.
- DTLZ7: Diversity of the solutions. The problem has 2^{m-1} disconnected Pareto optimal regions in the objective space.

To make the problems more challenging, for DTLZ1 we follow (Battiti & Passerini, 2010) and bound x_i within $[0.25, 0.75]$. For DTLZ2 we map x_i to $x_i/2 + 0.25$, $i = 1, \dots, n$, as suggested in (Brockhoff & Zitzler, 2007b).

3.5.2 Selected UFs and parameters

Each problem is used along Stewart and Tchebychef UFs to further scrutinize the effects of the selected UF on the results. The parameters of the Stewart UF for all MDM types described in Table 3.1 are calculated using the proposed method described in Section 3.4.2. The method requires a reasonably good approximation of the PF of each problem, which we obtained by doing 20 runs of NSGA-II with a population of size 1000 and 50 000 generations and accumulating the results into a single non-dominated approximation front for each problem. We found that this approach obtained a PF approximation with a better distribution than sampling uniformly from the analytical form of the known PF (Deb et al., 2005), specially for many-objective problems, which is an observation already reported elsewhere (Bezerra et al., 2018) for the DTLZ benchmark set. The file containing the complete design of experiments along with the parameters of the UFs can be found in the supplementary material.

In order to use the Stewart UF, it is important to scale the objective values to the interval $[0, 1]$ as suggested in the original study (Stewart, 1996). It is also desired to have PF points covering a greater part of this interval rather than squeezing them in a narrow range. To do so, we treat objectives below the estimated nadir point differently than those above it. For objective values that are below the estimated nadir point the scaling is done with respect to the following formulation:

$$0.95 \cdot \frac{z_i - z_i^{\text{ideal}}}{z_i^{\text{nadir}} - z_i^{\text{ideal}}} \quad (3.6)$$

For values above the estimated nadir point, a simple linear scaling is done as follows:

$$0.95 + 0.05 \cdot \frac{z_i}{z_i^{\text{worst}}} \quad (3.7)$$

where z_i^{worst} is the worst possible value of the i^{th} objective. Doing so, the utility values of solutions beyond the nadir point are squeezed in a narrow interval of $(0.95, 1]$ while the objectives within the bounds of ideal and nadir cover the interval $[0, 0.95]$. The nadir and ideal points were estimated for each problem from the data collected when running NSGA-II to obtain a good approximation of the PF of each problem, as described above. The worst objective values can be found analytically for the benchmark problems considered in our study (Deb et al., 2005). The scaling is not needed or applied for the Tchebychef utility function because it does not require objective values in the range $[0, 1]$.

3.5.3 Parameter settings of the Machine Decision-Maker (MDM)

Measuring the effect of cognitive biases of the DM is an important aspect that should be considered in experimenting on iEMOAs (Afsar et al., 2021). In this regard, the two selected

algorithms are compared and analyzed under several conditions:

1. Under ideal conditions where no non-idealities are applied.
2. With simulation of inconsistencies in MDM's decisions ($\sigma > 0 \wedge \gamma, q = 0$).
3. With simulation of additively dependent objectives ($\gamma > 0 \wedge \sigma, q = 0$).
4. With simulation of irrelevant objectives ($q > 0, \sigma, \gamma = 0$).

We have selected different values for $\sigma \in \{0.005, 0.01, 0.1, 0.2\}$ and $\gamma \in \{0.01, 0.05, 0.1, 0.5\}$ to impose different levels of noise and additive dependence, respectively. For q , we have simulated one irrelevant objective by setting $q = 1$ for many-objective problems.

Specifications of all tests along with most- and least-preferred solutions, \mathbf{z}^{MPS} and \mathbf{z}^{LPS} respectively, are reported in Table 3.2 along with their (true) utility values. The utility value of these solutions are used as a criteria to evaluate the performance of the algorithms. It is important to be reminded that we identify \mathbf{z}^{MPS} with regard to the approximated PF, generated as described in Section 3.4.2. However, the iEMOAs may succeed in finding solutions with even better utility that did not appear in our approximated PF. This is expected as iEMOAs focus on only specific parts of the PF and have better chance of getting closer to the PF in those focused regions.

3.5.4 BCEMOA and iTDEA parameter settings

All parameters of BCEMOA and iTDEA are set as suggested in the original papers (Battiti & Passerini, 2010; Köksalan & Karahan, 2010). For problems with two objectives the initial and final territory for iTDEA are set to 0.1 and 0.00001, respectively, which was one of the alternatives suggested by its original authors. For problems with 4 and 10 objectives these values change to 0.5 and 0.25. Any smaller values for these parameters would make the size of archive population large and the computational costs unaffordable. For the number of solutions presented to the DM in each interaction, the authors of iTDEA suggest $2m$, while the authors of BCEMOA performed tests with 3 to 50 solutions. Considering the cognitive burden on the DM, we set this number to 5. The population size in all experiments is set to 400.

To give a similar computational budget to both iEMOAs, the number of solution evaluations is set to 80 000. Since iTDEA evaluates only one solution per generation while BCEMOA creates a completely new population, 80 000 evaluations correspond to 80 000 generations in iTDEA and 200 generations in BCEMOA. The different iEMOAs distribute in

Table 3.2. List of UFs used in the tests. **P** indicates the benchmark problem; n : dimension of the problem; m : number of objectives; **Type** corresponds to the MDM type in Table 3.1; $U(\mathbf{z}^{\text{MPS}})$: utility of \mathbf{z}^{MPS} (most-preferred solution); $U(\mathbf{z}^{\text{LPS}})$: utility of \mathbf{z}^{LPS} (least-preferred solution); **UF**: UF type, where “ste” stands for the Stewart UF and “tch” for the Tchebychef UF.

P	n	m	Type	$U(\mathbf{z}^{\text{MPS}})$	$U(\mathbf{z}^{\text{LPS}})$	UF
1	6	2	2	0.0101	0.9038	ste
1	6	2	1	0.1023	0.8976	ste
1	6	2	3	0.3128	0.9970	ste
1	6	2	4	0.1426	0.9378	ste
1	6	2	-	0.1159	0.3476	tch
2	11	2	1	0.3660	0.6364	ste
2	11	2	2	0.1062	0.8933	ste
2	11	2	3	0.4627	0.9996	ste
2	11	2	4	0.1261	0.9977	ste
2	11	2	-	0.3170	0.7653	tch
7	21	2	1	0.1259	0.9692	ste
7	21	2	2	0.1097	0.8894	ste
7	21	2	3	0.4061	1.0000	ste
7	21	2	4	0.3525	0.9999	ste
7	21	2	-	0.0402	0.8506	tch
1	8	4	1	0.3560	0.8928	ste
1	8	4	2	0.0047	0.3643	ste
1	8	4	3	0.3019	0.8759	ste
1	8	4	4	0.0047	0.3781	ste
1	8	4	-	0.0405	0.1589	tch
2	13	4	1	0.3267	0.9936	ste
2	13	4	2	0.0253	0.5428	ste
2	13	4	3	0.4150	0.9989	ste
2	13	4	4	0.0214	0.5193	ste
2	13	4	-	0.0567	0.3222	tch
7	23	4	1	0.1481	0.9993	ste
7	23	4	2	0.0111	0.6991	ste
7	23	4	3	0.3269	0.9992	ste
7	23	4	4	0.0228	0.7759	ste
7	23	4	-	0.0699	0.4371	tch
1	14	10	1	0.0002	0.2164	ste
1	14	10	2	0.0001	0.1609	ste
1	14	10	3	0.0215	0.8690	ste
1	14	10	4	0.0108	0.7716	ste
1	14	10	-	0.0397	0.2404	tch
2	19	10	1	0.0003	0.2953	ste
2	19	10	2	0.0001	0.2448	ste
2	19	10	3	0.1117	0.9695	ste
2	19	10	4	0.0217	0.8263	ste
2	19	10	-	0.0675	0.2446	tch
7	29	10	1	0.0016	0.8045	ste
7	29	10	2	0.0007	0.7638	ste
7	29	10	3	0.1238	0.9927	ste
7	29	10	4	0.0460	0.9842	ste
7	29	10	-	0.0580	0.2113	tch

different ways the number of generations before the first interaction, between subsequent interactions and after the last one. Thus, there is a trade-off between the number of interactions and the intensity of the search before each interaction. Here, we perform experiments with 2, 3 and 4 interactions, as higher number of interactions do not seem to substantially increase the utility of solutions and, in fact, may decrease it due to the mentioned trade-off. Fewer

than two interactions would not exercise the learning mechanisms of the iEMOAs and can hardly be called “interactive” optimization. We would like to mention that the cognitive effort required from the DM may not be the same in the two algorithms as they are different in preference elicitation style. BCEMOA asks the DM to rank a subset of solutions, while iTDEA asks to select the best among a subset of solutions. Furthermore, BCEMOA returns a single solution that has the best utility value based on its estimated UF while iTDEA returns a subset of solutions from which the DM should select the final one. When comparing the two algorithms, one needs to account for the number of solution evaluations, CPU time, cognitive effort, number of evaluated solutions, quality of the final solution, and so on. However, it is not possible to make similar settings for both algorithms in every aspect. Take for instance, the number of solution evaluations. It is impossible to have the same number of solution evaluations while fixing the run time for both algorithms. Thus, the researcher should make decisions on the most important factors. When deciding on cognitive effort, ranking a subset of solutions may be more tedious than just selecting the best. However, we have decided on having the same number of solutions evaluated by the DM and the same number of interactions for both algorithms rather than opting for having the same cognitive effort as measuring the cognitive effort and proposing a fair number of interactions deserves its own research. The decision was mainly driven by the fact that the required feedback from the DM is insignificant in both algorithm as the number of solutions and number of interactions is not large and thus we compromised for having a bit of a difference in cognitive effort rather than having a different number of evaluated solutions or interactions which we think have a more significant effect on the performance of the algorithms.

Each experiment is repeated 20 times with different random seeds.

3.5.5 Implementations

The algorithms and the Machine DM are implemented in Python 3.7.6, using the NSGA-II and benchmark implementations provided by the Pygmo library 2.16.0 (Biscani, Izzo, & Yam, 2010a) and the SVM ranking capabilities in Preference Learning Toolbox (Farrugia, Martínez, & Yannakakis, 2015) powered by scikit-learn 0.23.1 (<http://scikit-learn.org/>). For our implementation of iTDEA in Python, we studied the C++ implementation provided by its original authors at <https://bitbucket.org/ibrahimkarahan/iTDEA>.

3.6 Results

Interactive methods are supposed to come up with a single solution (or at least small subset of non-dominated solutions) that are of high interest to the DM. Thus, it makes sense to compare the performance of the algorithms based on the utility value of the final solution returned by each of them. The results of experiments for tests in Table 3.2 are discussed in this section for two Stewart and Tchebychef utility functions. The box plots and the associated strip plots presented in this section report the recorded utility values of the final solution returned by each algorithm in each test. Tables in this section provide the utility values averaged over 20 runs for each test along with the corresponding standard deviation. The performance of the two algorithms on each test is compared in the tables and the one with superior performance is indicated in bold font where the p -value of the Wilcoxon test is less than 0.05. To save space, we focus here on the most relevant results. All other figures are available as supplementary material.

3.6.1 Experiments with no biases

In this section the performance of the algorithms are compared when no biases are applied. It should be noted that we have 4 different behaviors for UF. Considering 3 test problems and 3 different objective space dimensions, the number of tests for this UF sums up to 36. Tchebychef, however, cannot simulate different DM behaviors and as a result the number of tests for this UF is reduced to 9.

Tchebychef UF

The results of experiments for Tchebychef UF are illustrated in Figure 3.2. BCEMOA manages to find the MPS in problems with 2 objectives. For DTLZ1, aside from the cases when it converges to the MPS, it seems the BCEMOA gets trapped in a point other than the MPS. For instance, in DTLZ1 with $m = 2$, there is another point with a utility value of around 0.23 that attracts the algorithm. This can be expected as DTLZ1 has many local fronts, each of which attracts the algorithm (Deb et al., 2005). With the increase in the dimensions of the objective space, the performance of BCEMOA also deteriorates and the returned points get distant from the MPS. Generally, no significant improvement is observed in the results with the increase in the number of interactions. The only exception is DTLZ7 with 4 and 10 objectives where the values and the variability are slightly decreased for BCEMOA. These results do not contradict the results of the original study (Battiti & Passerini, 2010), which reports in detail the performance of BCEMOA for increasing number of interactions only with linear

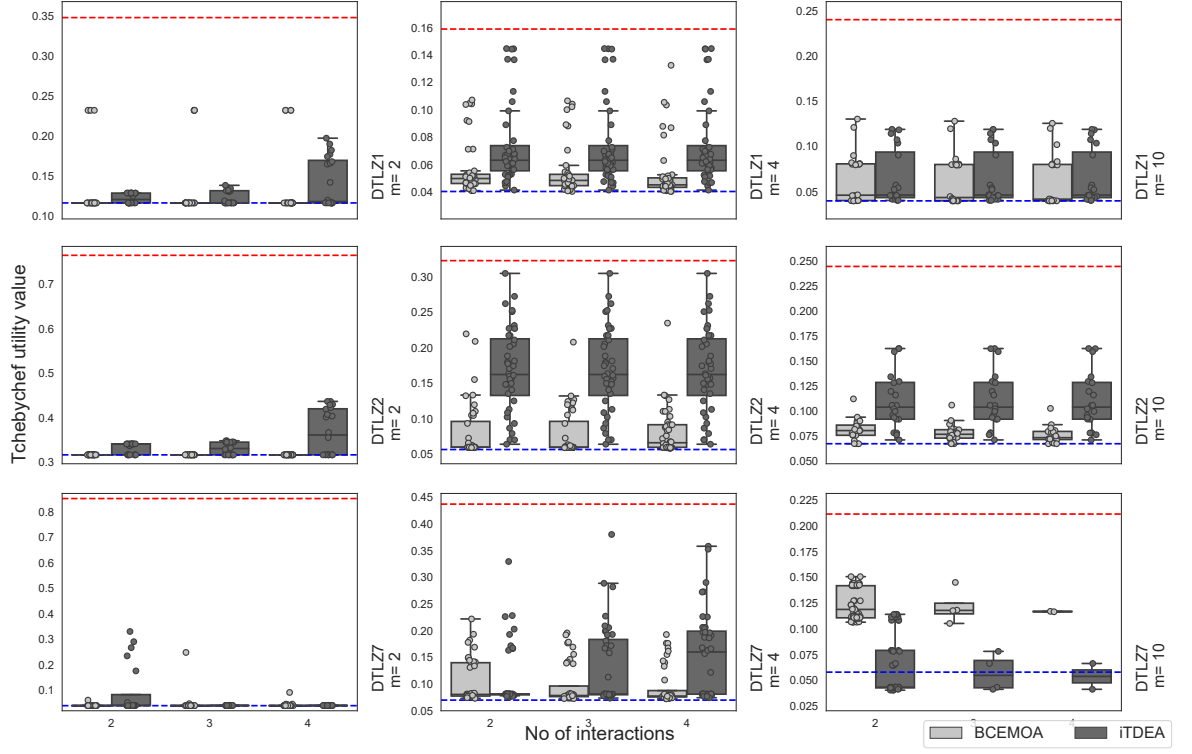


Figure 3.2. Results of the iEMOAs under ideal conditions with Tchebychev UF. Each row indicates a problem with a different objective function. Each column belongs to a different number of objective functions (m). The blue dashed line represents the utility value of the MPS and the red one illustrates the worst utility value among PF solutions. The number of interactions varies from 2 to 4. The box plots report the recorded utility values of the final solution returned by each algorithm averaged over 20 runs. Lower values are preferred.

UF, whereas for quadratic UFs, it briefly mentions that more interactions may not necessarily improve the results. The original study of iTDEA (Köksalan & Karahan, 2010) also indicates that, for complicated UFs, more interactions may lead to deteriorated results. Thus, it is not surprising to see the results getting worse or stay unchanged with more interactions.

BCEMOA finds better solutions than iTDEA in all of these tests, except those on DTLZ7 with 10 objectives, where iTDEA succeeds in finding solutions with better utility than the \mathbf{z}^{MPS} estimated from our approximation to the PF (Section 3.4.2). Those solutions are located below the blue dashed line (Fig. 3.2), which represents the utility value of \mathbf{z}^{MPS} . In other words, a single run of iTDEA using 80 000 solution evaluations was able to find a solution with better UF value than any solution in the approximation of the PF obtained by 20 runs of NSGA-II with 50 million evaluations (see Section 3.5.2). A higher number of interactions significantly improves results on DTLZ7 with 2 and 10 objectives. In other cases, whether there is a benefit of increasing the number of interactions depends on the iEMOA, e.g., on DTLZ7 with $m = 4$, BCEMOA benefits while iTDEA does not. In some cases, such as problems DTLZ1 and DTLZ2 with $m = 2$ and DTLZ7 with $m = 4$, a higher number of interactions results in iTDEA returning worse utility values, which may be explained by the fact that the total number of evaluations is fixed, thus a higher number of interactions means fewer generations before and after interactions, which may prevent the iTDEA from converging close enough

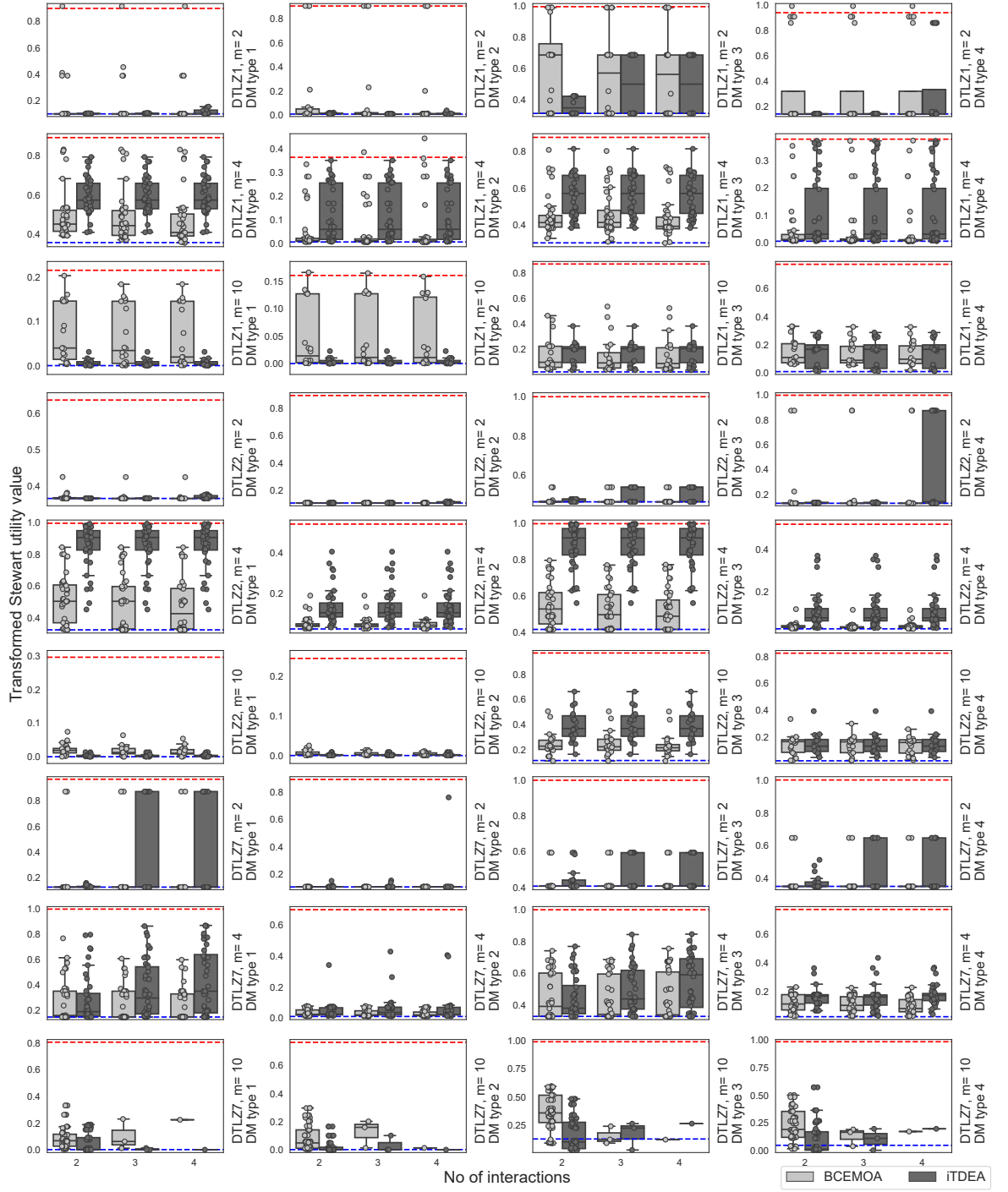


Figure 3.3. Results of the iEMOAs under ideal conditions with Stewart UF. Each column illustrates results related to a specific MDM type and each row belongs to a problem. The blue dashed line represents the utility value of the MPS and the red one illustrates the worst utility value among PF solutions. The number of interactions varies from 2 to 4. The box plots report the recorded utility values of the final solution returned by each algorithm averaged over 20 runs. Lower values are preferred.

to the PF before the MDM interacts.

The Stewart UF

The results pertaining to this section are summarized in Figure 3.3.

The performance of BCEMOA is much worse in experiments with Stewart's UF, specifi-

cally when MDM types 3 and 4 are simulated, e.g., see results on DTLZ1 with $m \in \{2, 10\}$ and DTLZ2 with $m = 10$ for both types, and on DTLZ7 with $m = 10$ for type 4 specifically. BCEMOA obtains better results in all tests with $m = 4$ and on a few other tests with $m = 2$, e.g., DTLZ7 and MDM types 1, 3 and 4. Generally, the results of BCEMOA rarely show significant improvements when increasing the number of interactions beyond three and, in some cases, results with four interactions are worse than with three and show signs of stagnation, e.g., as in the case of DTLZ7 with $m = 10$ and MDM types 1 and 4 (bottom row). This behavior can be explained by the fact that, with more interactions, fewer solution evaluations are performed before the first interaction, thus the MDM is presented solutions far from the Pareto front and may misdirect the iEMOA towards the wrong region of the front (DTLZ7 has a discontinuous front).

iTDEA manages to find the MPS in most tests with 2 objectives. In DTLZ7, iTDEA finds solutions below the blue line, which means solutions with better utility than any solution in the approximated PF used to generate the UFs (Section 3.4.2). Moreover, for problems with 10 objectives, iTDEA performs quite well on DTLZ1 with MDM types 1 and 2, DTLZ2 with type 1 and DTLZ7 with types 1, 2 and 4. Looking at all the results with Stewart’s UF, it appears that DM types 1 and 2 are relatively easier, as both algorithms find the MPS with at least a particular number of interactions. This is more obvious in problems DTLZ1 with 2 objectives and DTLZ2 with 10 objectives (first and sixth rows in Figure 3.3).

Generally, DM types 3 and 4 seem to be more troublesome and the results of both algorithms deteriorate. These DM types have a limited range of compensation (c.f. Table 3.1), which means any deviation from the narrow interval of satisfactory values of any objectives leads to large losses in the marginal utility value of that objective function.

The number of interactions here also has the same effect as with Tchebychef UF. In most cases, increasing the number of interactions beyond two does not improve the results for iTDEA. In some cases, e.g., all problems with $m = 2$ and MDM type 3, the results of iTDEA become noticeably worse when the number of interactions increases beyond two. In contrast, BCEMOA sometimes benefits from more than two interactions, e.g., DTLZ7, $m = 10$ and MDM type 3, however, a fourth interaction often undoes any benefits, as on DTLZ7, $m = 10$, and MDM type 1 and 4. Considering these results, we decided to focus on the results with 3 interactions in the remainder of the chapter, which is a compromise value for which both iEMOAs have both good and bad results and increasing the number of interactions further almost never improves the results.

3.6.2 Simulation of inconsistencies

This set of tests evaluates the results when there are inconsistencies in the decisions of the DM, which is simulated by adding random normal noise to the utility values. The results are depicted for four levels of noise (σ) for the Stewart and Tchebychef UFs in Table 3.3. The utility values returned by the two iEMOAs are statistically compared using Wilcoxon test and significantly lower (better) values (p -value < 0.05) are indicated in bold font. Additional figures can be found in the supplementary material. The number of interactions in all cases is three.

In general, larger values of σ , which simulate higher rates of inconsistencies in the DM's decisions, negatively affect the results of both iEMOAs, which is evidenced by an increase in mean utility value (higher is worse) and its standard deviation across runs (in parentheses). For problems with $m = 2$, any statistical differences between the BCEMOA and iTDEA disappears with higher values σ . With $m = 4$, BCEMOA is able to find better solutions than iTDEA even with values of $\sigma = 0.1$, but its advantage disappears for $\sigma = 2$. In the case of $m = 10$, iTDEA is able to find better solutions than BCEMOA even in the presence of large noise ($\sigma = 0.2$) and the difference between the results of iTDEA and BCEMOA grows larger as noise increases, which means that the noise has a stronger effect on BCEMOA than on iTDEA. A similar tendency can be observed even with the Tchebychef UF, where BCEMOA performs better with low noise whereas iTDEA gains back the advantage as noise increases. In this regard, it should be noted that both original studies (Battiti & Passerini, 2010; Köksalan & Karahan, 2010) indicated that their algorithms are robust towards noise, but the original analysis of iTDEA also mentioned that noise sometimes may lead to better results (Köksalan & Karahan, 2010). Thus our findings are consistent with the original studies.

3.6.3 Simulation of irrelevant objectives

These experiments test the performance of the algorithms when confronted with irrelevant objectives. Here, the number of irrelevant objectives is limited to $q = 1$ and the simulation is done only for problems with 4 and 10 objectives. The results are detailed in Table 3.4 showing utility values averaged over 20 runs along with standard deviation.

Generally speaking, having one irrelevant objective seems to have not much effect on the results in most tests and the effect is not consistently good or bad. For example, iTDEA performs slightly better with $q = 1$ than with $q = 0$ on DTLZ1 with $m = 10$ and MDM type 3 or 4, but performs worse on DTLZ7 with $m = 4$ and MDM type 2. BCEMOA seems to be more affected by the presence of an irrelevant objective, e.g., BCEMOA loses its advantage

over iTDEA on DTLZ7, $m = 4$, MDM type 4, and its results become noticeably worse on DTLZ7, $m = 10$ and MDM types 3 and 4. However, in all other cases, the effect of $q = 1$ is minor. These results suggest that both iEMOAs are able to ignore the irrelevant objective.

3.6.4 Simulation of additively dependent objectives

The effect of changing γ parameter, which simulates dependence among objectives, is illustrated in Table 3.5. In most cases, the imposed dependencies have almost no effect on the results unless it is exerted with its maximum intensity ($\gamma = 0.5$).

In the case of $m = 2$, both BCEMOA and iTDEA can keep the quality of their results up to $\gamma = 0.5$, when results deteriorate considerably, e.g., for DTLZ1 and DTLZ7 with MDM types 1, 2 and 4, utility values become at least two times worse from $\gamma = 0.1$ to $\gamma = 0.5$. For $m = 10$, iTDEA sees almost no effect from increasing γ , whereas BCEMOA results deteriorate only for DTLZ7 and MDM types 2, 3 and 4. For the Tchebychev UF, there is a clear pattern that shows the highest values of γ having a negative impact of utility values for $m = 2$ while having almost no effect for $m = 10$. In conclusion, both iEMOAs appear to be quite robust to the effects of additive dependency, except for $m = 2$ and large degrees of dependency, and iTDEA appears to be slightly more robust than BCEMOA, in general.

3.7 Discussion

We have found that even under ideal conditions, where no DM biases are simulated, the interactive algorithms considered here fail to perform well in at least some experimental conditions. This result emphasises the importance of studies focused on providing extensive results, providing a sufficient diversity of experimental settings in order to give some confidence in the robustness and consistency of performance, which we have uncovered is not there yet. We suggest that only by making use of MDMs is this possible, short of expensive and time-consuming experiments with large numbers of human DMs.

Further to this general observation, we have also found that different decision-making behaviors do have different effects on final results, underlining that it is important not only to vary test problems and interaction parameters but also the DMs simulated themselves.

Although direct comparison of iEMOAs should always be attempted only with extreme caution, and is not the primary goal of our research, we have removed sources of variation between the two algorithms in this study where possible. We note here that this has meant that, due to different mechanisms in each algorithm, the CPU time is different for them. For

instance, while mean CPU-time is 23 seconds for BCEMOA, it is increased to 484 for iTDEA. We believe that our approach, equalising solution evaluations and interactions, is justifiable, but we also note that the alternative would be to make comparisons on the basis of equally allocated (maximum) CPU time.

3.8 Conclusions

Based on Stewart’s UF, and non-idealities introduced in the literature, we here extended our MDM framework and used it to conduct a more realistic and more extensive benchmarking study on interactive multiobjective methods than previously available.

In comparison to benchmarking work undertaken elsewhere using other utility functions, we find the use of Stewart’s more psychologically plausible UF poses significant difficulties to the algorithms. To make the use of this UF general and applicable within the MDM framework, we proposed a model that can be used to identify parameters of the UF in order to achieve desired characteristics. Moreover, we also reformulated the Stewart UF to facilitate its use in minimization problems. Even though algorithms manage to find ‘most preferred solutions’ in some experiments, they fail to succeed in many others. The results indicate that even under ideal conditions (i.e. no decision-maker biases simulated) the performance of algorithms is far from desired or expected.

Future research is directed towards making the simulations more comprehensive with the simulation of realistic behaviors that were not discussed in this study such as learning, shift in preferences and fatigue. Our research here was mainly focused on reliable simulation of decision-making behaviors and their effects on performance on *existing benchmark test problems*. However, further research should investigate how to design more realistic test problems for evaluating interactive EMOAs. Finally, another important aspect is the cognitive effort of the DM. Our proposed MDM enables measuring cognitive effort as the number of queries to its internal UF, but we leave to future work such analysis. Our contribution here towards more realistic simulations of decision-making behaviors is not in conflict, nor replaces the need for studies with human DMs. On the contrary, we suggest that future research in interactive EMOAs should contrast the results obtained in this chapter, which are still artificial simulations of complex behaviors, with behavioral studies of human DMs, and use those behavioral studies to further inform the next generation of machine DMs. To motivate further research, we make our code publicly available at <https://shorturl.at/puvS2>.

Acknowledgment

We are grateful to Theo Stewart for helpful comments on an earlier version of this study; any remaining errors in the study are ours alone.

Table 3.3. The results of experiments with the simulation of inconsistencies in the decisions of the DM for BCEMOA and iTDEA. The mean (and standard deviation in parenthesis) over 20 independent runs is shown for each test. The values are rounded to 2 decimal points. Comparing the performance of 2 algorithms, the better (lower) values are indicated in bold font to highlight the winner where $p\text{-value} \leq 0.05$ for the Wilcoxon test. **P**: ID of the DTLZ benchmark problem. m : the dimension of the objective space. **Type**: DM Type.

σ	BCEMOA					iTDEA				
	.0	.005	.01	.1	.2	.0	.005	.01	.1	.2
P m Type	Stewart Utility Function									
1 2 1	.19(.2)	.25(.32)	.25(.32)	.26(.32)	.28(.32)	.1(.0)	.1(.0)	.1(.0)	.12(.02)	.15(.07)
1 2 2	.16(.33)	.13(.28)	.13(.28)	.14(.29)	.22(.35)	.01(.0)	.01(.0)	.01(.0)	.03(.04)	.04(.05)
1 2 3	.57(.27)	.54(.28)	.57(.27)	.62(.26)	.66(.26)	.5(.19)	.49(.19)	.49(.19)	.49(.2)	.51(.21)
1 2 4	.34(.34)	.35(.36)	.35(.36)	.36(.36)	.37(.36)	.14(.0)	.14(.0)	.15(.0)	.16(.02)	.16(.03)
2 2 1	.37(.01)	.37(.0)	.37(.02)	.42(.08)	.44(.1)	.37(.0)	.37(.0)	.37(.0)	.4(.04)	.45(.09)
2 2 2	.11(.0)	.11(.02)	.12(.03)	.16(.14)	.14(.03)	.11(.0)	.11(.0)	.11(.0)	.13(.03)	.14(.03)
2 2 3	.47(.03)	.49(.05)	.49(.05)	.55(.16)	.56(.16)	.5(.04)	.5(.04)	.5(.04)	.53(.08)	.63(.21)
2 2 4	.2(.23)	.25(.27)	.29(.31)	.27(.3)	.27(.28)	.13(.0)	.13(.0)	.13(.0)	.17(.17)	.26(.3)
7 2 1	.2(.23)	.2(.23)	.2(.23)	.21(.23)	.21(.23)	.39(.37)	.32(.33)	.32(.33)	.32(.33)	.28(.31)
7 2 2	.11(.0)	.11(.0)	.15(.17)	.19(.23)	.19(.23)	.11(.01)	.12(.01)	.15(.16)	.23(.28)	.23(.28)
7 2 3	.43(.07)	.44(.07)	.44(.07)	.48(.1)	.5(.18)	.48(.09)	.48(.09)	.48(.1)	.49(.1)	.54(.14)
7 2 4	.38(.09)	.39(.09)	.39(.09)	.44(.15)	.42(.15)	.46(.14)	.46(.14)	.46(.14)	.44(.13)	.48(.19)
1 4 1	.48(.13)	.49(.13)	.49(.13)	.52(.13)	.58(.15)	.59(.09)	.59(.09)	.59(.09)	.59(.09)	.59(.09)
1 4 2	.05(.09)	.06(.1)	.06(.09)	.09(.13)	.13(.14)	.12(.12)	.12(.11)	.12(.11)	.12(.11)	.12(.11)
1 4 3	.46(.12)	.46(.13)	.44(.11)	.5(.13)	.52(.16)	.56(.11)	.56(.11)	.56(.11)	.56(.11)	.56(.11)
1 4 4	.04(.09)	.04(.08)	.04(.08)	.06(.14)	.08(.11)	.1(.13)	.1(.13)	.1(.13)	.1(.13)	.1(.13)
2 4 1	.49(.17)	.52(.17)	.5(.17)	.58(.17)	.6(.18)	.85(.14)	.85(.14)	.85(.14)	.85(.14)	.85(.14)
2 4 2	.05(.03)	.05(.04)	.06(.06)	.09(.08)	.15(.15)	.13(.09)	.13(.09)	.13(.09)	.13(.09)	.13(.09)
2 4 3	.54(.12)	.55(.13)	.56(.12)	.61(.15)	.67(.16)	.88(.12)	.88(.12)	.88(.12)	.88(.12)	.88(.12)
2 4 4	.03(.01)	.04(.02)	.04(.03)	.1(.11)	.09(.08)	.11(.09)	.11(.09)	.11(.09)	.11(.09)	.11(.09)
7 4 1	.26(.14)	.26(.15)	.3(.18)	.32(.19)	.4(.23)	.37(.22)	.38(.21)	.38(.21)	.38(.21)	.39(.22)
7 4 2	.03(.02)	.04(.02)	.04(.02)	.04(.03)	.06(.07)	.06(.07)	.09(.13)	.09(.13)	.09(.12)	.11(.14)
7 4 3	.44(.14)	.43(.13)	.44(.14)	.5(.14)	.58(.15)	.5(.15)	.52(.16)	.52(.16)	.54(.16)	.54(.17)
7 4 4	.11(.05)	.11(.05)	.12(.07)	.13(.06)	.17(.11)	.15(.08)	.18(.1)	.18(.1)	.19(.11)	.22(.13)
1 10 1	.07(.07)	.09(.08)	.07(.07)	.1(.1)	.12(.1)	.01(.01)	.01(.01)	.01(.01)	.01(.01)	.01(.01)
1 10 2	.05(.06)	.05(.06)	.04(.06)	.08(.07)	.1(.09)	.0(.01)	.0(.01)	.0(.01)	.0(.01)	.0(.01)
1 10 3	.15(.14)	.14(.09)	.14(.09)	.2(.15)	.27(.18)	.17(.09)	.17(.09)	.17(.09)	.17(.09)	.17(.09)
1 10 4	.14(.08)	.14(.09)	.15(.09)	.18(.12)	.21(.13)	.15(.1)	.15(.1)	.15(.1)	.15(.1)	.15(.1)
2 10 1	.02(.01)	.02(.01)	.03(.03)	.05(.04)	.05(.05)	.0(.0)	.0(.0)	.0(.0)	.0(.0)	.0(.0)
2 10 2	.0(.0)	.01(.0)	.01(.01)	.02(.04)	.02(.03)	.0(.0)	.0(.0)	.0(.0)	.0(.0)	.0(.0)
2 10 3	.25(.1)	.27(.09)	.26(.07)	.32(.16)	.35(.17)	.39(.12)	.39(.12)	.39(.12)	.39(.12)	.39(.12)
2 10 4	.14(.07)	.14(.07)	.13(.06)	.15(.07)	.18(.11)	.15(.08)	.15(.08)	.15(.08)	.15(.08)	.15(.08)
7 10 1	.1(.11)	.11(.1)	.09(.09)	.18(.14)	.2(.15)	.01(.01)	.08(.12)	.08(.12)	.09(.13)	.09(.13)
7 10 2	.12(.1)	.09(.11)	.1(.11)	.16(.14)	.24(.15)	.03(.06)	.06(.09)	.06(.09)	.05(.07)	.06(.09)
7 10 3	.15(.08)	.36(.16)	.36(.17)	.42(.21)	.43(.21)	.17(.13)	.2(.17)	.2(.17)	.18(.16)	.19(.16)
7 10 4	.13(.08)	.22(.15)	.23(.15)	.28(.19)	.33(.18)	.1(.1)	.14(.16)	.14(.16)	.14(.16)	.17(.18)
P m Type	Tchebychef Utility Function									
1 2 -	.14(.05)	.14(.05)	.15(.05)	.2(.11)	.24(.1)	.12(.01)	.12(.01)	.12(.01)	.17(.05)	.19(.06)
2 2 -	.32(.0)	.32(.01)	.34(.04)	.43(.11)	.46(.15)	.33(.01)	.33(.01)	.33(.01)	.36(.09)	.47(.15)
7 2 -	.05(.05)	.04(.01)	.04(.02)	.1(.15)	.16(.19)	.04(.0)	.04(.0)	.04(.0)	.06(.06)	.14(.19)
1 4 -	.06(.02)	.06(.02)	.06(.02)	.08(.04)	.08(.03)	.07(.03)	.07(.03)	.07(.03)	.07(.03)	.07(.03)
2 4 -	.08(.03)	.09(.04)	.09(.04)	.13(.06)	.16(.07)	.17(.06)	.17(.06)	.17(.06)	.17(.06)	.17(.06)
7 4 -	.1(.04)	.12(.05)	.12(.05)	.17(.09)	.21(.1)	.13(.08)	.16(.1)	.16(.1)	.19(.11)	.19(.11)
1 10 -	.06(.03)	.07(.03)	.07(.03)	.07(.04)	.08(.05)	.06(.03)	.06(.03)	.06(.03)	.06(.03)	.06(.03)
2 10 -	.08(.01)	.08(.01)	.08(.01)	.11(.04)	.11(.04)	.11(.03)	.11(.03)	.11(.03)	.11(.03)	.11(.03)
7 10 -	.12(.02)	.12(.02)	.13(.01)	.14(.02)	.15(.03)	.06(.02)	.07(.03)	.07(.03)	.08(.04)	.08(.04)

Table 3.4. The results of experiments with the simulation of irrelevant objectives for many-objective problems. The mean (and standard deviation in parenthesis) over 20 independent runs is shown for each test. The values are rounded to 2 decimal points. Comparing the performance of 2 algorithms, the better (lower) values are indicated in bold font to highlight the winner where $p\text{-value} \leq 0.05$ for the Wilcoxon test. **P**: ID of the DTLZ benchmark problem. m : the dimension of the objective space. **Type**: type of Stewart UF (Table 3.1). $U(\mathbf{z}^{\text{MPS}})$: utility of \mathbf{z}^{MPS} . q : number of simulated irrelevant objectives.

				BCEMOA		iTDEA	
				$q = 0$	$q = 1$	$q = 0$	$q = 1$
P	m	Type	$U(\mathbf{z}^{\text{MPS}})$	Stewart Utility Function			
1	4	1	0.356	0.48(0.13)	0.47(0.11)	0.59(0.09)	0.61(0.1)
1	4	2	0.005	0.05(0.09)	0.07(0.09)	0.12(0.12)	0.12(0.12)
1	4	3	0.302	0.46(0.12)	0.45(0.11)	0.56(0.11)	0.57(0.11)
1	4	4	0.005	0.04(0.09)	0.04(0.06)	0.1(0.13)	0.11(0.12)
2	4	1	0.327	0.49(0.17)	0.47(0.15)	0.85(0.14)	0.81(0.14)
2	4	2	0.025	0.05(0.03)	0.06(0.04)	0.13(0.09)	0.13(0.08)
2	4	3	0.415	0.54(0.12)	0.5(0.11)	0.88(0.12)	0.85(0.13)
2	4	4	0.021	0.03(0.01)	0.03(0.01)	0.11(0.09)	0.1(0.09)
7	4	1	0.148	0.26(0.14)	0.21(0.11)	0.37(0.22)	0.41(0.26)
7	4	2	0.011	0.03(0.02)	0.03(0.02)	0.06(0.07)	0.11(0.14)
7	4	3	0.327	0.44(0.14)	0.44(0.14)	0.5(0.15)	0.54(0.16)
7	4	4	0.023	0.11(0.05)	0.17(0.03)	0.15(0.08)	0.17(0.09)
1	10	1	0.000	0.07(0.07)	0.06(0.07)	0.01(0.01)	0.0(0.0)
1	10	2	0.000	0.05(0.06)	0.05(0.06)	0.0(0.01)	0.0(0.0)
1	10	3	0.021	0.15(0.14)	0.15(0.14)	0.17(0.09)	0.12(0.07)
1	10	4	0.011	0.14(0.08)	0.13(0.1)	0.15(0.1)	0.09(0.08)
2	10	1	0.000	0.02(0.01)	0.02(0.03)	0.0(0.0)	0.0(0.0)
2	10	2	0.000	0.0(0.0)	0.01(0.01)	0.0(0.0)	0.0(0.0)
2	10	3	0.112	0.25(0.1)	0.26(0.11)	0.39(0.12)	0.34(0.1)
2	10	4	0.022	0.14(0.07)	0.14(0.07)	0.15(0.08)	0.14(0.06)
7	10	1	0.002	0.1(0.11)	0.12(0.09)	0.01(0.01)	0.07(0.06)
7	10	2	0.001	0.12(0.1)	0.08(0.09)	0.03(0.06)	0.05(0.06)
7	10	3	0.124	0.15(0.08)	0.35(0.17)	0.17(0.13)	0.18(0.12)
7	10	4	0.046	0.13(0.08)	0.22(0.15)	0.1(0.1)	0.13(0.13)
P	m	Type	$U(\mathbf{z}^{\text{MPS}})$	Tchebychef Utility Function			
1	4	–	0.040	0.06(0.02)	0.06(0.02)	0.07(0.03)	0.08(0.03)
2	4	–	0.057	0.08(0.03)	0.08(0.03)	0.17(0.06)	0.15(0.05)
7	4	–	0.070	0.1(0.04)	0.1(0.04)	0.13(0.08)	0.19(0.1)
1	10	–	0.040	0.06(0.03)	0.06(0.03)	0.06(0.03)	0.07(0.03)
2	10	–	0.067	0.08(0.01)	0.08(0.01)	0.11(0.03)	0.13(0.03)
7	10	–	0.058	0.12(0.02)	0.14(0.03)	0.06(0.02)	0.07(0.03)

Table 3.5. Results of experiments with the simulation of additively dependent objectives for BCEMOA and iTDEA. Mean utility (and standard deviation) over 20 independent runs. Values are rounded to 2 decimal points. Comparing the performance of the two iEMOAs, significantly better (lower) values are indicated in boldface (Wilcoxon test with $p\text{-value} \leq 0.05$). **P**: DTLZ problem; m : number of objectives; **Type**: type of Stewart UF (Table 3.1);

			BCEMOA					iTDEA				
γ			.0	.01	.05	.1	.5	.0	.01	.05	.1	.5
P	m	Type	Stewart Utility Function									
1	2	1	.19(.2)	.19(.2)	.19(.2)	.19(.2)	.49(.38)	.1(.0)	.1(.0)	.1(.0)	.1(.0)	.38(.35)
1	2	2	.16(.33)	.16(.33)	.2(.31)	.2(.3)	.42(.41)	.01(.0)	.01(.01)	.05(.03)	.05(.03)	.32(.37)
1	2	3	.57(.27)	.62(.26)	.62(.26)	.73(.16)	.71(.29)	.5(.19)	.54(.19)	.56(.18)	.67(.08)	.72(.27)
1	2	4	.34(.34)	.34(.34)	.34(.34)	.34(.34)	.6(.38)	.14(.0)	.14(.0)	.14(.0)	.14(.0)	.51(.34)
2	2	1	.37(.01)	.37(.01)	.37(.01)	.37(.0)	.49(.14)	.37(.0)	.37(.0)	.37(.0)	.37(.0)	.55(.13)
2	2	2	.11(.0)	.11(.0)	.11(.0)	.11(.0)	.46(.4)	.11(.0)	.11(.0)	.11(.0)	.11(.0)	.66(.37)
2	2	3	.47(.03)	.47(.03)	.48(.03)	.48(.03)	.5(.04)	.5(.04)	.51(.04)	.5(.04)	.5(.04)	.51(.04)
2	2	4	.2(.23)	.24(.27)	.2(.23)	.2(.23)	.46(.38)	.13(.0)	.13(.0)	.13(.0)	.31(.33)	.65(.35)
7	2	1	.2(.23)	.2(.23)	.2(.23)	.2(.23)	.5(.38)	.39(.37)	.35(.35)	.35(.35)	.35(.35)	.43(.37)
7	2	2	.11(.0)	.11(.0)	.18(.23)	.15(.17)	.5(.4)	.11(.01)	.12(.01)	.12(.01)	.23(.28)	.42(.39)
7	2	3	.43(.07)	.44(.08)	.44(.08)	.44(.08)	.54(.07)	.48(.09)	.46(.09)	.47(.09)	.47(.09)	.54(.05)
7	2	4	.38(.09)	.38(.09)	.38(.09)	.38(.09)	.52(.13)	.46(.14)	.44(.14)	.44(.14)	.44(.14)	.5(.12)
1	4	1	.48(.13)	.49(.13)	.49(.13)	.48(.12)	.53(.15)	.59(.09)	.62(.09)	.62(.09)	.62(.09)	.62(.09)
1	4	2	.05(.09)	.05(.09)	.05(.09)	.05(.1)	.18(.16)	.12(.12)	.13(.12)	.13(.12)	.13(.12)	.13(.12)
1	4	3	.46(.12)	.45(.11)	.44(.11)	.45(.11)	.49(.11)	.56(.11)	.59(.11)	.59(.11)	.59(.11)	.59(.11)
1	4	4	.04(.09)	.04(.09)	.05(.1)	.04(.09)	.11(.16)	.1(.13)	.11(.12)	.11(.12)	.11(.12)	.11(.12)
2	4	1	.49(.17)	.5(.17)	.48(.17)	.49(.17)	.62(.14)	.85(.14)	.81(.15)	.81(.15)	.81(.15)	.81(.15)
2	4	2	.05(.03)	.05(.03)	.06(.03)	.06(.03)	.06(.05)	.13(.09)	.13(.1)	.13(.1)	.13(.1)	.13(.1)
2	4	3	.54(.12)	.54(.12)	.54(.11)	.53(.1)	.61(.12)	.88(.12)	.85(.14)	.85(.14)	.85(.14)	.85(.14)
2	4	4	.03(.01)	.03(.02)	.03(.01)	.04(.02)	.1(.12)	.11(.09)	.1(.07)	.1(.07)	.1(.07)	.1(.07)
7	4	1	.26(.14)	.25(.14)	.25(.14)	.26(.14)	.25(.17)	.37(.22)	.39(.22)	.39(.22)	.39(.22)	.42(.25)
7	4	2	.03(.02)	.03(.02)	.03(.03)	.04(.03)	.06(.02)	.06(.07)	.04(.03)	.04(.03)	.05(.04)	.06(.07)
7	4	3	.44(.14)	.43(.14)	.42(.14)	.44(.15)	.4(.12)	.5(.15)	.55(.16)	.54(.16)	.54(.16)	.56(.19)
7	4	4	.11(.05)	.1(.05)	.11(.06)	.12(.05)	.17(.03)	.15(.08)	.14(.07)	.14(.07)	.14(.07)	.17(.09)
1	10	1	.07(.07)	.07(.07)	.06(.07)	.08(.09)	.09(.09)	.01(.01)	.0(.0)	.0(.0)	.0(.0)	.0(.0)
1	10	2	.05(.06)	.05(.06)	.05(.06)	.07(.07)	.06(.06)	.0(.01)	.0(.0)	.0(.0)	.0(.0)	.0(.0)
1	10	3	.15(.14)	.15(.14)	.14(.12)	.14(.12)	.18(.18)	.17(.09)	.14(.06)	.14(.06)	.14(.06)	.14(.06)
1	10	4	.14(.08)	.14(.09)	.14(.08)	.15(.09)	.13(.08)	.15(.1)	.13(.08)	.13(.08)	.13(.08)	.13(.08)
2	10	1	.02(.01)	.02(.03)	.02(.03)	.02(.03)	.04(.05)	.0(.0)	.0(.0)	.0(.0)	.0(.0)	.0(.0)
2	10	2	.0(.0)	.01(.01)	.01(.01)	.01(.01)	.01(.03)	.0(.0)	.0(.0)	.0(.0)	.0(.0)	.0(.0)
2	10	3	.25(.1)	.26(.1)	.26(.1)	.27(.12)	.25(.1)	.39(.12)	.39(.11)	.39(.11)	.39(.11)	.39(.11)
2	10	4	.14(.07)	.14(.07)	.13(.08)	.15(.07)	.16(.05)	.15(.08)	.14(.06)	.14(.06)	.14(.06)	.14(.06)
7	10	1	.1(.11)	.09(.08)	.09(.09)	.09(.08)	.14(.12)	.01(.01)	.06(.09)	.06(.09)	.06(.09)	.06(.09)
7	10	2	.12(.1)	.11(.1)	.11(.11)	.11(.1)	.12(.12)	.03(.06)	.03(.05)	.03(.05)	.03(.05)	.03(.05)
7	10	3	.15(.08)	.36(.16)	.38(.16)	.36(.15)	.36(.16)	.17(.13)	.12(.12)	.12(.12)	.12(.12)	.12(.12)
7	10	4	.13(.08)	.24(.16)	.24(.15)	.26(.17)	.4(.23)	.1(.1)	.11(.13)	.11(.13)	.11(.13)	.1(.13)
P	m	Type	Tchebychef Utility Function									
1	2	–	.14(.05)	.14(.05)	.14(.05)	.14(.05)	.25(.16)	.12(.01)	.12(.01)	.12(.01)	.12(.01)	.2(.07)
2	2	–	.32(.0)	.32(.0)	.32(.0)	.32(.0)	.52(.23)	.33(.01)	.33(.01)	.33(.01)	.33(.01)	.63(.21)
7	2	–	.05(.05)	.04(.0)	.07(.14)	.12(.23)	.81(.14)	.04(.0)	.04(.0)	.04(.0)	.27(.35)	.85(.0)
1	4	–	.06(.02)	.06(.02)	.06(.02)	.06(.02)	.07(.03)	.07(.03)	.08(.03)	.08(.03)	.08(.03)	.08(.03)
2	4	–	.08(.03)	.08(.03)	.08(.03)	.08(.04)	.14(.07)	.17(.06)	.17(.05)	.17(.05)	.17(.05)	.17(.05)
7	4	–	.1(.04)	.1(.04)	.11(.05)	.17(.06)	.38(.04)	.13(.08)	.13(.07)	.13(.07)	.15(.07)	.21(.1)
1	10	–	.06(.03)	.06(.03)	.07(.03)	.06(.03)	.06(.03)	.06(.03)	.06(.03)	.06(.03)	.06(.03)	.06(.03)
2	10	–	.08(.01)	.08(.01)	.08(.01)	.08(.01)	.09(.03)	.11(.03)	.11(.03)	.11(.03)	.11(.03)	.11(.03)
7	10	–	.12(.02)	.14(.01)	.15(.01)	.15(.01)	.15(.01)	.06(.02)	.08(.03)	.08(.03)	.08(.03)	.08(.03)

Chapter 4

Detecting Hidden and Irrelevant Objectives in Interactive Multi-Objective Optimization

Shavarani, S. M., López-Ibáñez, M., & Allmendinger, R.

IEEE Transactions on Evolutionary Computation

Evolutionary multi-objective optimization algorithms (EMOAs) typically assume that all objectives that are relevant to the decision-maker (DM) are optimized by the EMOA. In some scenarios, however, there are *irrelevant* objectives that are optimized by the EMOA but ignored by the DM, as well as *hidden* objectives that the DM considers when judging the utility of solutions but are not optimized. This discrepancy between the EMOA and the DM's preferences may impede the search for the most-preferred solution and waste resources evaluating irrelevant objectives. Research on objective reduction has focused so far on the structure of the problem and correlations between objectives and neglected the role of the DM. We formally define the concepts of irrelevant and hidden objectives here and propose methods for detecting them, based on uni-variate feature selection and recursive feature elimination, that use the preferences already elicited when a DM interacts with a ranking-based interactive EMOA (iEMOA). We incorporate the detection methods into an iEMOA capable of dynamically switching the objectives being optimized. Our experiments show that this approach can efficiently identify which objectives are relevant to the DM and reduce the number of objectives being optimized while keeping and often improving the utility, according to the DM, of the best solution found.

4.1 Introduction

In many real-world optimization problems, there are tens of numerical features of a candidate solution that could, in principle, be optimized by means of an Evolutionary Multi-Objective Algorithm (EMOA) and it is often tempting to model as many objectives as possible (Sinha et al., 2013). However, the runtime of various steps within an EMOA increases with the number of objectives (Jensen, 2003; Allmendinger et al., 2022) and, in the case of hypervolume-based methods, this increase is exponential (Beume, Fonseca, López-Ibáñez, Paquete, & Vahrenhold, 2009). Moreover, the fraction of solutions that are Pareto-optimal increases exponentially with the number of objectives (Purshouse & Fleming, 2003b; Singh et al., 2011), which complicates the *a posteriori* decision-making phase (Brockhoff & Zitzler, 2009).

Previous research on objective reduction considered removing objectives that are highly correlated to other objectives (Brockhoff & Zitzler, 2009; Sinha et al., 2013) or do not significantly alter the dominance relations among solutions (Brockhoff & Zitzler, 2006, 2007a). However, regardless of the structure of the problem, some of the objectives may not be relevant to the DM and they can be removed from the optimization model. Interactive EMOAs (iEMOAs) (Jaszkiewicz & Branke, 2008) iteratively elicit and exploit preference information of a decision maker (DM) to guide the optimizer towards preferred solutions. It is possible to exploit the elicited information to identify and remove objectives that are not relevant to the DM.

Such a scenario can happen if this particular DM was not consulted during the modeling phase or her preferences changed during the optimization due to learning (Campigotto & Passerini, 2010) and “preference drift” (Pu & Chen, 2008). Thus, there may be objectives that are optimized by the iEMOA but are *irrelevant* to the DM. In other cases, the DM’s preferences may depend on both the value of the objectives being optimized and the value of other numerical features that are measured by the system and observed by the DM but are not optimized. Such features are said to be *hidden* from the optimizer and would lead to results that are not satisfactory to the DM if not considered by the optimizer (Battiti & Campigotto, 2010; Kryston et al., 2022). Stewart (1996) discusses the concept of “unmodelled” criteria, which appear in the DM’s internal utility function but are missing from the preference model.

This discrepancy may make the elicited preferences seem non-rational, e.g. when a solution that is dominated with respect to the modelled objectives is preferred by the DM over a non-dominated one because the former is better than the latter with respect to features that are not optimized as objectives. Let us consider a simplified example inspired by the real-world problem discussed by Ramos-Pérez, Miranda, Segredo, León, and Rodríguez-León (2021) of

planning school lunches in terms of not only cost, but also a number of metrics of nutritional value and food variety. Imagine two candidate menu plans with three features $\mathbf{z}_1 = (7, 4, 2)$ and $\mathbf{z}_2 = (7, 3, 6)$, where each feature is total cost, the total amount of calcium, and the variety of vegetables, respectively. Further assume that the iEMOA was designed to optimize only the first two features as objectives, thus, \mathbf{z}_1 dominates \mathbf{z}_2 , i.e., \mathbf{z}_1 is not worse in cost and is better in amount of Calcium than \mathbf{z}_2 . However, a DM (e.g., the nutritionist of a particular school who is aware that children in this school already have a diet rich in Calcium outside school but struggle to eat their vegetables) may not look at the second objective (Calcium) and instead wishes to maximize the third feature (vegetables variety), thus prefer \mathbf{z}_2 over \mathbf{z}_1 .

One may argue that the system should allow the DM to specify which features must be optimized. But in practice a DM may only realize the relevance of a feature during the optimization process (Pu & Chen, 2008). Moreover, the DM may not be conscious of the particular features that are guiding her decisions, e.g., the nutritionist may directly compare the composition of the menus instead of looking at any summary metrics provided by the iEMOA. In some problems a DM may be able to judge solutions according to qualitative aspects (e.g., the quality of the behavior of a robot performing a task) without being aware that there exist numerical features (e.g., the number of turns or minimum distance to walls) that could act as “proxy” objectives (Trianni & López-Ibáñez, 2015) to that aspect.

In the above scenarios, it would be desirable if an iEMOA could (1) detect the discrepancy between the objectives being optimized and the features that influence the preferences provided by the DM at each interaction; and (2) dynamically select the features that are optimized after each interaction. In the case of *irrelevant* objectives, removing them from the optimization phase increases the efficiency of any EMOA (as fewer objectives are optimized) and may also help to find better solutions (Brockhoff & Zitzler, 2006).

To the best of our knowledge, for the first time in the literature we consider the preference information collected by an iEMOA when interacting with the DM as an opportunity to detect irrelevant objectives and remove them during the optimization. In addition, our proposal is also able to detect numerical features that are measured but not optimized by the iEMOA before the interaction, but are correlated with the DM’s preferences. We propose that, after each interaction, the iEMOA dynamically activates the optimization of such *hidden* objectives, thus adapting the search to the preferences of the DM. By removing irrelevant objectives and optimizing hidden ones, an iEMOA is able to adapt to a diverse range of DMs and preference changes during optimization without requiring the simultaneous optimization of every potential objective.

Our main contributions can be summarized as follows:

- The formal definition of *irrelevant* and *hidden* objectives in interactive multi-objective optimization.
- A method to detect irrelevant and hidden objectives from the ranking information provided by a DM and to dynamically update the set of objectives after each interaction. The approach draws on feature selection methods and can be applied to any ranking-based iEMOA.
- A benchmarking approach that simulates irrelevant and hidden objectives using classical multi-objective problems. This approach is demonstrated for DTLZ problems (Deb et al., 2005) and multi-objective NK-landscapes ρ MNKs (Verel, Liefooghe, Jourdan, & Dhaenens, 2013).
- The empirical validation of the proposed detection method using:
 - i. Problems of varying dimensionality, complexity, and Pareto front structure.
 - ii. Different utility functions that simulate different DMs.
 - iii. Different feature selection methods to detect relevant objectives.
- A sensitivity analysis to understand the performance impact of key parameters of the proposed approach.

Experimental results show that the proposed method can almost always replace irrelevant objectives with relevant ones quickly and significantly improve the utility of the solutions found.

The rest of the chapter is organized as follows. Several fundamental concepts on which this work is based are defined in Section 4.2. A summarized background on previous efforts towards objective reduction is given in Section 4.3. In Section 4.4, the proposed method and several variants of it are elaborated in detail. The experimental setup is laid out in Section 4.5. The results of the experiments are discussed in Section 4.6. Finally Section 4.7 provides conclusions and future research directions.

4.2 Definitions

Let us consider an optimization problem with n decision variables, where, given a solution vector $\mathbf{x} = (x_1, \dots, x_n)$ from the feasible decision space \mathcal{X} , we can compute a set $F = \{f_1, \dots, f_m\}$ of m numerical *features*, $f_i: \mathcal{X} \rightarrow \mathbb{R}$. In principle, all these features could be optimized as objectives. In the following, we assume minimization without loss of generality.

Definition 4.2.1 (Potential objectives). All m features in F are called *potential* objectives.

Let us assume as well that, either due to decisions made during the modeling phase or efficiency reasons, only a subset $\hat{F} \subseteq F$ ($\hat{m} = |\hat{F}|$) of the potential objectives must be minimized as optimization objectives, resulting in the following multi-objective optimization problem:

$$\begin{aligned} &\text{Minimize} && (f_1(\mathbf{x}), \dots, f_{\hat{m}}(\mathbf{x})) \\ &\text{subject to} && \mathbf{x} \in \mathcal{X} \end{aligned} \tag{4.1}$$

where $f_i \in \hat{F}$ are the objectives minimized by the optimization method, while $F \setminus \hat{F}$ are not.

Definition 4.2.2 (Active objective). An objective is *active* if it must be optimized by the optimization method. The set of active objectives is denoted by \hat{F} . Inactive objectives ($F \setminus \hat{F}$) are either evaluated but ignored by the optimization method or not evaluated at all.

In the EMOA literature, computational cost is often measured in terms of *solution evaluations*, where each *solution evaluation* usually means the evaluation of all its (active) objectives. Here we will consider *objective evaluations* instead because different solutions may be evaluated for different subsets of objectives, and these subsets may also vary in cardinality.

Definition 4.2.3 (Objective evaluation). The evaluation of any of the objectives f_i corresponding to a solution \mathbf{x} is counted as one objective evaluation. Thus, the cost of a solution evaluation is \hat{m} objective evaluations.

Although most EMOAs assume that the set of active objectives is decided in the modeling phase and remains constant, it is possible to change the set of active objectives during the optimization by choosing any subset of potential objectives, as we will show later. When solving the above problem in terms of Pareto optimality, an EMOA only considers active objectives.

Definition 4.2.4 (Dominance (Zitzler et al., 2008)). Given two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, we say that \mathbf{x} dominates \mathbf{y} if the former is not worse than the latter in any objective and it is strictly better in at least one, i.e., $\forall f_i \in \hat{F}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists f_j \in \hat{F}, f_j(\mathbf{x}) < f_j(\mathbf{y})$.

Definition 4.2.5 (Pareto optimal (Zitzler et al., 2008)). A feasible solution $\mathbf{x} \in \mathcal{X}$ is called Pareto optimal or non-dominated if there is no $\mathbf{y} \in \mathcal{X}$ that dominates it. The set of (mutually non-dominated) Pareto optimal solutions is the *Pareto set*.

Definition 4.2.6 (Pareto front (Zitzler et al., 2008)). The image of the Pareto set on the objective space defined by \hat{F} is known as the Pareto front (PF).

Definition 4.2.7 (Redundant objectives (Gal & Leberling, 1977)). An objective is called *redundant* if it can be removed from the set of active objectives without changing the set of Pareto optimal solutions. Saxena et al. (2009) extend this definition to include objectives that are not conflicting with a non-redundant objective.

The above definitions are independent of the preferences of a human DM interacting with an EMOA. In the case of interactive EMOAs (iEMOAs), the DM provides preference information, e.g., by ranking a subset of solutions, to guide the algorithm towards the DM's most preferred solution. Let us assume the DM can observe the value of all potential objectives when comparing solutions. For reasons explained in the introduction, there may exist a discrepancy between the active objectives being optimized by the iEMOA and the objectives considered by the DM when comparing solutions.

We can formally define this discrepancy in the case of non-ad-hoc interactive methods, which assume there exists a utility function (UF) guiding the DM's decisions but unknown to the iEMOA (Steuer & Gardiner, 1991; Fowler et al., 2010b; Köksalan & Karahan, 2010; Pedro & Takahashi, 2013; Afsar et al., 2021). Ad-hoc methods assume that no such UF exists (Steuer & Gardiner, 1991). Due to the popularity of UFs in modeling preferences, the vast majority of iEMOAs are non-ad-hoc methods, thus we focus on them in the remainder of the chapter. Without loss of generality, we assume an UF of the form $U: \mathbb{R}^m \rightarrow \mathbb{R}$, whose input is the vector-valued function $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ with components being the set F of potential objectives. Although U receives as input the value of all potential objective functions, it may not use all those values to calculate its output.

Definition 4.2.8 (Irrelevant objectives). An objective $f_i \in F$ is called *irrelevant* if its value does not affect the value of the DM's UF. That is, any two solutions $\mathbf{x}, \mathbf{y} \in \mathcal{X}$ with the same value in all potential objectives except f_i should also have the same utility value, i.e., $f_j(\mathbf{x}) = f_j(\mathbf{y}), \forall f_j \in F \setminus \{f_i\} \Rightarrow U(\mathbf{f}(\mathbf{x})) = U(\mathbf{f}(\mathbf{y}))$.

Hereafter, $F_{\text{DM}} \subseteq F$ denotes the set of objective functions relevant to the DM, thus the set of irrelevant objectives is given by $F \setminus F_{\text{DM}}$.

Definition 4.2.9 (Hidden objectives). An objective $f_i \in F$ is *hidden* if it is relevant but not (currently) active, i.e., $f_i \in F_{\text{DM}} \wedge f_i \notin \hat{F}$.

Hidden objectives may confuse the iEMOA, since the interaction with the DM may be consistent with the dominance criterion for the objectives in F_{DM} but not for the objectives in \hat{F} . If $F_{\text{DM}} \subset \hat{F}$, then no hidden objectives exist, but the iEMOA is optimizing some irrelevant objectives, which makes the problem more challenging for the iEMOA and is wasteful if the evaluation of those objectives is expensive. Similarly, if $F_{\text{DM}} = \hat{F}$, then neither hidden nor

irrelevant objectives exist, and the iEMOA is optimizing precisely the objectives that the DM cares about.

In the rest of the chapter, when considering benchmark problems and known UFs, we will assume for simplicity that irrelevant objectives are not a (trivial) function of relevant ones nor vice versa, so that the set of relevant objectives F_{DM} , and, hence, irrelevant and hidden ones, can be inferred from the definition of the UF. In practice, the DM’s UF is unknown and, in the case of black-box optimization, we may not know whether an objective is a function of other objectives, thus an objective is considered irrelevant if its value does not seem to influence the DM’s decisions.

From the above definitions, it can be concluded that while *redundant* objectives are determined based on the structure of the problem, *irrelevant* and *hidden* objectives are defined from the DM’s perspective. An irrelevant objective may be redundant or not, however, a redundant objective cannot be relevant unless the DM’s preferences are somehow inconsistent with Pareto optimality. On the other hand, a redundant objective may *appear to be* relevant if it is correlated with a relevant objective. While there are studies on the detection and elimination of redundant objectives, which we review in the next section, there is no prior research on the identification of irrelevant and hidden objectives to the best of our knowledge. Our focus here is to fill this gap and we propose a method to tackle it in Section 4.4.

4.3 Background and Literature Review

We have carried out a thorough literature review of methods for reducing the number of objectives, which we briefly summarize here. Many of these studies use the term *dimension* reduction to refer to the same concept. However, to avoid confusion with methods that reduce the number of decision variables (Allmendinger & Knowles, 2010), we use the term “objective reduction”.

Most of the studies on objective reduction focus on selecting *a priori* a subset of objectives to facilitate the optimization process while preserving Pareto optimal solutions as much as possible. Early proposals (Gal & Leberling, 1977; Agrell, 1997) make strict assumptions about the problem structure that are impossible to meet for real-life problems. Recent approaches (Brockhoff & Zitzler, 2007a; Singh et al., 2011) sacrifice to exactly identify the correct subset of objectives and capture the entire PF in order to increase applicability.

Other approaches identify similar objectives and recombine them into a single one. For example, harmonic levels (Freitas et al., 2013) and aggregation trees (de Freitas et al., 2015) are used to identify harmonious objectives (improvement of one objective does not lead to

deterioration of the others). In particular, aggregation trees are used *a posteriori* for facilitating the decision-making phase. Similarly, principal component analysis (PCA) has been used to identify correlated objectives that may be combined into a single objective *a posteriori* to facilitate decision-making (Lygoe et al., 2010) or during the optimization process to increase computational efficiency (Saxena, Duro, Tiwari, Deb, & Zhang, 2013). However, Costa and Oliveira (2010) have shown that objectives that are deemed redundant by PCA may be “informative”, i.e., contain trade-off information that would be lost if omitted.

Finally, projection methods map all objectives into two or three dimensions for visualization (Fieldsend & Everson, 2013). These methods aim to help decision-making *a posteriori* (after optimization), however, they do not help the optimization process itself and do not consider the DM’s preferences.

Interactive methods iteratively elicit the DM’s preferences to direct the search toward preferred regions of the PF (Branke et al., 2008; Xin et al., 2018). However, it is possible to use the provided information to identify objectives that are relevant to the DM, a task that has not been achieved in existing iEMOAs. Generally, preference elicitation and interaction style can take two forms (Tomczyk & Kadziński, 2020b). Direct preference elicitation requires the DM to identify some parameters of the preference model directly, which can be in the form of the reference point (aspiration level/goal) (see, e.g., PBEA (Thiele et al., 2009), WASF-GA (Ruiz et al., 2015b)), reservation levels (González-Gallardo et al., 2021), and weights (see, e.g., R-NSGAII (Deb & Sundar, 2006)), among others. On the other hand, in indirect approaches, the DM is required to provide some holistic judgments, which tend to be less demanding, mainly in the form of exemplary decisions. When indirect approaches are used, the DM is not required to have prior knowledge of solution space and the optimization algorithm (Jacquet-Lagreze & Siskos, 2001). Indirect queries can be in the form of pairwise comparisons (Battiti & Passerini, 2010; Branke et al., 2010; Branke et al., 2016; Tomczyk & Kadziński, 2019b, 2019c; Benabbou et al., 2020), selecting the best among a small subset of solutions (Fowler et al., 2010a; Köksalan & Karahan, 2010; Tomczyk & Kadziński, 2021a), accepting or rejecting a presented trade-off (Zionts & Wallenius, 1983), or ordering a subset of solutions (Sinha et al., 2018). Generally, iEMOAs that require the DM to rank a subset of solutions are known as ranking-based iEMOAs. In this research, we propose an approach that can use the solutions that are ranked by the DM in ranking-based iEMOAs to identify relevant objectives and update the set of active objectives accordingly.

4.4 Methods

As described above, existing approaches for objective reduction are mainly concerned with removing redundant or correlated objectives without interacting with a DM. In this section, we propose a method that is able to use the DM’s preferences, elicited when interacting with an iEMOA, and equip the iEMOA with the ability to identify irrelevant objectives as well as hidden ones, and switch them dynamically during the optimization process. In practice, we have found that it is relatively easy and efficient to extend iEMOAs with this capability, as switching objectives can only happen after an interaction and the number of interactions is always much smaller than the number of generations of the iEMOA.

In a nutshell, our proposal works as follows. At some point during its execution, the iEMOA interacts with the DM by showing the value of all potential objectives of a selected subset of solutions and asking the DM to rank the solutions according to her preferences. Feature selection, applied to the rankings and the objective values, is used to identify which objectives have the most significant effect on the ranking. The method uses this information to possibly activate currently inactive objectives and/or deactivate currently active ones. The iEMOA then continues its execution using not only the ranking information provided but possibly a new set of active objectives. In what follows, we describe our proposal in detail.

4.4.1 Feature Selection

We explore two feature selection methods in this study to understand the relevance of this algorithmic component: Uni-variate feature selection and recursive feature elimination (RFE). Hereafter, feature and objective are used interchangeably in this context.

Uni-variate Feature Selection

We propose the application of F-test¹ uni-variate feature selection for identifying the most relevant features. The F-test assumes that the data is normally distributed and p -values may be unreliable for large deviations from normality. If the normality assumption is not valid, then the alternative approach is to use the non-parametric mutual information (Kraskov, Stögbauer, & Grassberger, 2011, 1; Ross, 2014), which measures the dependency between two random variables. Our explanation focuses on the F-test but it is easily extended to the mutual information-based test. Preliminary experiments have shown that the normality assumption is valid for the problems considered in our study, as verified by the D’Agostino-Pearson test (D’Agostino & Pearson, 1973), and the F-test provides slightly better accuracy. These

¹The name of the test is unrelated to the set F of potential objectives.

experiments also indicate that the accuracy of the method is acceptable even with small sample size and improves significantly when the training set increases to 10 and 15. The details of these experiments are provided in Appendix B.

In uni-variate methods, each feature is considered independently and any correlation between features is ignored (Destrero, Mosci, Mol, Verri, & Odone, 2009). Let T be the set of solutions presented to the DM at an interaction, where $\mathbf{z}_j \in T$ is the vector of objective values of the j^{th} solution presented to the DM, and z_{ji} denotes the value of its i^{th} objective out of the m potential objectives ($f_i \in F$). The DM ranks the solutions according to her own preferences (smaller rank values are more preferred). The vector of rankings is given by \mathbf{r} , where r_j is the rank corresponding to $\mathbf{z}_j \in T$. There is no restriction on the rankings and two solutions may have the same rank.

The procedure for F-test uni-variate feature selection can be described as follows (for a more detailed introduction to F-test feature selection please refer to (Heiman, 2000)):

Step 1: The correlation ρ_i between each objective (feature) i and \mathbf{r} is computed as

$$\rho_i = \sum_{j=1}^{|T|} \frac{(z_{ji} - \bar{z}_{\cdot i}) \cdot (r_j - \bar{\mathbf{r}})}{S_{z_{\cdot i}} S_{\mathbf{r}}}, \quad (4.2)$$

where $\bar{z}_{\cdot i}$ and $S_{z_{\cdot i}}$ are the mean and standard deviation of the i^{th} objective value over all solutions in T , respectively, and $\bar{\mathbf{r}}$ and $S_{\mathbf{r}}$ are the same for the vector of rankings.

Step 2: The F-statistic for each objective is computed as

$$\mathcal{F}_i = \frac{\rho_i}{1 - \rho_i} \cdot (|T| - 2). \quad (4.3)$$

Here, the F-statistic is a notion of how well an objective can explain the rankings provided by the DM.

Step 3: The p -values corresponding to each F-statistic can be calculated by any statistical software.

Step 4: Features with lower p -value are selected. Number of selected features can be either fixed to a given value k ($2 \leq k < |F|$) or variable. In the latter case, objectives with p -values less than a predetermined threshold $\tau \in (0, 1]$ are selected (or at least two objectives). These two variants are explained in Section 4.4.2.

The lower the p -value, the better is the corresponding objective function in explaining the DM's rankings. The pseudo-code of uni-variate feature selection is illustrated in Algorithm 1.

Table 4.1. A numerical instance of uni-variate feature selection. $N_{\text{exa}} = 5$ solutions are ranked (\mathbf{r}) by the DM on a problem with $m = 4$ objectives. The p -value of each objective is calculated as explained in the Algorithm 1. $k = 2$ objectives with minimum p -values are activated (f_1 and f_4 in bold-face) and the EMOA continues the optimization with the new set of active objectives.

$N_{\text{exa}} = 5$	f_1	f_2	f_3	f_4	\mathbf{r}
1	0.71	0.63	0.37	0.45	4
2	0.65	0.08	0.89	0.40	3
3	0.51	0.64	0.75	0.12	1
4	0.65	0.84	0.79	0.31	2
5	0.95	0.32	0.86	0.82	5
p -values	0.03	0.45	0.82	0.01	

Algorithm 1: Uni-Variate Feature Selection

Input:

F : Set of all potential objectives (features)

T : Set of objective vectors ranked by the DM

\mathbf{r} : Vector of ranks

and either

- $k \in [2, |F|) \subset \mathbb{N}$ (for fixed number of objectives) or
- $\tau \in (0, 1] \subset \mathbb{R}$ (for variable number of objectives)

```

1 for  $i \leftarrow 1$  to  $|F|$  do
2   Step 1: Calculate  $\rho_i$  using Eq. (4.2)
3   Step 2: Calculate  $\mathcal{F}_i$  using Eq. (4.3)
4   Step 3: Calculate  $p_i$  ( $p$ -value) from  $\mathcal{F}_i$ 
5 if Fixed number of objectives then
6    $\hat{F} \leftarrow k$  objectives from  $F$  with lowest  $p$ -value
7 else
8    $\hat{F} \leftarrow \{f_i \in F \mid p_i < \tau\}$ 
9   if  $|\hat{F}| < 2$  then
10     $\hat{F} \leftarrow 2$  objectives from  $F$  with lowest  $p$ -value
11 return  $\hat{F}$  (selected objectives)

```

To illustrate the procedure after a given interaction, we outline an example in Table 4.1. We assume a fixed number of objectives ($k = 2$) is considered here. $N_{\text{exa}} = 5$ solutions are evaluated by the DM as indicated by \mathbf{r} . The p -value of each objective is obtained as explained by uni-variate feature selection. Regardless of the state of the algorithm and set of active objectives before this interaction, the two objectives f_1 and f_4 are activated after this interaction, and other objectives (f_2 and f_3) are deactivated.

Recursive Feature Elimination

RFE is different from uni-variate feature selection in that it first uses logistic regression to build a model based on all the features to predict the rankings, and then excludes from the selected subset the feature with the minimum contribution to the regression model (Guyon,

Algorithm 2: Recursive Feature Elimination

Input: F : Set of all potential objectives (features) T : Set of ranked solutions \mathbf{r} : Vector of ranks

and either

- $k \in [2, |F|] \subset \mathbb{N}$ (for fixed number of objectives) or
- $\tau \in (0, 1] \subset \mathbb{R}$ (for variable number of objectives)

```
1  $\hat{F} \leftarrow F$ 
2 while  $|\hat{F}| > 2$  do
3   Step 1:  $M \leftarrow \text{Build\_Model}(T, \mathbf{r}, \hat{F})$ 
4   Step 2:  $f_j \leftarrow \arg \max_{f_i \in \hat{F}} \phi(f_i)$ 
5   if Fixed number of objectives then
6     if  $|\hat{F}| = k$  then
7       break
8   else if  $\phi(f_j) < \tau$  then
9     break
10  Step 3:  $\hat{F} \leftarrow \hat{F} \setminus f_j$ 
11 return  $\hat{F}$  (selected objectives)
```

Weston, Barnhill, & Vapnik, 2002). There are several ways to measure the contribution of the i^{th} objective to the regressed model, and here to be consistent with uni-variate variant, we use the statistical significance level (p -value), $\phi(f_i) \in [0, 1]$, for the objective's coefficient in the regressed model, which can be calculated with any statistical software. In the next iteration, the model is built again using the pruned set of objectives. The process is repeated until the size of the pruned set is equal to k in the case of fixed number of objectives. For the variable number of objectives, the pruning stops when the remaining objectives are all significant.

The detection method using RFE is depicted in Algorithm 2.

4.4.2 Fixed versus Variable Number of Active Objectives

The number of features (active objectives) selected can be defined in different ways. Here we explore the following two alternatives:

Fixed Number of Objectives (k)

The optimization starts with k active objectives and this number is kept constant throughout the optimization process such that activating an inactive objective implies deactivating an active one. The benefit of this approach is that the iEMOA only needs to handle a specific number of objectives, which is simpler than handling a variable number of objectives. The

downside is that some relevant objectives may remain hidden if the number of relevant objectives $|F_{\text{DM}}|$ is larger than k . Thus, the goal is to identify the k most relevant objectives for the DM out of all potential objectives.

Variable Number of Objectives

We select the subset of objectives that meets a predetermined threshold τ . The lower the value of τ , the lower would be the number of objectives with acceptable p -values. If there is only one objective with a p -value lower than τ , then the two objectives with lowest p -values are selected instead. For RFE, the process stops if the size of the pruned set of objectives is reached 2.

Having two feature selection methods and two approaches with fixed and variable number of objectives as explained above, we have four total variations defined as follows:

1. k -HD: Uni-variate feature selection with fixed number of objectives
2. τ -HD: Uni-variate feature selection with variable number of objectives
3. k -HDR: RFE with fixed number of objectives
4. τ -HDR: RFE with variable number of objectives

The convention used for naming the variants can be described as follows; the prefix k indicates the variants with fixed number of objectives and the prefix τ specifies the variable number of objectives. HD indicates that Hidden/irrelevant objective Detection is on. The suffix R is added when the recursive feature elimination is used. The proposed methods can be applied to any ranking-based iEMOA for objective reduction and/or detection of hidden objectives in order to find the objectives that are relevant to the DM. Here, we will focus on extending BCEMOA (Battiti & Passerini, 2010) with our proposed method. We chose BCEMOA due to its popularity and availability of the source code. The proposed method however, can be integrated with any ranking-based algorithm. In what follows, the modified BCEMOA, here called BCEMOA-HD, is explained in detail.

4.4.3 BCEMOA-HD

BCEMOA (Battiti & Passerini, 2010) is an iEMOA based on NSGA-II. It starts with a population of randomly generated solutions (pop), and the population is evolved with NSGA-II for gen_1 generations. Next, at each interaction step, the best N_{exa} solutions are selected from the evolved population, all potential objectives are evaluated and presented to the DM for

ranking. The objective vectors and their ranks are then used to train a support vector machine (SVM) model to learn a utility function (U_{svm}). The learned U_{svm} replaces the crowding distance in the next generations. Further interactions with the DM provide additional samples to re-train the SVM model and improve the predictions of the learned utility function.

Similar to the original BCEOMA, the BCEMOA-HD algorithm, proposed here, starts with a set of active objectives \hat{F} . All inactive objectives ($F \setminus \hat{F}$) do not need to be evaluated during the optimization and do not participate in dominance ranking and evolution of the population. At each interaction with the DM, all objectives in F are evaluated for the solutions that are presented to the DM. Immediately after each interaction, the feature selection method described in Section 4.4.1 is applied to the objective vectors and their rankings to identify relevant objectives and update \hat{F} . Consequently, the population may need to be updated by evaluating any objective $f_i \in \hat{F}$ that has become active. SVM is also used to learn U_{svm} based on active objectives in the updated \hat{F} and their rankings. An overview of BCEMOA-HD is shown in Algorithm 3.

As described above, compared to the original BCEMOA, we have modified the algorithm in Lines 10 and 11, where feature selection is deployed and the set of active objectives is updated, respectively. A further modification applied to the original BCEMOA is in the selection of the best solutions presented to the DM. When there is no variance in the values of some objective, for example, because its values are near-optimal, then their correlation with the rankings provided by the DM is undefined, and their p -value will be set to 1. As a result, the feature selection will deactivate the objective and replace it with an irrelevant objective that might have a higher correlation by chance. To preserve the elitism and avoid losing the DM's desired solution, the solution that was ranked best by the DM in the last interaction is always included in the next set of solutions presented to the DM by BCEMOA-HD. This way, we make sure we will not lose the DM's most desired solution so far and the utility of the selected solution does not decrease.

4.5 Experimental Setup

To evaluate the effectiveness of the proposed method, we design a set of experiments that cover different aspects of the problem of identifying hidden and irrelevant objectives. In the experiments with variable number of objectives, we seek to investigate how the methods perform for objective reduction purposes, thus all objectives are active from the start of the run ($\hat{F} = F$). In the case of fixed number of objectives, only specific objectives are active ($\hat{F} \subset F$) at the start.

Although this research is motivated by real-world scenarios, it would be very difficult to

Algorithm 3: BCEMOA-HD

Input:

N_{int} : Total number of interactions
 N_{exa} : Number of training examples per interaction
 pop : Population of solutions
 gen_1 : Generations before first interaction
 gen_i : Generations between two interactions
 F : Set of potential objectives
 \hat{F} : Set of active objectives
 and either

- $k < |F|$ (for fixed number of objectives) or
- τ (for variable number of objectives)

```

1  $T \leftarrow \emptyset, r \leftarrow \emptyset$ 
2  $pop \leftarrow$  run NSGA-II for  $gen_1$  generations
3   optimizing only  $\hat{F}$ 
4 for 1 to  $N_{\text{int}}$  do
5    $T_i \leftarrow$  select  $N_{\text{exa}}$  solutions
6   Evaluate solutions in  $T_i$  for all objectives in  $F$ 
7    $r_i \leftarrow \text{DM\_ranks}(T_i)$ 
8    $T \leftarrow T \cup T_i$ 
9    $\mathbf{r} \leftarrow \mathbf{r} \cup r_i$ 
10   $\hat{F} \leftarrow \text{feature\_selection}(F, T, \mathbf{r}, k \text{ or } \tau)$ 
11  Evaluate  $pop$  for  $f_i \in \hat{F}$ 
12   $U_{\text{svm}} \leftarrow \text{train\_SVM}(\{z_{ji} \mid \mathbf{z}_j \in T \wedge f_i \in \hat{F}\}, \mathbf{r})$ 
13   $\text{Crowding\_Distance} \leftarrow U_{\text{svm}}$ 
14   $pop \leftarrow$  run NSGA-II for  $gen_i$  generations
15    optimizing only  $\hat{F}$ 
16 return Best  $\mathbf{x} \in pop$  ranked first by non-dominated sorting and then  $U_{\text{svm}}$  considering only  $\hat{F}$ 

```

analyze the methods using complex real-world problems and human DMs. Instead we use well-known benchmarking problems from the literature, which we extend to simulate hidden and irrelevant objectives, and we simulate a human DM using various utility functions.

In what follows, a detailed description of the design of the experiments is laid out.

4.5.1 Simulation of Active and Inactive Objectives

We create synthetic problems that feature irrelevant and hidden objectives by extending existing benchmark problems as follows: Given a problem with $m = |F|$ potential objectives, we extend it with $\mathbf{d} \subseteq [1, m]$, an ordered index set of active objectives such that $i < j \rightarrow d_i < d_j$, specifying which objectives are active (optimized), i.e., $i \in \mathbf{d}$ iff $f_i \in \hat{F} \subseteq F$, where f_i is the i^{th} potential objective function. That is, given a solution \mathbf{x} , whose objective vector is $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$, the optimizer only considers $\hat{\mathbf{f}}(\mathbf{x}) = (f_{d_1}(\mathbf{x}), \dots, f_{d_m}(\mathbf{x}))$ and is able to change the set of active objectives by changing the indices in \mathbf{d} .

On the other hand, feature selection methods and the DM have access to $\mathbf{f}(\mathbf{x})$. In particular, when asked to rank a solution \mathbf{x} , the simulated DM evaluates $U(\mathbf{f}(\mathbf{x}))$, where U is the

utility function that measures the DM’s preferences, and U simulates irrelevant objectives by disregarding those components of \mathbf{f} , as we explain below.

The above technique can be applied to any multi-objective optimization problem. We describe next the underlying benchmark problems used in our experiments.

4.5.2 Underlying Benchmark Problems

We applied the above technique to two well-known numerical and binary benchmark problems, namely, multi-objective NK landscape problems with correlation between objectives (ρ MNK) (Verel et al., 2013) and DTLZ problems (Deb et al., 2005) with $m \in \{4, 10, 20\}$ objectives. Problems with $m = 4$ help us to better understand and investigate the dynamics of the proposed methods, while larger number of objectives allows us to evaluate the efficiency of the feature selection with variable number of objectives in many-objective problems.

ρ MNK problems allow us to analyse the effects of correlation among objectives and smoothness of the landscape on the performance of the proposed method. We consider ρ MNK instances with different values of correlation among objectives $\rho \in \{-0.25, 0, 0.25, 0.5, 0.75, 0.9\}$, taking into account the restriction that $\rho \geq -1/(m-1)$ (Verel et al., 2013) and different values of parameter K , which controls the smoothness of the landscape, namely, $K \in \{1, 4, 6, 8\}$ for problems with 4 objectives and $K \in \{1, 5, 10, 15\}$ for many objective problems, considering the constraint $K < n$. The greater the value of K , the more rugged is the fitness landscape. The value of n is kept fixed at 10 for problems with $m = 4$, 20 for problems with $m = 10$, and 30 for problems with $m = 20$ for ρ MNK problems.

From the DTLZ test suite, we focus on DTLZ1, DTLZ2 and DTLZ7, which were also used in the experiments of the original BCEMOA (Battiti & Passerini, 2010) and also in (Brockhoff & Zitzler, 2007b) for objective reduction. DTLZ1 contains $11^k - 1$ local Pareto-optimal fronts. Thus, it can be used to test the ability of the algorithm to deal with multiple local attractors. DTLZ2 has a concave Pareto front. Finally, DTLZ7 has 2^{m-1} disconnected Pareto-optimal regions in the objective space and is used to check the diversity of the solutions and the performance of the algorithm in disconnected feasible space.

As suggested in (Deb et al., 2005), the decision space dimension (n) is $m + 4$ for DTLZ1, $m + 9$ for DTLZ2 and $m + 19$ for DTLZ7. With DTLZ problems, optimizing a subset of objectives will optimize the rest of the objectives as well. To make the problem more challenging and also to avoid collapsing the PF to one point when projected to $k < m$ objectives, we follow (Brockhoff & Zitzler, 2007b) and map x_i to $x_i/2 + 0.25$, $i = 1, \dots, n$, for DTLZ2 and bound x_i within $[0.25, 0.75]$ for DTLZ1, which is also suggested in (Battiti & Passerini,

2010). Please note that this modification is not needed for DTLZ7 as it does not collapse to a single point.

4.5.3 Machine Decision Maker (MDM)

We adhere to the MDM framework introduced in (López-Ibáñez & Knowles, 2015) and simulate the DM's preferences with an utility function (UF) that explicitly expresses which objectives are relevant, so that we can assess the effectiveness of the methods proposed here for identifying relevant objectives.

We define \mathbf{c} as the ordered index set of relevant objectives such that $i < j \rightarrow c_i < c_j$, and consider the following quadratic UFs that were proposed in experiments on the original BCEMOA (Battiti & Passerini, 2010):

$$\text{UF1}(\mathbf{f}) = 0.28f_{c_1}^2 + 0.38f_{c_2}^2 + 0.29f_{c_1}f_{c_2} + 0.05f_{c_1} \quad (4.4)$$

$$\text{UF2}(\mathbf{f}) = 0.6f_{c_1}^2 + 0.05f_{c_1}f_{c_2} + 0.23f_{c_1} + 0.38f_{c_2} \quad (4.5)$$

$$\text{UF3}(\mathbf{f}) = 0.44f_{c_1}^2 + 0.14f_{c_2}^2 + 0.09f_{c_1}f_{c_2} + 0.33f_{c_1} \quad (4.6)$$

In addition, we consider the following Tchebychev UF

$$U_{\text{tch}}(\mathbf{f}) = \max_{i \in \mathbf{c}} w_i |f_i - f_i^*| \quad (4.7)$$

with $\mathbf{0}$ as ideal point f^* . The weights w_i of irrelevant objectives ($i \notin \mathbf{c}$) are set to zero while the weights of relevant objectives were manually selected for each problem such that the most preferred solution is away from the corner points as far as possible.²

In the UFs above, the DM only considers $F_{\text{DM}} = \{f_i | i \in \mathbf{c}\}$, while other objectives are irrelevant. Our simulation strategy assumes that relevant objectives are not strongly correlated with irrelevant ones. Otherwise, an objective that does not explicitly appear in the definition of a particular UF may still be identified as relevant by the proposed methods. We explain how we selected relevant objectives for each problem in the next section.

Although the UFs in Eqs. 4.4-4.7 are designed for minimization, we reverse and scale all utility values in the experiments to the range $[0, 1]$, such that 1 corresponds to the best utility value and 0 to the worst one, for consistency with multi-attribute utility theory (Keeney & Raiffa, 1993).

²The weights used for each problem are given in the Appendix B.

4.5.4 Selecting Relevant Objectives

Projection of the PF on lower dimensions might make it collapse to a single point, for some problems. This is true for DTLZ problems even when the problem is bounded (Brockhoff & Zitzler, 2007b). Thus, careful consideration should be given when using these problems to simulate active and inactive objectives. For instance, the PF of DTLZ7 collapses to a point if the first two objectives are active and the rest are inactive. After a careful examination, here the first and fourth objectives are selected as relevant for DTLZ problems ($\mathbf{c} = \{1, 4\}$). In the case of ρ MNK problems the first two objectives are selected ($\mathbf{c} = \{1, 2\}$). For ρ MNK problem with four objectives, having $F_{\text{DM}} = \{f_1, f_2\}$ and given an initial $\mathbf{d} = \{2, 4\}$, we can see that f_1 is a hidden objective (relevant but not optimized), f_2 is both relevant and optimized, f_3 is irrelevant and not optimized, and f_4 is irrelevant and optimized.

4.5.5 Evaluation of the Results

The experiments are performed in three different modes to enable the assessment of the proposed algorithms:

1. *Golden mode*: No interaction is done in this mode and the algorithm directly accesses the true UF of the DM instead of learning a UF. Moreover, only relevant objectives are optimised from the start to end. This is the ideal scenario.
2. *Only learning mode*: This mode corresponds to the original BCEMOA without any detection of hidden objectives. The algorithm does not have access to the DM's UF and instead a UF is learned from the rankings provided by the MDM, i.e. at each interaction the MDM uses its true UF to rank solutions. Predictions from the learned UF are used to rank non-dominated solutions, replacing the crowding distance in NSGA-II. The algorithm still uses non-dominated sorting as the first criteria to rank solutions. Both non-dominated sorting and the learned UF only consider the set of active objectives $\hat{\mathbf{f}}(x)$. The set of active objectives never changes, that is, \mathbf{d} remains constant throughout the run.
3. *Learning + detection mode*: This is our proposed BCEMOA-HD that performs detection of hidden objectives and is able to modify the set of active objectives. Within this mode, we test 4 variants of the HD method (see Section 4.4.2): k -HD, τ -HD, k -HDR, τ -HDR. Similar to the *Only learning mode*, the optimization algorithm relies on non-dominated sorting and an UF that is learned based on $\hat{\mathbf{f}}(x)$, and not on the MDM's true UF. However, in this mode, \mathbf{d} and subsequently $\hat{\mathbf{f}}$ may change after each interaction with

the ultimate goal of converging to the objectives that are actually relevant for the DM (F_{DM}).

Having these three modes makes it possible to evaluate the performance of the proposed method compared to the original BCEMOA and to the best solution that is achieved under an ideal scenario in *Golden* mode. The criteria for evaluation of the performance is the true utility value of the final solution returned by the algorithm. For *Learning + detection*, we also record the active objectives after each interaction to investigate how good the proposed method performs in detecting the relevant objectives during an optimization run. This allows us to measure the number of objective evaluations (see Definition 4.2.3). Understanding the relationship between utility and the computational/resource effort invested is of particular relevance to problems where objective evaluations are expensive, time-consuming and/or resource-intense (Allmendinger, 2012). In such problems, identifying high-utility solutions with as few as possible objective evaluations is preferred (or even needed in order to avoid premature termination of the optimization process due to a lack of resources). As a side effect, fewer objective evaluations means a reduced level of complexity, e.g. in case objectives are heterogeneous (Allmendinger & Knowles, 2021), and a reduced cognitive load on the DM as ranking is done considering fewer objectives comparisons in total.

Parameter Settings

All variants of BCEMOA use the parameter settings proposed in the original paper (Battiti & Passerini, 2010), including the parameters of the SVM learning model. In particular, the total number of generations is 500 and $N_{\text{exa}} = 5$ solutions are shown to the DM at each interaction. Within BCEMOA, NSGA-II uses a population size is 100 and creates 100 new solutions at each generation. NSGA-II runs for $gen_1 = 200$ generations before the first interaction and there are $gen_i = 30$ generations between subsequent interactions. The total number of generations after the last interaction is calculated as $500 - gen_1 - gen_i(N_{\text{int}} - 1)$. Thus, changing the number of interactions (N_{int}) would not alter total number of generations. We run experiments with 1, 3 and 6 interactions for DTLZ problems. For ρ MNK problems, we only consider 6 interactions and, instead, we investigate the effect of different levels of correlation (ρ) and ruggedness (K). Each algorithmic run was repeated 40 times with different random seeds.

Implementations

The algorithms, machine DM and ρ MNK problems are implemented in Python 3.7.6. The implementations of NSGA-II within BCEMOA and DTLZ benchmarks are provided by the

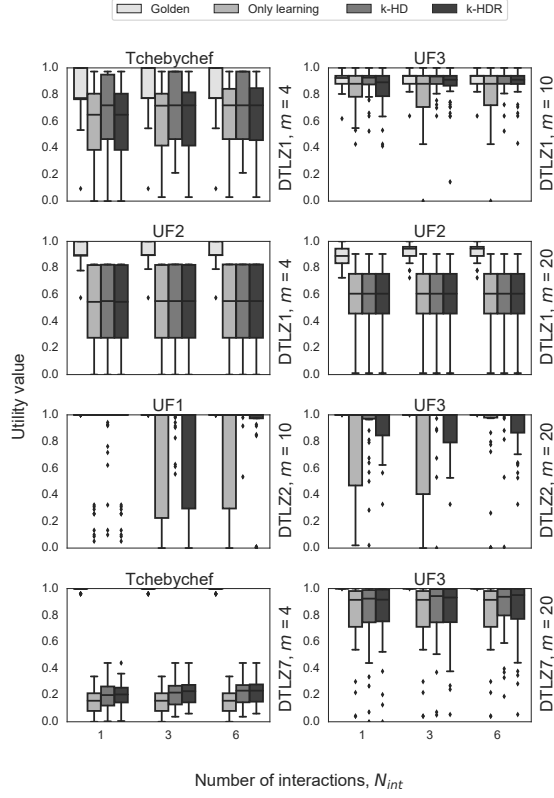


Figure 4.1. Comparison of the performance of different modes for DTLZ problems. The number of active objectives is fixed at $k = 2$. The vertical axis is the utility value (larger is better). The horizontal axis indicates the number of interactions.

Pygmo library 2.16.0 (Biscani, Izzo, & Yam, 2010b), the uni-variate feature-selection and RFE implementations are based on Scikit-learn 0.23.1 (<http://scikit-learn.org/>). To motivate further research, we make our code publicly available at <https://github.com/ShavaraniMahdi/Detecting-Hidden-and-Irrelevant-Objectives-in-Interactive-Multi-Objective-Optimization>.

4.6 Experimental Results

The interactive methods are designed to help the DM reach a satisfying solution and, thus, the utility value of the final solution returned by the algorithm is used to evaluate the performance of the interactive methods (Afsar et al., 2021). The utility values are normalized to the $[0, 1]$ interval in all the results. In this section, we focus on the most important findings and some figures are omitted to save space. The complete set of results and figures can be found in Appendix B.

4.6.1 DTLZ Problems with Fixed Number of Objectives

The results of experiments on DTLZ problems with a fixed number of active objectives are illustrated in Figure 4.1. When using BCEMOA-HD with DTLZ1 problem with $m = 4$

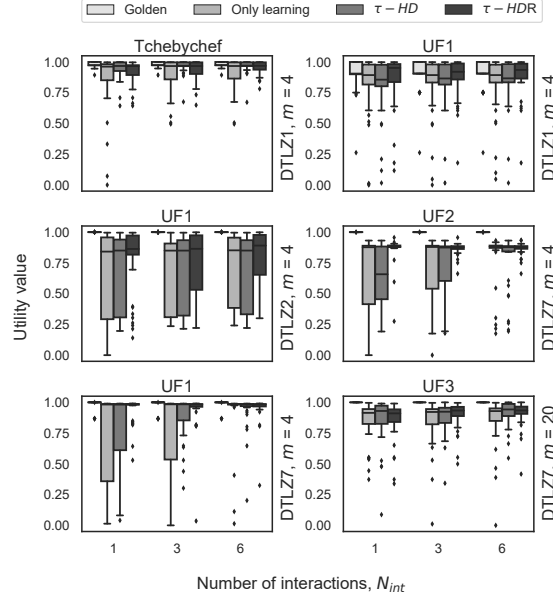


Figure 4.2. Comparison of the performance of different modes for DTLZ problems. The number of active objectives is not fixed ($\tau = 0.02$) and detection mode is used as an objective reduction technique. The vertical axis is the utility value (larger is better). The horizontal axis indicates the number of interactions.

or $m = 20$ and fixed number of active objectives, almost no improvement is observed in utility value compared to *Only learning* when comparing means, although the BCEMOA-HD manages to find better solutions in some instances. However, when $m = 10$, the performance of k -HD and k -HDR are significantly better than *Only learning* and almost as good as *Golden* mode except for UF2, which still shows no improvement.

For DTLZ2, improvements in the performance can be seen when BCEMOA-HD is used in the case of $m = 10$ and $m = 20$ together with UF1 and UF2. Another important observation is the better performance of k -HDR with more interactions, although it fails to get as good as k -HD.

For DTLZ7, there are slight improvements when detection methods (k -HD, k -HDR) are used. Complete list of figures can be found in the Appendix B.

In general, we observe that the proposed methods can significantly improve the utility value of the final solutions with respect to *Only learning* for the UFs tested. In some cases where it fails to do so, the utility value is not deteriorated by the method.

4.6.2 DTLZ Problems with Variable Number of Objectives

In this set of experiments, the effectiveness of the BCEMOA-HD is investigated with regard to objective reduction capabilities and thus the number of active objectives is not fixed. Thus, the execution of the algorithms start with all the objectives being active. The key results of these experiments are illustrated in Figure 4.2. For DTLZ1 with $m = 4$, the τ -HD and τ -HDR

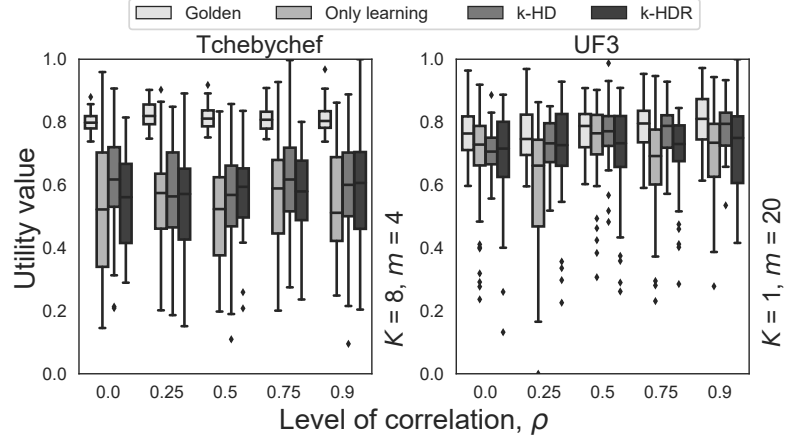


Figure 4.3. Comparison of the performance of different modes for ρ MNK problems. The number of active objectives is fixed to $k = 2$. The vertical axis is the utility value (larger is better). The horizontal axis indicates the ρ values.

perform better than the *Only learning* mode on Tchebychef UF, while for other UFs they have almost the same performance in terms of the utility value. With $m = 10$ and $m = 20$, τ -HD and τ -HDR perform as well as the *Golden* mode while τ -HDR is slightly outperformed by τ -HD. Results for DTLZ2 are identical to those of DTLZ1, i.e., τ -HD and τ -HDR perform as well as *Golden* mode and outperform *Only learning* mode with $m = 10$ and $m = 20$. For $m = 4$ τ -HD and τ -HDR outperform *Only learning* when UF3 is used, but cannot perform as well as *Golden* mode. In general, the superiority of τ -HD and τ -HDR compared to *Only learning* becomes more prevalent with a higher number of objectives. Although in some cases the algorithms return solutions with similar utility, we will show in Section 4.6.4 that the detection method results in significant computational savings.

4.6.3 ρ MNK Problems

The results of experiments on ρ MNK problems with a fixed and variable number of objectives are depicted in Figures 4.3 and 4.4, respectively. In most of these experiments, the proposed methods outperform the *Only learning* mode. In the case of the fixed number of objectives, in 30% of the cases this better performance of the proposed method is significant (based on the Wilcoxon signed-rank test with Holm's adjustment for multiple comparisons, p -value < 0.05). For the variable number of objectives, the significant observations increase to 50%. The detailed comparison of different modes related to Figs 4.3 and 4.4 can be found in the Appendix B. In the next section, we will further analyze the results and show that the relevant objectives are indeed detected, which means that a similar solution utility is reached in fewer objective evaluations and the proposed method can improve the computational efficiency of the interactive methods.

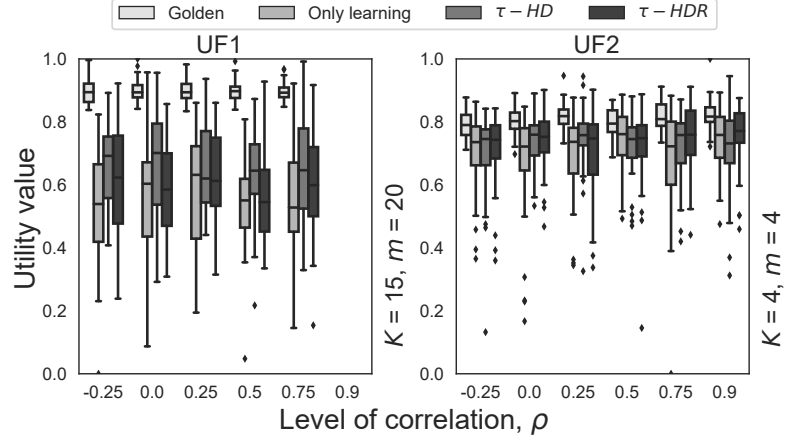


Figure 4.4. Comparison of the performance of different modes for ρ MNK problems. The number of active objectives is not fixed ($\tau = 0.02$) and detection mode is used as an objective reduction technique. The vertical axis is the utility value (larger is better). The horizontal axis indicates the ρ values.

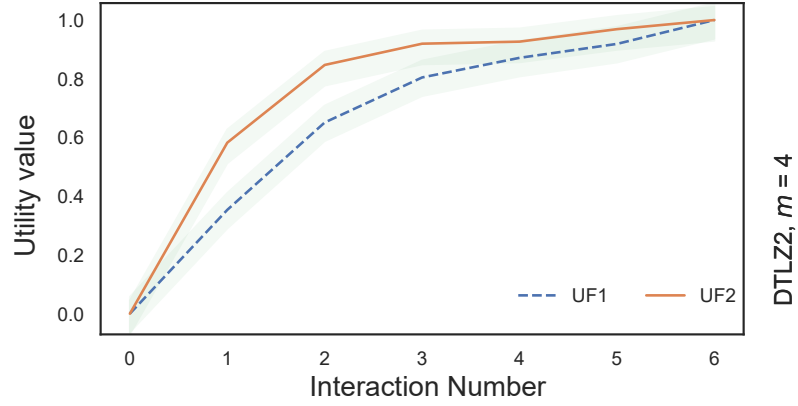


Figure 4.5. Utility of the best-so-far solution within a single run after each interaction on DTLZ2 problem when using τ -HD ($\tau = 0.02$). The lines show the mean value over 40 runs and the shaded area shows the 95% confidence interval around the mean. The results for other test problems are similar.

4.6.4 Further Analysis

Anytime Behavior within Each Run

Figure 4.5 illustrates the change in the utility value of the best solution gained after each interaction in a single run of the algorithm averaged over 40 runs. We observe that all interactions lead to some improvement in the utility of the best solution found, but the improvements become smaller with subsequent interactions.

Power of Detection of Relevant Set of Objectives

In terms of the power of the detection, a heatmap plot is provided in Figure 4.6. The plot illustrates the number of times the relevant and irrelevant objectives are activated by τ -HD across all experiments on ρ MNK problems with 10 objectives. The x-axis shows interactions

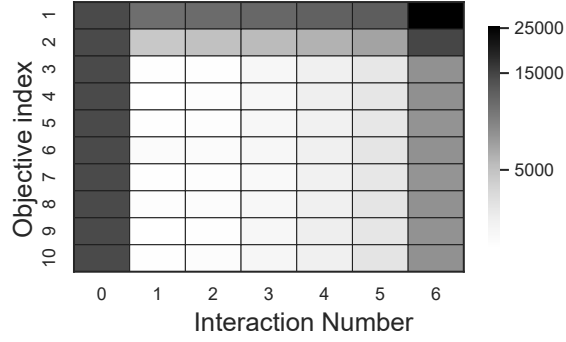


Figure 4.6. Convergence of the τ -HD ($\tau = 0.02$) towards relevant objectives ($c = \{1, 2\}$) through interactions on problem ρ MNK with 10 objectives. The x-axis indicates the interaction number and the y-axis indicates the index of the objective functions. The darker color, the higher is the number of the times the objective has been activated. The relevant objectives are found after the first interaction in the majority of the experiments.

within a single run. Interaction 0 refers to the state of the algorithm before the first interaction, when all objectives are active. The y-axis shows the index of all potential objectives. Each cell in the heatmap indicates the number of times the potential objective shown in the y-axis was active after the interaction shown in the x-axis. It can be observed that after the first interaction most of the objectives are deactivated while the first and second objectives, which are the relevant ones, are kept active. It can be easily verified that the τ -HD converges fast towards the relevant objectives. Another observation is that after the 6th interaction, almost all objectives become active, although to a lesser degree compared to the relevant ones. This observation is explained as follows: When the relevant objectives are optimized to their near-optimal value with respect to the DM’s UF, the values of these objectives will be nearly constant in the solutions presented to the DM. During feature selection, the correlation for such objectives would be undefined (Eq. 4.2), the F-statistic will be set to 0.0 and the p -value to 1.0, thus the objectives would be identified as irrelevant and replaced with inactive ones that, by chance, show some correlation with the rankings of the DM.

Analysis of the Threshold Parameter (τ) and Computational Efficiency

As an important parameter of τ -HD and τ -HDR, τ indirectly controls the number of active objectives; thus, careful examination should be given in determining its value. To inspect the effect of parameter τ , the DTLZ problems with $m = 20$ objectives are solved with different values of τ . The results for other problems are similar and hence not discussed here. Setting $m = 20$ provides a better illustration of the efficiency of the proposed method in reducing the computational requirements and objective evaluations. When $\tau = 1$, all objectives have p -value less than the threshold and, thus, all of them are active; this means no objective reduction is performed and the mode is identical to *Only learning*. The results in Figure 4.7 show that the performance of the τ -HD improves with lower values of τ on DTLZ problems.

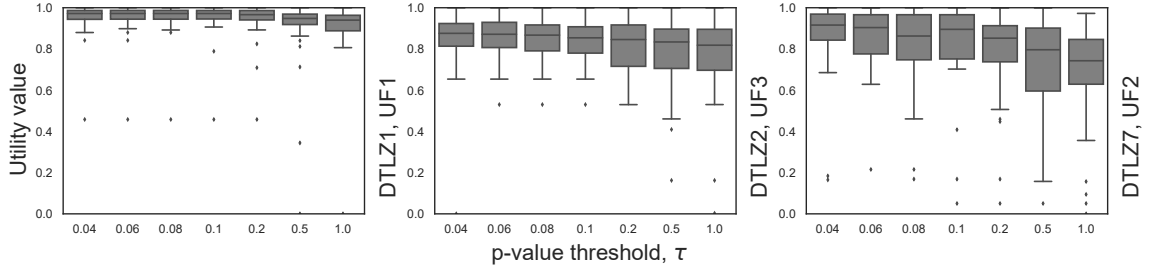


Figure 4.7. Performance analysis of τ -HDR with different values of τ for DTLZ problems with $m = 20$. The number of interactions in all runs is 6.

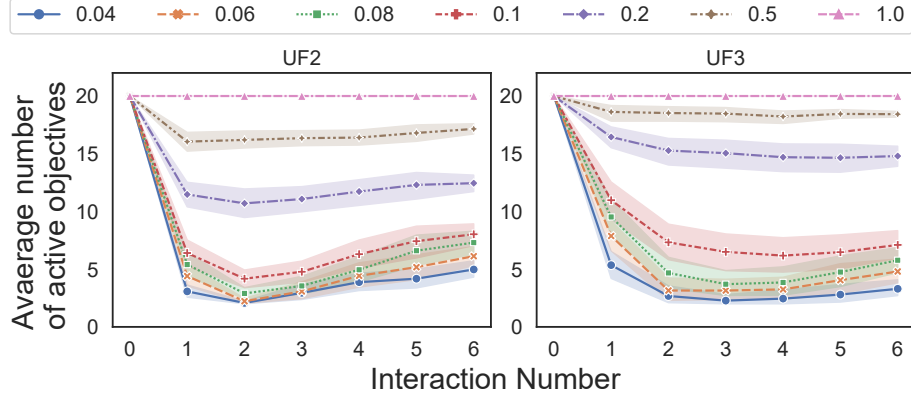


Figure 4.8. Number of active objectives within a single run after each interaction for different values of τ in τ -HDR mode on DTLZ1 (left figure) and DTLZ2 (right figure) test problems with 20 objectives. The x-axis indicates the interaction number and the y-axis is the number of active objectives after each interaction averaged over 40 runs. The shaded areas show the 95% confidence interval around the mean. The results for other test problems are identical.

On the other hand, reducing the τ value would reduce the number of active objectives (up to a minimum of 2 active objectives). A lower number of active objectives increases the efficiency of an EMOA and, by avoiding the evaluation of inactive objectives, possibly translates to savings in computational and economical costs. To illustrate these potential savings, Figure 4.8 shows the changes in the number of active objectives after each interaction, averaged over 40 runs. The shaded areas show the 95% confidence interval around the mean. It can be clearly verified that after the first interaction, the number of active objectives experiences a steep decrease. As expected, when $\tau = 1$, no objective reduction is performed.

Another important criterion is the ratio of evaluations of relevant objectives to those of irrelevant ones. Since inactive objectives are not evaluated during optimization, we measure the total number of objective evaluations (Def. 4.2.3) only for active objectives and we observe that τ -HD effectively reduces this total number. For instance, when $\tau = 1$ (equivalent to *Only learning* mode), an experiment on DTLZ1 with UF3, uses 600,000 objective evaluations and only 10% of these evaluations pertain to relevant objectives. However, when τ -HD is used, only 45,000 objective evaluations are done of which 30,000 (67%) are dedicated to relevant objectives. In general, objective evaluations are reduced by up to 80% compared to *Only learning* when τ -HD or τ -HDR is used. These savings could be used to run the optimizer for

longer, leading to improved solutions (although we do not explore this possibility here).

4.7 Conclusion and Future Work

This study has considered multi-objective problems that are solved by means of iEMOAs and where only an unknown subset of all the potential objectives are of relevance to the DM. In this context, we provided formal definitions of irrelevant, hidden and active objectives that complement the definition of redundant objectives already studied in the literature. We propose here a detection method that may be incorporated into any ranking-based iEMOA to identify irrelevant and hidden objectives. Furthermore, we show that an iEMOA able to dynamically change the active objectives can use this method to find solutions with higher utility for the DM in fewer objective evaluations. In addition, for the purpose of benchmarking, we propose a methodology for the simulation of irrelevant, hidden, and active objectives.

Two variants of the method with a fixed and variable number of active objectives were studied. The results show that the variant with the variable number of objectives is useful for dimension reduction purposes, reducing the number of active objectives even after the first interaction. This eliminates unnecessary evaluations of irrelevant objectives, thus saving computational effort, and improves the utility of the final solution returned by the iEMOA. The variant with a fixed number of active objectives has shown to be able to both remove irrelevant objectives and activate hidden ones. We also explored the application of recursive feature selection. However, the results indicate that there is no gain in using this method over the uni-variate feature selection. Comparing the results achieved for different test problems, we observed that the improvements in the final utility value are more significant for DTLZ problems. However, savings with regard to objective evaluations are achieved for both test problems. These savings may be most beneficial in problems where objective evaluations are expensive in terms of computational time, economical cost, or physical resources. We showed experimentally that the value of τ affects the number of active objectives and can be used as a tool to control this aspect. We considered four different UFs to simulate DMs with different preferences. Future studies should consider other UFs, such as the Sigmoid UF (Shavarani et al., 2021).

The methods proposed here are the first attempt to tackle the detection of irrelevant and hidden objectives, and there is scope to obtain further improvements. We observed that, in some experiments, once the relevant objectives have reached near-optimal values with respect to the DM's UF, the proposed methods may replace active relevant objectives with irrelevant ones. Thus, it would be desirable to introduce a mechanism that avoids such a behavior. Future studies should also consider DM simulations of learning and preference drift and how

our proposed detection method can cooperate with an iEMOA to detect and adapt to such changes. Our proposal relies on uni-variate feature selection based on the correlation between objectives and DM's rankings. Considering nonlinear regression in feature selection would be a subject worth studying. In the case of BCEMOA, the learned SVM model could be used to identify relevant objectives. However, our proposal here is more general and does not require that the iEMOA uses a specific learning model.

Chapter 5

An Interactive Decision Tree-Based Evolutionary Multi-Objective Algorithm

Shavarani, S. M., López-Ibáñez, M., Allmendinger, R. , & Knowles, J.

A summary of this chapter is presented at Evolutionary Multi-Criterion Optimization (EMO2023)

The chapter is Submitted to the European Journal of Operations Research

Recent research on interactive evolutionary multi-objective optimization algorithms has illustrated that an algorithm with good performance under ideal conditions may not perform well under real-life conditions where non-idealities such as cognitive biases are imposed. The main feature of any interactive method is how preferences are modeled and used to direct the search. Many studies model preferences as a linear utility function, which are too simple to reflect realistic decision-making behaviors. Some studies use extended binary classification techniques, such as one-vs-one, to learn to rank. Such an aggregated representation is not always desired and is vulnerable to significant errors with small biases in the preference data. Here, we propose decision trees to learn the preferences of the DM from interactions and automate fast pairwise comparisons based on the trade-offs of two solutions. To cancel out possible biases and errors in estimations, we use the trained tree in holistic comparisons to determine solutions that survive each generation. The experimental results confirm the superiority of our method in learning the DM's preferences and robustness towards biases and non-idealities. Notably, the algorithm has a significantly better performance in many-objective problems.

5.1 Introduction

Many real-life optimization problems have several conflicting objectives to be optimized simultaneously (Purshouse et al., 2014). Given m objective functions $(f_i(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R})$

and assuming minimization of all objectives without loss of generality, a Multi-Objective Optimization Problem (MOP) (Miettinen, 1999) has the following general form:

$$\begin{aligned} \min \quad & \mathbf{z} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to} \quad & \mathbf{x} \in \mathcal{X} \end{aligned}$$

where $\mathbf{x} \subseteq \mathbb{R}^n$ is defined as a decision vector from the non-empty set of feasible solution space $\mathcal{X} \subseteq \mathbb{R}^n$, and $\mathbf{f}(\mathbf{x} : \mathbb{R}^n \rightarrow \mathbb{R}^m)$ maps a decision vector to $\mathbf{z} \subseteq \mathbb{R}^m$ in the objective space. When $m > 3$, MOPs are considered as many-objective problems.

Definition 5.1.1 (Domination). A solution $\mathbf{x} \in \mathcal{X}$ dominates solution $\mathbf{y} \in \mathcal{X}$ if the two following conditions hold (Deb, 2001):

- $f(\mathbf{x}_i) \leq f(\mathbf{y}_i) \quad \forall i \in \{1, \dots, m\}$
- $f(\mathbf{x}_i) < f(\mathbf{y}_i) \quad \exists i \in \{1, \dots, m\}$

Definition 5.1.2 (Non-domination). A feasible solution $\mathbf{x} \in \mathcal{X}$ is said to be non-dominated if there is no solution $\mathbf{y} \in \mathcal{X}$ that dominates it. Non-dominated solutions are also known as Pareto optimal solutions.

Definition 5.1.3 (Pareto optimal set). The set of all Pareto optimal solutions.

Definition 5.1.4 (Pareto front). The set of all objective functions corresponding to the solutions in the Pareto optimal set is called the Pareto front (PF). In other words, PF is the image of Pareto optimal solutions on the objective space.

Due to the conflicting nature of objective functions in MOPs, it is generally impossible to have a single solution where all the objectives attain their optimal values. Instead, the general goal of solving MOPs is to reach a small subset of Pareto optimal solutions with interesting trade-offs or the most preferred one thereof. Evolutionary Multi-Objective Optimization Algorithms (EMOAs) naturally work with a population of solutions and can generate a representation of PF solutions in a single run. Hence they are well aligned with the requirements of MOPs. However, with an increase in the number of objectives, EMOAs lose their selection pressure, and their performance declines exponentially (Khare et al., 2003; Deb et al., 2005; Brockhoff & Zitzler, 2006; Ishibuchi et al., 2008). Interactive algorithms compensate for this issue by exploiting the DM's discrimination and preference information to build the DM's preference model, which is then used to direct the search and generate only those parts of the PF that are interesting to the DM. Thus, iEMOAs reduce computational costs and support the DM in reaching a desirable solution while incurring minimal cognitive effort (Branke et al., 2008; Greco et al., 2010).

However, recent studies (López-Ibáñez & Knowles, 2015; Shavarani et al., 2021) have indicated that interactive methods may not perform as expected under realistic conditions. Specifically, their performance declines with a higher number of objectives or when the elicited preference information is affected by human-specific biases such as inconsistent decisions (Shavarani et al., 2021) and fatigue (Zujevs & Eiduks, 2008). Thus, even the algorithms that perform well under ideal conditions seem to lack robustness to complications and biases that exist in real-life situations. The primary sources of the problem should be sought in the core features of an interactive method, which include interaction style, preference model, and the way the preferences are exploited inside the method to direct the search towards the Most Preferred Solution (MPS) (Xin et al., 2018).

To this end, we suggest adopting Decision Trees (DTs) in learning the preferences of the DM and developing a novel Decision Tree-based iEMOA: DTEMOA. DTs have been widely used in learning user preferences (Xia et al., 2008; Cheng et al., 2009; Yu et al., 2011; Dalip, Gonçalves, Cristo, & Calado, 2013), but not in interactive methods. As non-parametric supervised learning methods, DTs can be seen as a piece-wise constant approximation (Breiman, Friedman, Stone, & Olshen, 1984) that delivers classification by recursive partitioning of the input features (Rokach & Maimon, 2007). Utilizing DTs does not require any particular assumptions about the DM's preference model. DTs are competitive with other learning methods in terms of accuracy and speed, superior in terms of interpretability and comprehensiveness (Cheng et al., 2009) and thus, are very popular among classification techniques (Mitchell, 1997; Zhao & Ram, 2004). Unlike other learning algorithms that act as black-box models, DTs are easy to understand, interpret and visualize. This characteristic is beneficial in helping human DMs understand and trust the process and final results (Allen et al., 2019). They do not require any pre-processing or scaling of the data, which is not an easy task in MOPs with unknown PF, and they can work with different types of data from categorical to continuous (Allen et al., 2019). DTs naturally detect relevant features and simplify the learned model (Xia et al., 2008; Yu et al., 2011). Combining feature selection and learning is particularly beneficial in many-objective problems to improve the accuracy of the learned model.

We use binary classification DTs to learn the desirable trade-offs in pairwise comparisons and to predict the preferred solution from the DM's perspective, overcoming problems with learning-to-rank algorithms, which mainly are in the form of an ensemble of estimators making them susceptible to significant errors when confronted with small biases and non-idealities such as inconsistent decisions (Cheng et al., 2009). Besides, using DTs, we do not make any assumption on the DM's preferences and the learned model is in terms of simple rules. The experimental results presented in this chapter indicate that the performance of the

proposed algorithm is competitive with other iEMOAs in terms of the desirability of the final solution returned by the algorithm and robust to biased and inconsistent preferences.

In what follows, a summarized review of previous efforts on interactive methods is provided. DTs are introduced in Section 5.2.3. The full details of the algorithm and the way DTs are utilized in DTEMOA are explained in Section 5.3. Experimental design is laid out in Section 5.4, and the results are discussed in Section 5.5. Further insight into the preference learning and critical differences of DTEMOA with other interactive methods compared in this study are set out in Section 5.5.1. Finally, conclusions and future research directions are provided in Section 5.6.

5.2 Literature Review

Depending on the stage that the preference information is elicited from the DM, Multi-Objective Optimization algorithms can be categorized into “no-preference”, “a priori”, “a posteriori”, and interactive methods (Horn, 1997; Van Veldhuizen & Lamont, 2000).

“No-preference” methods aim to find neutral compromise solutions without any need to preference information from the DM. Instead, some assumptions are made about what a reasonable compromise/trade-off would be (Miettinen et al., 2008). Methods of Global Criterion (Yu, 1973; Zeleny, 1973) and Neutral Compromise Solution introduced by Wierzbicki (1999) fall within this category. In “a priori” methods, DM provides preference information before starting the optimization process. This information may be in the form of aspiration levels, goals, or weights of objective functions, mainly used to determine the lexicographic order or a linear/non-linear aggregation of objectives. Examples of these methods can be found in (Fonseca & Fleming, 1993; Deb & Sundar, 2006; Jaimes, Montaña, & Coello, 2011).

“A priori” methods bring the decision-making phase before the optimization phase. Not being aware of the possibilities, limitations, the form of the objective space, and consequences of the decisions, the DM may be too optimistic or pessimistic in defining the preference information and bias the search direction (Branke et al., 2008; Miettinen et al., 2008). Scalability is another concern, and the difficulty in determining the preferences increases with the number of objectives (Rachmawati & Srinivasan, 2006).

In “a posteriori” methods, a set of well-distributed PF solutions is generated, from which the DM selects the MPS. The number of solutions required for covering the whole PF increases exponentially with the number of objectives (Singh et al., 2011), which, in turn, increases both the required cognitive effort during the decision-making phase and computational costs during the optimization phase, preventing the algorithm from getting close to

the actual PF (Miettinen et al., 2008).

Interactive methods overcome problems with “a posteriori” and “a priori” methods by interlacing optimization and preference elicitation (decision making) to support the DM in reaching a desirable result (Miettinen et al., 2008). The decision-making phase elicits preference information to guide the search in the following optimization phase towards areas of the PF that are interesting to the DM (Miettinen, 2008; Greco et al., 2010) or in the last step to select the MPS among the Pareto optimal ones. The DM can learn more about his preferences and become more informed about the problem during the process (Miettinen et al., 2008). Thus, interactive methods help the DM justify his expectations and achieve a satisfying solution, a process referred to as psychological convergence (Miettinen, 2008).

The preference information can be elicited directly or indirectly (Branke et al., 2010). In the direct approach, the DM is required to specify some parameters of the solution process, such as reference points (López-Jaimes & Coello Coello, 2014), weights (Narukawa et al., 2016b), and bounds (Ruiz et al., 2019). In contrast, in indirect preference elicitation the DM is required to make some holistic judgements, also known as exemplary decisions, that may be in the form of pairwise comparisons (Greenwood, Hu, & D’Ambrosio, 1996; Phelps & Köksalan, 2003; Battiti & Passerini, 2010; Branke et al., 2010; Branke, Greco, Słowiński, & Zielniewicz, 2015; Branke et al., 2016), selecting the best among a small subset of solutions (Fowler et al., 2010a; Köksalan & Karahan, 2010), ranking a subset of solutions (Deb, Sinha, Korhonen, & Wallenius, 2010b; Sinha et al., 2018), and selecting best or worst among a subset of solutions (Köksalan & Karahan, 2010; Chugh, Sindhya, Hakanen, & Miettinen, 2015; Hakanen, Chugh, Sindhya, Jin, & Miettinen, 2016). Providing direct information imposes higher cognitive effort on the DM (Battiti & Passerini, 2010). Interactive methods are also different in the optimizer they use. In this regard, NSGAII (Deb et al., 2002) has been widely used as the search engine to drive interactive methods (Battiti & Passerini, 2010; Pedro & Takahashi, 2014). However, the core part of interactive methods is how they model DM’s preferences. In this regard, iEMOAs can be categorized into two broad categories of non-ad-hoc and ad-hoc methods (Miettinen, 1999).

5.2.1 Non-ad-hoc Methods

Application of Multi-Attribute Utility Theory (MAUT) (Keeney & Raiffa, 1993) has been predominant in preference modeling, and utility functions (UFs) have been a popular way to represent DM’s interest in a particular solution (Rachmawati & Srinivasan, 2006; Afsar et al., 2021). Non-ad-hoc methods assume the existence of an underlying utility function that derives the decisions of the DM (Steuer & Gardiner, 1991). Thus, these methods attempt to

estimate such a UF in line with DM's decisions (Jaszkiewicz, 2004; Quan et al., 2007a; Battiti & Passerini, 2010). The estimated UF may be used as a scalarization function to transform the problem into a single objective problem, as a criterion to break ties between solutions on the same front, or to automate pairwise comparisons.

Zionts and Wallenius (1976) used weight space reduction to learn a linear UF's weights. The method is built on assumptions such as the concavity of objective functions and the convexity of constraints and was later extended to handle pseudo-concave value functions (Zionts & Wallenius, 1983). A similar approach was used in Quan et al. (2007a) and (Battiti & Campigotto, 2010). Despite most methods that use the UF for ranking the solutions, the former uses the estimated UF to determine the dominance relations among any two solutions that the DM does not evaluate.

Thiele et al. (2009) use expressed aspiration levels in building a scalarization function for ranking the population. In (Miettinen & Mäkelä, 2006), the DM classifies objectives as needing improvement or otherwise. Then, several scalarizing functions are formulated synchronously using the same preference information. Generally, this study is the extended version of previous research discussed in (Miettinen & Mäkelä, 2000).

In Necessary-preference-enhanced Evolutionary Multi-objective Optimizer (NEMO1) (Branke et al., 2010) the DM's preferences are used to derive a set of compatible linear value functions, and the most representative value function replaces the crowding distance. NEMO1 was limited to linear UFs, which are not capable of reflecting a realistic image of decision-making behaviors. Later, they proposed a revised version of NEMO that lifts this constraint on the shape of the UF to some extent (Branke et al., 2016).

In IEMO/D (Tomczyk & Kadziński, 2019a), DM's pair-wise comparisons are used to detect upper and lower limits for weight vectors of compatible L-norm utility functions, which are used to rank individuals in the population. IEMO/D extends a basic variant of MOEA/D (Zhang & Li, 2007).

Brockhoff et al. (2014) uses DM's input to estimate the weights of a hyper-volume-based indicator. The contribution to the weighted hyper-volume indicator is used as the fitness of each solution within the EMO algorithm. Battiti and Passerini (2010) ignores the assumption of any particular shape for the UF and leaves it to the Support Vector Machine (SVM) to fit the best UF to DM's provided ranks. Nevertheless, the UF shape is limited by the parameters of the SVM and the kernel settings. Otherwise, the computational costs would increase.

5.2.2 Ad-hoc Methods

Ad-hoc methods use different schemes to model and incorporate preferences. Aspiration levels and reference points have been deemed a convenient method of preference modeling (Ojalehto et al., 2016), and many studies try to bias the search and converge the population towards elicited aspiration levels or goals. (Fonseca & Fleming, 1993) is one of the first studies that used the declared goals to handle the optimization directions. Later, they gave the option to the DM to specify the order of priority of objectives used in comparing two solutions, and objectives with higher priority are compared first (Fonseca & Fleming, 1998). Ben Said et al. (2010) prioritize solutions closer to the aspiration level after dominance ranking. Similar iEMOAs that ask for reference points were proposed in (Deb & Sundar, 2006; Ruiz et al., 2015b). Such modeling of preferences using goals or aspiration levels ignores the critical notion of trade-offs between the objectives (Rachmawati & Srinivasan, 2006). (Branke, Kaussler, & Schmeck, 2001) is one of the earliest studies that utilize the trade-off information where the DM's desired trade-offs are used to modify the dominance relation to narrow the search space.

Specifying trade-offs, weights, and any form of providing parameters to each objective function is troublesome when the number of objectives increase (Rachmawati & Srinivasan, 2006). Fuzzy logic was also proposed for preference modeling (Rachmawati & Srinivasan, 2005). However, determining membership functions is computationally expensive, particularly in many-objective problems (Rachmawati & Srinivasan, 2006).

Köksalan and Karahan (2010) modeled DM's preferences as conditions that should be observed for approval of new solutions into the population to account for DM's preferences. In this regard, new solutions are accepted into the population if the distance to the closest individual in the population is greater than a predetermined threshold. DM's preferences are considered by defining a smaller threshold around the DM's selected solution, leading to a greater number of solutions in that region. The method's performance is highly dependent on the quality of the scaling. Otherwise, the direction of the search would be biased. Thus, applying the method to problems with unknown ideal and nadir points is not straightforward. Greco et al. (2010) apply Dominance-based Rough Set Approach (DRSA) to model DM's preference model in terms of some "if...then..." which is used to cut off non-interesting solutions from the population.

As summarized here, there have been many improvements in learning the preferences, and researchers have proposed complicated formulations to provide more realistic models. However, almost no study has examined the ability of those models to capture the preferences of the DM (Rachmawati & Srinivasan, 2006) and few studies that have scrutinized the per-

formance of interactive methods indicate they lack the expected performance and robustness under realistic conditions (Shavarani et al., 2021).

5.2.3 Decision Trees

One of the first DTs was ID3 (Quinlan, 1986), which was generally designed for nominal/integer features. Later, Quinlan (1993) created C4.5 to overcome the limitation of ID3 by transforming continuous values into discrete partitions. Classification and Regression Trees (CART) (Breiman et al., 1984) is a superior version of C4.5 that supports regression on numerical targets without a need to rule sets. Among all tree-building methods, CART is the most popular one (Cheng et al., 2009).

DTs have been widely used for ranking and preference learning. Allen et al. (2019) use trees to learn about and demonstrate the decision-making behaviors in ranking a given set of options. DTs have been used in web search ranking, learning user preferences in web browsing and patterns in navigating between web pages in (Mor & Minguillon, 2003; Zhu, Greiner, & Häubl, 2003; Fox, Karnawat, Mydland, Dumais, & White, 2005; Liu & Kešelj, 2007; Chrysostomou, Chen, & Liu, 2011; Svore & Burges, 2011).

Binary classification methods such as regression and SVM can not be used for multi-class classification. However, some extensions (such as one-vs-rest and one-vs-one) make this task possible by learning an individual model for each pair of labels and then aggregating the predictions into a unique ranking (Hüllermeier, Fürnkranz, Cheng, & Brinker, 2008; Murphy, 2012). Some other methods try to learn a utility function for each individual rank (Har-Peled, Roth, & Zimak, 2002; Dekel, Singer, & Manning, 2003; Bishop, 2006a). In one-vs-rest, each class label is contrasted with all other class labels, and the one with the highest probability is predicted. In one-vs-one, also known as *all pairs*, *pairwise learning*, *round robin learning*, a model is constructed for all possible pairs of classes and in predictions, the label with highest vote is predicted (Fürnkranz & Höllermeier, 2003). However, such a transformation of ranking values to real values and aggregation of different models are mostly susceptible to lots of bias (Cheng et al., 2009). Cheng et al. (2009) propose DTs to address the aforementioned problems. DTs can also be used for ranking a given set of options. There are two approaches for using DTs for ranking (Yu et al., 2011). One approach is to consider each rank as a class and then predict the label/rank of a given data. However, aside from the fact that this approach does not provide much knowledge on preferences and decision-making behavior, the predicted data should have the same number of items as the training data. The second approach is to consider all possible rankings as the target labels, the number of which adds up to $m!$, even without considering tie ranks. Thus, other studies modify the splitting

criteria or entropy to use DTs for ranking (Cheng et al., 2009). We propose using DTs, to our knowledge for the first time, in the context of an interactive EMOA for preference learning.

5.2.4 Decision Tree Classification

DTs consist of nodes that partition data according to a rule. “Root” is defined as a node at the top with no edge directed towards it. Other nodes with edges emanating from them are called internal nodes, while those with no edge going out of them are called “leaves”. Training a DT involves determining the criteria (feature) and its corresponding partitioning threshold that should be assigned to the root and its subsequent nodes to increase the information gain and accuracy of the tree. We provide a summary of the CART method (Breiman et al., 1984), which is used in this research. Interested readers are referred to the original book and references therein for further details.

Assume there exists a set of training examples $T = \{\mathbf{e} \in \mathbb{R}^m\}$, where m is the number of features/attributes. Further on, assume a vector $\mathbf{y} \in \mathbb{R}^{|T|}$, which holds a label, also called class variable or target attribute, for each training example in T . Each training example \mathbf{e} and its corresponding label y is denoted as (\mathbf{e}, y) . DTs partition the values of each feature such that similar target values are classified together. Let Q_o be the data at node o having N_o samples. Given a potential partition $\theta = (k, t_o)$ corresponding to the k^{th} feature ($1 \leq k \leq m$) and a threshold t_o , the goal is to split the data into right and left branches, which are respectively denoted as $Q_o^{\text{right}}(\theta)$ and $Q_o^{\text{left}}(\theta)$, such that:

$$\begin{aligned} Q_o^{\text{left}}(\theta) &= \{(\mathbf{e}, y) \mid \mathbf{e}_k \leq t_o\} \\ Q_o^{\text{right}}(\theta) &= Q_o \setminus Q_o^{\text{left}}(\theta) \end{aligned} \tag{5.1}$$

The best split is selected based on the value of a score $G(Q_o, \theta)$ and a given criterion $H(Q_o)$, which is calculated as follows:

$$G(Q_o, \theta) = \frac{N_o^{\text{left}}}{N_o} H(Q_o^{\text{left}}(\theta)) + \frac{N_o^{\text{right}}}{N_o} H(Q_o^{\text{right}}(\theta)) \tag{5.2}$$

where N_o^{left} is the number of samples on the left branch after applying the partition $\theta = (k, t_o)$. Definition of N_o^{right} follows accordingly. A candidate split $\theta^* = \arg \min_{\theta} G(Q_o, \theta)$ is then selected for that node.

The criterion, $H(Q_o)$, differs from one DT to another. In CART and for categorical classification, the criterion is called Gini Impurity, which calculates information loss or impurity and has the best performance in most problems (Breiman et al., 1984; Waheed, Bonnell, Prasher, & Paulet, 2006; Xia et al., 2008). Assuming the target values $l \in \{1, 2, \dots, L\}$ for

node o , the proportion of samples of class l to the samples of other classes in node o is:

$$P_{ol} = \frac{1}{N_o \sum_{y \in Q_o} \mathbf{I}(y = l)} \quad (5.3)$$

where \mathbf{I} is the indicator function. Consequently, Gini impurity is calculated as:

$$H(Q_o) = \sum_{l=1}^L P_{ol}(1 - P_{ol}) \quad (5.4)$$

5.3 Decision-Tree Based EMOA

We focus on indirect preference elicitation in our method, where the DM is asked to provide a ranking of a subset of non-dominated solutions at each interaction. We do not attempt to learn a complete ordering of a set of solutions. Instead, similar to (Quan, Greenwood, Liu, & Hu, 2007b), we try to learn how the DM performs pairwise comparisons. As in (Greco et al., 2010), the preferences of the DM are modeled by simple rules. We use DTs to generate such rules to predict the preferred (winner) solution in pairwise comparisons.

5.3.1 Preference Elicitation

In each interaction, a small sample of solutions is randomly selected from the population ($S \subseteq \text{pop}$, $|S| = N_{\text{exa}}$) and their objective vectors $Z = \{\mathbf{z} = \mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in S\}$ are presented to the DM. The DM ranks the options based on their utility. There is no requirement for a complete ordering of the solutions; the algorithm can handle a partial ordering. However, we assume the DM provides a complete ordering for simplicity. Let $a \succ b$ denote that a is preferred over b . We represent the total order provided by the DM with the vector \mathbf{r} such that the rank of \mathbf{z}_i is r_i (lower rank values are better) and $\mathbf{z}_i \succ \mathbf{z}_j \iff r_i < r_j$. Thus, given \mathbf{r} , it is possible to infer pairwise comparisons. For instance, the objective vector that is ranked 1 is preferred over another one ranked 2.

5.3.2 Preference Learning

We aim to learn a DM's preferences from the ranking provided as well as predict (simulate) her preference when evaluating the trade-offs of two objective vectors. DTEMOA infers pairwise orderings from the elicited preference information that can be in the form of a complete or partial ranking of a subset of solutions. To construct the training set T , we use the well-known pairwise transformation (Herbrich, Graepel, & Obermayer, 1999), i.e., for each pair

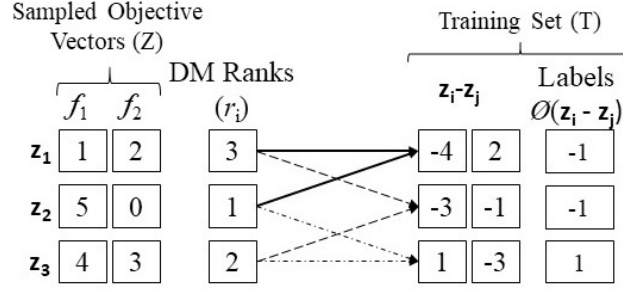


Figure 5.1. Construction of the training set from the ranked solutions over a problem with 2 objectives. The number of solutions ranked by the DM is $|Z| = 3$. For each pair of ranked solutions ($\mathbf{z}_i, \mathbf{z}_j \in Z$) an example, labeled by $\phi(\mathbf{z}_i - \mathbf{z}_j)$, is added to the training set. The size of the training set is $\binom{|Z|}{2}$.

of objective vectors $\mathbf{z}_i, \mathbf{z}_j \in Z$, we create a training example $\mathbf{z}_i - \mathbf{z}_j$ labeled with:

$$\phi(\mathbf{z}_i - \mathbf{z}_j) = \text{sign}(r_j - r_i) = \begin{cases} +1 & \text{if } \mathbf{z}_i \succ \mathbf{z}_j \\ -1 & \text{otherwise;} \end{cases} \quad (5.5)$$

i.e., an example is labeled $+1$ if the first solution is preferred over the second one and -1 , otherwise. An instance of such a process is given in Figure 5.1, where $Z \subset \mathbb{R}^2$, $|Z| = 3$, and each possible pairwise comparison of the ranked set generates an example of the training set T .

In the next step, T is used to train the DT to predict the preferred solution in pairwise comparisons of unseen data as well as the probability of such a decision. Subsequent interactions add more training examples to T , increasing the accuracy of the model. To predict the preferred solution when comparing two objective vectors \mathbf{z} and \mathbf{z}' , the trade-off vector $\mathbf{z} - \mathbf{z}'$ is given as the input to the trained DT, which predicts its class (label) $\phi(\mathbf{z} - \mathbf{z}') \in \{+1, -1\}$ indicating whether \mathbf{z} is preferred over \mathbf{z}' . The probability of the sample being in class $c \in \{+1, -1\}$ at leaf node o is calculated by $\frac{N_o^c}{N_o}$, where N_o^c is the number of samples of class c in node o , and N_o is the total number of samples in o .

5.3.3 Determination of the Score

Each solution in the population is given a score used to sort solutions. To calculate the score of a solution \mathbf{x} , it is compared with all other solutions in the population and the probability that it is preferred over the compared one is calculated using the trained DT. The sum of all such probabilities is the score of the solution \mathbf{x} :

$$\text{score}(\mathbf{x}) = \sum_{\substack{\mathbf{x}' \in \text{pop} \\ \mathbf{x} \neq \mathbf{x}'}} \Pr \{ \mathbf{f}(\mathbf{x}) \succ \mathbf{f}(\mathbf{x}') \} = \sum_{\substack{\mathbf{x}' \in \text{pop} \\ \mathbf{x} \neq \mathbf{x}'}} \Pr \{ \phi(\mathbf{f}(\mathbf{x}) - \mathbf{f}(\mathbf{x}')) = 1 \} \quad (5.6)$$

Algorithm 4: DTEMOA

Input: N_{int} : Total number of interactions
 N_{exa} : Number of solutions evaluated by the DM per interaction
 pop : Population of solutions
 gen_1 : Generations before first interaction
 gen_i : Generations between two interactions

Output: The most preferred solution

```
1  $T \leftarrow \emptyset$ 
2  $pop \leftarrow$  run NSGA-II for  $gen_1$  generations
3 for 1 to  $N_{\text{int}}$  do
4    $Z \leftarrow$  select  $N_{\text{exa}}$  solutions
5    $\mathbf{r} \leftarrow$  ask the DM to rank the solutions in  $Z$ 
6    $T \leftarrow T \cup \{(\mathbf{z}_i - \mathbf{z}_j, \phi(\mathbf{z}_i - \mathbf{z}_j)) \mid \forall \mathbf{z}_i, \mathbf{z}_j \in Z, \phi(\mathbf{z}_i - \mathbf{z}_j) = \text{sign}(r_j - r_i)\}$ 
7   Train DT using  $T$ 
8    $pop \leftarrow$  run NSGA-II for  $gen_i$  generations
9   replacing crowding distance with  $score$  (Eq. 5.6)
10 return best  $\mathbf{x} \in pop$  ranked first by non-dominated sorting and then  $score$ 
```

5.3.4 Using DTs in EMOAs

Given the above steps, let us explain how the preferences of the DM are considered in an EMOA to guide the search toward the most preferred parts of the PF. We use NSGA-II (Deb et al., 2002) as the underlying optimizer. After some generations of the NSGA-II, the algorithm stops to make the first interaction, elicits a ranking from a subset of solutions and consequently builds the training set T as explained above. A DT is trained using T to predict the preferred solution in pairwise comparisons. In the following generations, the DT is used to calculate the scores of the solutions (Eq. 5.6) to sort solutions with the same non-dominated sorting rank, i.e., the solution scores replace the crowding distance of NSGA-II to differentiate between non-dominated solutions. The solutions with a higher score have a better chance of survival and participation in mating and the generation of new offspring. The pseudo-code of the algorithm is illustrated in Algorithm 4.

When dealing with DMs, it is crucial to gain the DM's trust and confidence in the process and the results. When using DTs, learned trees can be conveniently visualized, summarizing all the rules elicited from the data set. Such a visualization is illustrated in Fig. 5.2, showing a DT that was built based on preference information elicited from the DM in an experiment on the DTLZ1 problem. The elicited information and the resulting training examples used for building the tree are moved to Appendix C to save space.

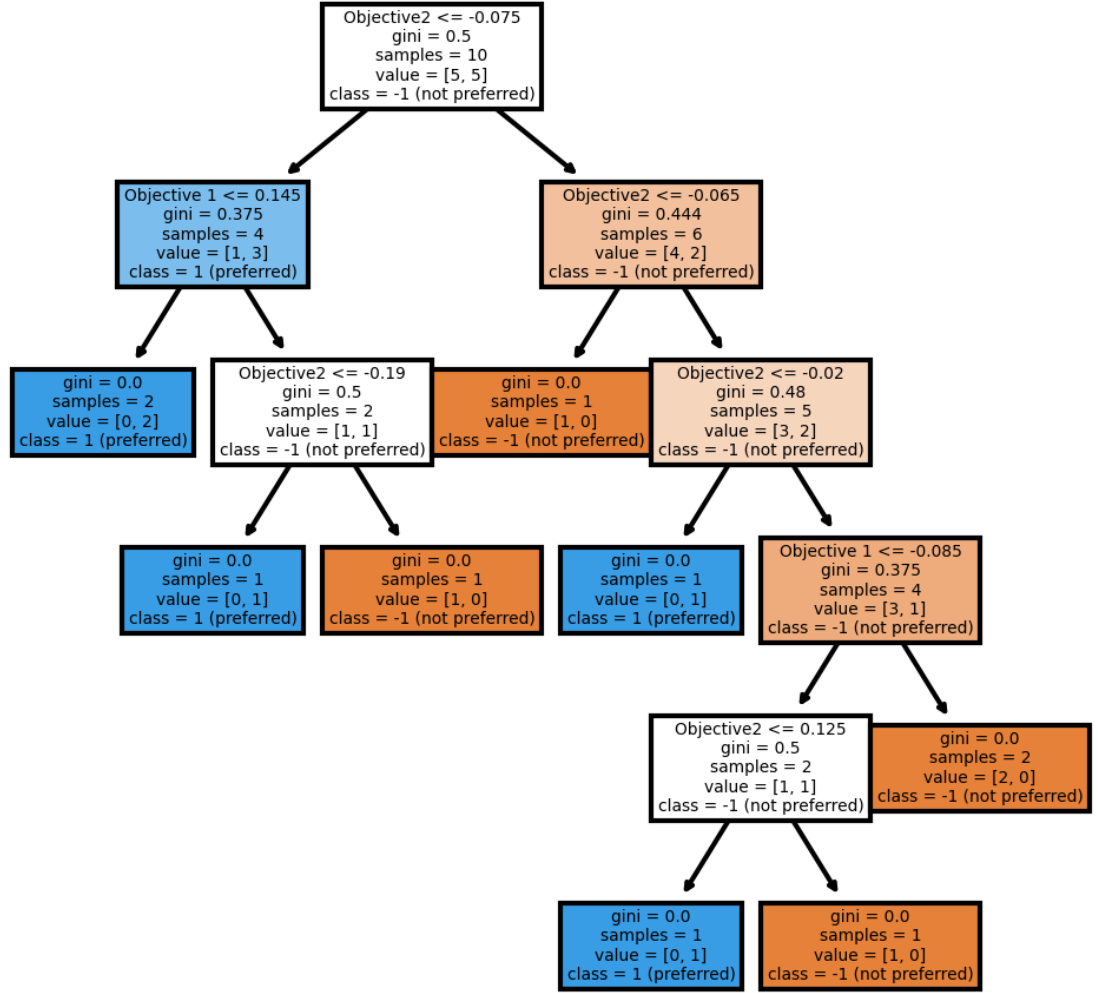


Figure 5.2. A decision tree example built on the information elicited in one interaction on problem DTLZ1 with $m = 2$ objectives. Here objective l indicates the trade-off value for l^{th} objective, i.e., $z_{il} - z_{jl}$ when comparing solution i with solution j . The *samples* field indicates the number of samples that fall on that node. The c^{th} element of *value* in each node o indicates N_o^c , the number of samples that belong to the c^{th} class. Gini impurity is a measure (from 0 to 0.5) of the quality of the split.

5.4 Experimental Design

5.4.1 Machine Decision Maker (MDM)

To simulate a human DM in the experiments, we have used the MDM introduced in (López-Ibáñez & Knowles, 2015) and improved in (Shavarani et al., 2021) (Chapter 3). The MDM is composed of a utility function (UF) that simulates the actual preferences of the DM and simulations of cognitive biases and other non-ideal decision-making behaviors that can happen in interactions with the DM. Here we consider the sigmoid UF introduced by Stewart (1996) for experimenting on iEMOAs. Particularly, we use a modified version which was introduced in Chapter 3 for minimization problems without violating any of its underlying assumptions

Table 5.1. Description of different DM behaviors simulated by combinations of τ_i and λ_i for minimization problems. The table is based on its original counterpart (Stewart, 1996) proposed in (Shavarani, López-Ibáñez, & Knowles, 2021) for minimization problems.

Case	τ_i	λ_i	Description
1	[0.6, 0.9]	[0.1, 0.4]	Mainly compensatory preferences.
2	[0.6, 0.9]	[0.6, 0.9]	Mainly compensatory preferences, but with sharp preference threshold.
3	[0.1, 0.4]	[0.1, 0.4]	Limited range of compensation, all higher values nearly equally undesirable.
4	[0.1, 0.4]	[0.6, 0.9]	Limited range of compensation, plus sharp preference threshold.

(hereafter called Stewart UF) and is formulated as follows:

$$U'(\mathbf{z}) = \sum_{i=1}^m w_i u_i(z_i) \quad (5.7)$$

$$u_i(z_i) = \begin{cases} \lambda_i + \frac{(1 - \lambda_i)(1 - e^{-\beta_i(\tau_i - z_i)})}{1 - e^{-\beta_i\tau_i}} & \text{if } 0 \leq z_i \leq \tau_i \\ \frac{\lambda_i \cdot (e^{\alpha_i(1 - z_i)} - 1)}{e^{\alpha_i(1 - \tau_i)} - 1} & \text{if } \tau_i < z_i \leq 1 \end{cases}$$

There are 4 parameters in Equation 5.7 that control the shape of this UF:

τ_i : Reference level, i.e., the point where losses are separated from gains with a steep inflation rate (loss can be interpreted as those values of the objective that are not satisfying to the DM).

λ_i : The utility value at the reference level.

α_i : The non-linearity of the function over losses.

β_i : The non-linearity of the function over the gains. Having $\alpha_i > \beta_i > 0$ satisfies various assumptions about the behavior of human DMs (Stewart, 1996; Shavarani et al., 2021).

One of the benefits of Stewart UF is the ability to simulate different decision-making behaviors, as depicted in Table 5.1. It has been shown that Stewart UF imposes more difficulties on the algorithms (Shavarani et al., 2021).

We also make use of the Tchebychef UF, which is formulated as follows:

$$U(\mathbf{z} = \mathbf{f}(\mathbf{x})) = \max_{i=1 \dots m} w_i |z_i - z_i^*| \quad (5.8)$$

where w_i is the weight of each objective function, and \mathbf{z}^* is the ideal or Utopian point. Since the selected benchmark problems are to be minimized and preserve consistency, it is assumed that the DM prefers solutions with lower utility values.

Aside from the tests under ideal conditions, we use the MDM capability of simulation of inconsistencies in the decisions of the DM. The MDM adds a normally distributed random noise with mean 0 and variance σ . The variance σ controls the amount of noise. To investigate the robustness of the algorithms, we test with $\sigma \in \{0.005, 0.01, 0.1, 0.2\}$.

5.4.2 Benchmark Problems

We follow (Battiti & Passerini, 2010; Köksalan & Karahan, 2010) and perform our experiments on well-known DTLZ benchmark problems (Deb et al., 2005). DTLZ1, DTLZ2, DTLZ7 are selected from the DTLZ test suite. Each of these problems exposes different difficulties to the algorithm and allows one to investigate its performance from various aspects. DTLZ1 contains $11^k - 1$ local PFs, and each of them can attract an EMOA before reaching the global PF. DTLZ2 investigates the performance of an EMOA in getting close to true PF. Finally, DTLZ7 has 2^{m-1} disconnected Pareto-optimal regions in the objective space. It is used to check the diversity of the solutions. The problem dimension n and the dimension of the objective space m are selected in a way that n is $m + 4$ for DTLZ1, $m + 9$ for DTLZ2 and $m + 19$ for DTLZ7, as suggested in (Deb et al., 2005); we consider $m \in \{2, 4, 10\}$. Also to make the problems more challenging, for DTLZ1 we follow (Battiti & Passerini, 2010) and limit x_i to $[0.25, 0.75]$, and for DTLZ2 we map x_i to $x_i/2 + 0.25$ as suggested by (Brockhoff & Zitzler, 2007b). We consider all possible combinations of different problem sets, number of objective functions, different value functions and different decision-making behaviors. Full details about the set of 45 test configurations can be found in Appendix C.

5.4.3 Evaluating Performance and the Competing Algorithms

Interactive methods are supposed to develop the best solution or a small subset of non-dominated solutions that maximize the DM's satisfaction. Thus, it makes sense to evaluate the performance of the interactive methods based on the utility value of the returned solution (Afsar et al., 2021). We compare the performance of DTEMOA against two state-of-the-art algorithms: BCEMOA (Battiti & Passerini, 2010) and iTDEA (Köksalan & Karahan, 2010). Preference learning in BCEMOA is similar to the way it is done by DTEMOA in that both learn to rank solutions, however, BCEMOA uses support vector machines (SVM) for preference learning. iTDEA uses a completely different preference learning scheme, where the preferences of the DM are reflected in the search process by prioritizing solutions in the proximity of the DM's selected solution. Both BCEMOA and DTEMOA use NSGA-II as their underlying optimizer. iTDEA still uses the same non-domination and mating methods, but in each generation only one solution is created and whether or not it is accepted to the population

Table 5.2. Parameter settings of the iEMOAs. In addition, total solution evaluations is 80 000, the population size $|pop| = 200$, the number of iterations is $N_{\text{int}} \in \{2, 3, 4\}$, and the DM ranks $N_{\text{exa}} = 5$ solutions per interaction. After N_{int} interactions, the algorithms continue running without further interaction until reaching N^{gen} generations.

Parameter	iTDEA	BCEMOA	DTEMOA
Total generations (N^{gen})	80 000	400	400
Generations before 1st interaction (gen_1)	$N^{\text{gen}}/3$	200	200
Generations between further interactions (gen_i)	$\frac{N^{\text{gen}}}{2(N_{\text{int}}-1)}$	20	20

depends on the position of that solution in the objective space and its distance to its closest solution in the population. The iTDEA is sensitive to the threshold parameter that controls the acceptable distance for the new solution. A large threshold would prevent solutions to enter the population, while a small one would make the population grow uncontrolled and increase computational costs, specially in many-objective problems. These similarities and differences were the main motivation behind selecting these two algorithms.

5.4.4 Algorithm Parameter Settings

The parameters of the selected algorithms are illustrated in Table 5.2. iTDEA generates and evaluates only one solution per generation. Thus, the number of generations is set to $N^{\text{gen}} = 80\,000$ to have an equal number of objective evaluations in all compared algorithms. As suggested in (Köksalan & Karahan, 2010), the first interaction of iTDEA happens after $\frac{N^{\text{gen}}}{3}$ generations, and the number of generations between each subsequent interaction is given by $gen_i = \frac{N^{\text{gen}}}{2(N_{\text{int}}-1)}$. Thus, with a larger number of interactions, gen_1 becomes smaller. The experimental study will investigate the impact of different numbers of interactions, N_{int} . The initial and final territory parameters of iTDEA are respectively set to 0.1 and 0.00001 in problems with two objectives, which was one of the alternatives suggested in (Köksalan & Karahan, 2010). For problems with $m = 4$ and 10 objectives, these values change to 0.5 and 0.25, respectively. Any smaller values for these parameters would make the size of the archive population large and the computational costs unaffordable. Other parameters for BCEMOA and iTDEA are set as suggested in (Battiti & Passerini, 2010; Köksalan & Karahan, 2010). The hyper-parameters of the DT in DTEMOA and SVM in BCEMOA are tuned by grid search cross-validation at each interaction (Shavarani, López-Ibáñez, Allmendinger, & Knowles, 2022).

We run experiments with a different number of interactions $N_{\text{int}} \in \{2, 3, 4\}$ to test the effect of interactions on the results. Our initial experiments indicate that a higher number of interactions does not seem to increase the quality of the solutions substantially. Each experiment is repeated 40 times with different random seeds.

5.4.5 Implementations

BCEMOA, DTEMOA, iTDEA, MDM and utility functions are implemented in Python version 3.7.6. The NSGA-II and DTLZ benchmark implementations were acquired from the Pygmo library 2.16.0 (Biscani et al., 2010b), the SVM ranking model from the Preference Learning Toolbox (Farrugia et al., 2015) powered by scikit-learn 0.23.1 (Pedregosa et al., 2011). Scikit-learn is also used to implement the DT models

5.5 Results & Discussion

This section is divided into two parts. In the first part, we focus on our proposed preference learning technique and provide some insight into its performance. In the second part, the performance of the proposed algorithm is compared with iTDEA and BCEMOA.

5.5.1 Assessing Ranking Performance

One of the main differentiating characteristics of iEMOAs is the way they adapt to the DM's preferences and the way they learn and model the preferences (Xin et al., 2018). The methods compared in our study differ significantly in how they interact and adapt to the DM. Both BCEMOA and DTEMOA try to learn a model to rank solutions but use different learning techniques. We can directly compare their accuracy independently of other algorithmic aspects. We evaluate the accuracy of the preference learning models of BCEMOA and DTEMOA on a randomly-generated population of 400 solutions for DTLZ7 problem with different number of objectives $m \in \{2, 4, 10\}$. It is also interesting to see how each algorithm exploits accumulated information elicited in different interactions. Thus, we simulate 5 interactions. Before each interaction, NSGA-II is used to evolve the population for 20 generations, and 5 solutions are selected randomly from the non-dominated front and ranked by a UF. The preference learning of both algorithms is applied to the elicited data. Then the accuracy of each model in ranking the non-dominated solutions in the population is measured by counting the proportion of correct pairwise rankings to all possible pairwise rankings:

$$\text{Acc} = 100 \cdot \frac{\sum_{i=1}^{|pop|-1} \sum_{j=i+1}^{|pop|} \mathbf{I}\{r_i < r_j \wedge \hat{r}_i < \hat{r}_j\}}{\sum_{i=1}^{|pop|-1} \sum_{j=i+1}^{|pop|} \mathbf{I}\{r_i < r_j\}}, \quad (5.9)$$

where \mathbf{I} is the indicator function which is equal to 1 if the given condition is true, otherwise 0; $r_i = U(\mathbf{f}(\mathbf{x}_i))$ is the true rank of the solution $\mathbf{x}_i \in pop$ according to the UF, and \hat{r}_i is the rank predicted by the learning models for the same solution. The experiments are repeated 40 times for each problem.

Table 5.3 summarizes the results. In particular, the accuracy of both methods is similar with 2 objectives. However, the preference learning technique in DTEMOA performs better when the number of objectives increases.

“nSol” indicates the number of non-dominated solutions that are sorted by the learned preference model (SVM for BCEMOA and DT-based for DTEMOA). It is easy to see that by increasing the number of objectives from 2 to 4, all the solutions become non-dominated with respect to the current population which is an expected behavior and problem of non-domination sorting. For problems with 2 objectives, both methods perform well in ranking the population with an accuracy of almost 100. With an increase in the number of objective functions, the accuracy of the methods decreases. The drop in performance is much more significant in SVM. With the accumulation of data over different interactions, the accuracy of the DT-based method increases significantly and almost recovers to 100. However, the SVM fails to achieve significant gains with larger preference data.

5.5.2 Comparison of the Performance with Other iEMOAs

The overall performance of the algorithms, measured by the utility value of the returned solution (lower is better), over all of the 45 tests with all combinations of the number of objectives, UFs, problems, and decision-making behaviours, is illustrated in Table 5.4 grouped in tests with the simulation of inconsistent decisions ($\sigma > 0$) and ideal conditions ($\sigma = 0$). DTEMOA outperforms BCEMOA and iTDEA and the Wilcoxon test indicates the significance of the differences (p -value 10^{-5}). The second important observation is the robustness of the DTEMOA towards noise and inconsistencies in the DM’s decisions. The performance of DTEMOA has declined by 1.6% when inconsistent decisions are simulated; this change is insignificant compared to the deterioration in the performance of BCEMOA(17%) and iTDEA(3.5%).

Figure 5.3 further details the performance of the algorithms with different levels of noise and inconsistencies in the decisions of the DM where the effects of 4 different levels of σ is evaluated. This figure only includes the most significant experiments where some of the worst and best performances of the algorithms are observed. In general, DTEMOA seems robust when noise is introduced into DM’s decisions, like the other two algorithms. However, its performance deteriorates with high levels of σ in some cases which are included in the Figure Figure 5.3. However, its performance still is better than iTDEA and BCEMOA when confronted with such high levels of noise in most cases, except few instances which are depicted in the figure.

In section 5.5.1, it has been shown that DTEMOA’s preference-learning improves with

Table 5.3. Mean accuracy (and standard deviation) of preference learning in DTEMOA and BCEMOA on the DTLZ7 problem for different UFs. The different number of interactions indicate how the algorithms exploit the accumulated preference data over interactions. The number of solutions presented to the DM at each interaction is $N_{\text{exa}} = 5$. nSol is the number of solutions in the non-dominated front that are ranked by the algorithms. nSol is equal to 400 in all interactions when $m = 10$ or 20. Thus, this information is omitted for brevity. The best accuracy for each UF, m and N_{int} is highlighted in bold face.

m	2			4			10		20	
Method	BCEMOA	DTEMOA	nSol	BCEMOA	DTEMOA	nSol	BCEMOA	DTEMOA	BCEMOA	DTEMOA
N_{int}	Linear									
2	89.4(11.3)	95.5(1.9)	122.45	71.9(11.3)	86.2(8.9)	400	67.2(6)	91.5(12.8)	57.9(3.4)	90.4(12)
3	87.2(11.7)	95.9(2)	353.07	70.4(8.1)	88.1(7.9)	400	67.1(5.9)	94.6(8.5)	59.4(3.9)	92(10.1)
4	89.2(8.1)	94.5(6.7)	400	72.2(4.8)	87(10.7)	400	67.4(5.9)	95.4(8.1)	60.2(4.5)	95.7(3.8)
5	91(5.6)	94.8(4.7)	400	73.5(4.1)	88.7(8)	400	67.1(6)	96.8(3.1)	60.8(3.6)	96.2(3.7)
	Polynomial									
1	97(5.1)	96.8(4.9)	30.92	78.4(10.8)	95.3(9.1)	368.48	64.8(5.4)	87.2(18.3)	55.5(4.3)	78.7(24.1)
2	94.2(9)	97.6(1.4)	122.45	75.7(10.4)	95.2(9.9)	400	68(5.8)	94.7(10.1)	58.2(3.6)	95.6(11.2)
3	92.6(10.2)	97.4(1.9)	353.07	74.8(8.4)	97.5(7.7)	400	67.5(6.5)	98.8(3.4)	59.8(4.1)	96.9(8.4)
4	94.1(8.2)	96.4(7.4)	400	75.8(5.5)	97.2(8.5)	400	68(5.4)	97.4(8.7)	60.8(4.7)	99.3(2.8)
5	94(6.4)	97.4(1.1)	400	76.5(5)	98.6(8.6)	400	68.1(6.1)	100(0)	61.5(4)	100(0)
	Stewart									
1	82.4(8.6)	79.4(11.4)	30.92	85.9(8.8)	80.9(6.3)	368.48	63.5(7.1)	59.1(6.8)	62.5(5.1)	55.2(8)
2	79.4(11.3)	84.8(12.1)	122.45	84.2(7.1)	83(7.2)	400	68.1(7.2)	62.2(6.5)	65.2(4.8)	59.5(7.8)
3	82.9(9)	84.9(11.4)	353.07	82.2(7)	82.9(5.7)	400	69.5(5.8)	67(6)	68.7(5)	62.1(9.3)
4	85.7(7.7)	86.5(5.5)	400	81.6(7.1)	83.5(2.9)	400	71.1(6.1)	69.3(6.3)	70.9(4.5)	64.5(6.6)
5	87.5(8.1)	84(11.3)	400	81(6.7)	81.8(6.5)	400	70.3(4.5)	71.3(7.1)	71.6(5.2)	71.8(7.9)
	Tchebychef									
1	100(0)	97.9(5.8)	30.79	77(11.4)	95.6(8.3)	370.30	64.1(5.7)	84.9(19.8)	55.3(4.2)	75.4(24.3)
2	100(0)	99.9(0.2)	122.83	75.5(11.4)	95(10.1)	400	67.3(5.8)	96.4(8.5)	58.5(3.7)	94.9(12.3)
3	100(0)	99.7(1.1)	354.17	74(7.8)	96.8(8.8)	400	67.1(6.5)	98.8(3.4)	60.1(3.6)	95.9(9.5)
4	100(0)	100(0)	400	75.6(5.8)	97(9.6)	400	67.7(5.8)	97.2(9.9)	61.1(5)	99(3.3)
5	100(0)	100(0)	400	76.7(5.1)	98.2(9.9)	400	68.3(6)	100(0)	61.6(4.1)	100(0)

more interactions. To test this hypothesis, we scrutinize the performance data with respect to the number of interactions in Table 5.5. The mean utility value of the final solution returned by DTEMOA improves from 0.184 to 0.177 when the number of interactions increases from 2 to 4. The same decrease is observed for BCEMOA, where the mean utility value is decreased from 0.206 to 0.203. The performance of iTDEA is surprisingly worsened by the increase of interactions. This finding is compatible with those in the original study of iTDEA (Köksalan & Karahan, 2010), where the authors have indicated that more interactions may lead to worsened results, particularly when more complicated utility functions are used.

It was also illustrated that the learning mechanism in DTEMOA surpasses BCEMOA's when the number of objectives increases. Table 5.6 outlines the performance of the algorithms for the different number of objective functions (m). The table shows the averaged utility value over all tests with 3 interactions, which seemed to be a reasonable number of

Table 5.4. Comparing the robustness of algorithms over experiments under ideal conditions and with the simulation of inconsistencies in the DM’s decisions. The utility values (to be minimized) are averaged over all tests. The values are rounded to 3 decimal points. The Wilcoxon test rejects the equality of the performance between algorithms (p -value $< 10^{-5}$), except for BCEMOA and iTDEA where the inconsistencies are simulated. The p -values are adjusted using Holm’s method for multiple comparisons.

	BCEMOA	DTEMOA	iTDEA	DTEMOA vs. BCEMOA	DTEMOA vs. iTDEA	BCEMOA vs. iTDEA
	mean utility value			p -value		
$\sigma = 0$	0.206	0.179	0.259	0.000	0.000	0.000
$\sigma > 0$	0.239	0.183	0.271	0.000	0.000	0.000

Table 5.5. Comparing the performance of the algorithms over over different number of interactions and under ideal conditions. The utility values (to be minimized) are averaged over all tests. The values are rounded to 3 decimal points. The Wilcoxon test is used to illustrate the significance level and the p -values are adjusted using Holm’s method for multiple comparisons.

	BCEMOA	DTEMOA	iTDEA	DTEMOA vs. BCEMOA	DTEMOA vs. iTDEA	BCEMOA vs. iTDEA
N_{int}	mean utility value			p -value		
2	0.206	0.184	0.214	0.000	0.032	0.192
3	0.206	0.179	0.259	0.000	0.000	0.000
4	0.203	0.177	0.278	0.000	0.000	0.000

interactions based on our observations from Table 5.5. It can be seen that BCEMOA and DTEMOA have the same performance on tests with 2 objectives. However, the mean utility values of final solutions achieved by BCEMOA are worse than DTEMOA’s by 11% in tests with 4 objectives. This difference is increased to 58% when the objective dimensions are increased to 10. This result can be attributed to the fact that decision trees have more attributes to draw rules with the increase in the objective function.

The performance of iTDEA is competitive to DTEMOA for tests with 2 and 10 objectives. Nevertheless, it has a bad performance in problems with 4 objectives. The performance of iTDEA is highly sensitive to its internal parameters, such as the territory and scaling parameters. There is no clear guideline on how to tune such parameters. Thus, we suspect that tuning such parameters might help increase the method’s accuracy. A task that does not seem trivial.

The same pattern is observed in experiments with the simulation of inconsistencies. The only difference is that the inconsistencies affect the performance of BCEMOA and iTDEA and even for tests with 2 objective, the differences in the performance are significant. Table 5.7 compares the algorithms when confronted with different DM types. The performance of DTEMOA is significantly superior to BCEMOA and iTDEA in all cases, except for tests with DM Type 4, when its performance is slightly worse than BCEMOA. However, the differ-

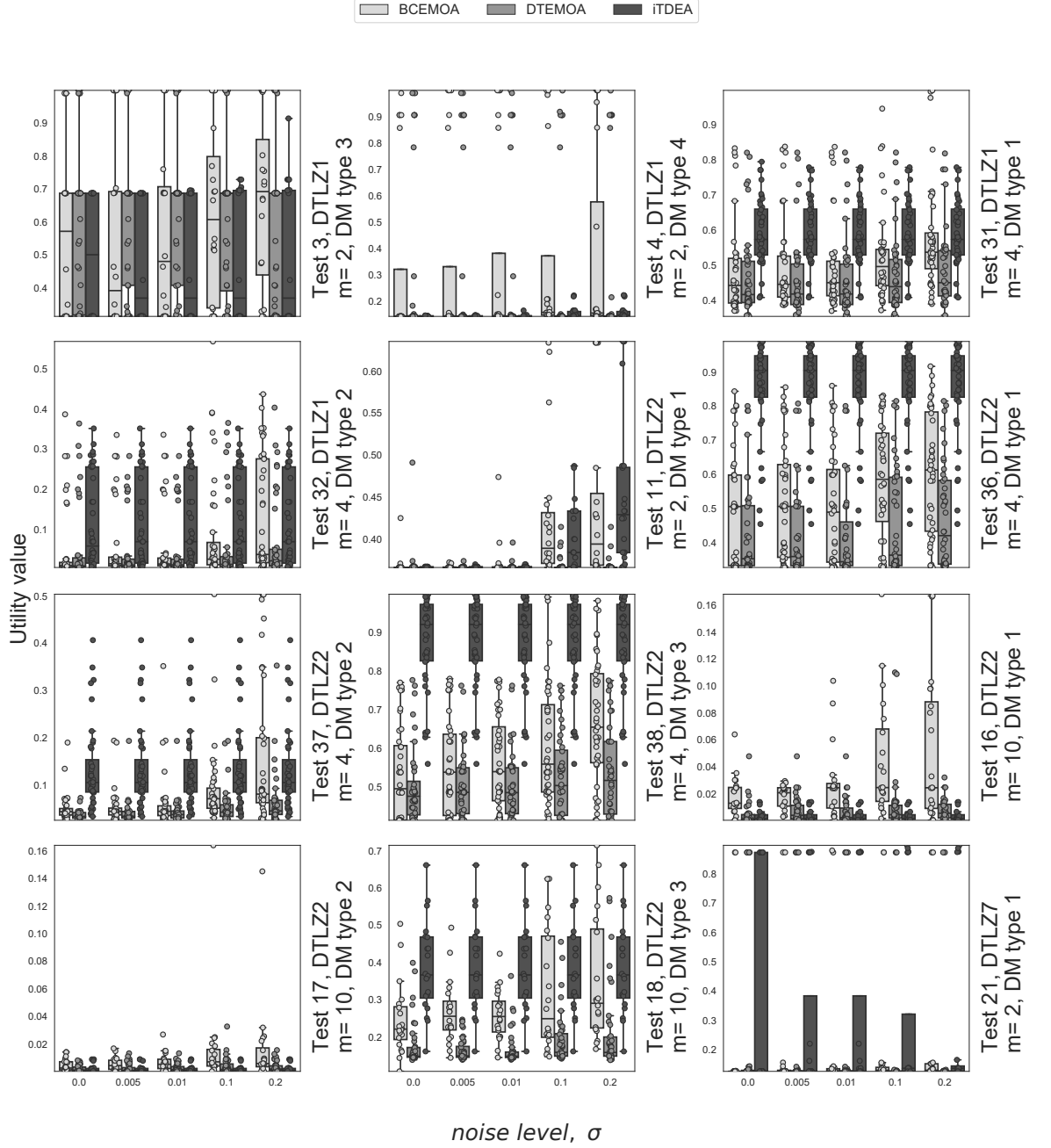


Figure 5.3. Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA with simulation of inconsistencies in DM's decisions and Stewart UF.

ence is not significant (p -value = 0.739). With the simulation of inconsistencies, the performance of BCEMOA and iTDEA is highly affected, while only minor changes are observed for DTEMOA. Thus, DTEMOA outperforms other algorithms in all cases (p -value $< 10^{-5}$).

To save space, we have not presented results for each test but reported on the aggregated results and limited our comments to the main aspects of the performance. The full details of experiments on each test are provided in Appendix C through box plots and tables containing statistical data, including standard deviation and p -values of Wilcoxon's tests of significance. However, here we will discuss the most important ones.

As shown earlier in experiments under ideal conditions, DTEMOA outperforms BCE-

Table 5.6. Comparing the performance of algorithms over various number of objective functions (m). The utility values are averaged over all tests and are rounded to 3 decimal points. Algorithms are compared pairwise and the p -value for Wilcoxon test is reported. The p -values are adjusted using Holm's method for multiple comparisons.

	BCEMOA	DTEMOA	iTDEA	DTEMOA vs. BCEMOA	DTEMOA vs. iTDEA	BCEMOA vs. iTDEA
m	mean utility value			p -value		
Under ideal conditions						
2	0.270	0.270	0.256	1.000	1.000	1.000
4	0.215	0.193	0.319	0.002	0.000	0.000
10	0.122	0.077	0.097	0.000	0.652	0.000
With simulation of inconsistencies in DM's decisions						
2	0.292	0.266	0.268	0.000	0.012	0.055
4	0.244	0.200	0.328	0.000	0.000	0.000
10	0.116	0.081	0.103	0.000	0.158	0.000

Table 5.7. Comparing the performance of algorithms over different decision-making types. The utility values are averaged over all tests. The values are rounded to 3 decimal points. Algorithms are compared pairwise and the p -value for Wilcoxon test are adjusted using Holm's method for multiple comparisons and illustrated in the table.

	BCEMOA	DTEMOA	iTDEA	DTEMOA vs. BCEMOA	DTEMOA vs. iTDEA	BCEMOA vs. iTDEA
DM Type	mean utility value			<i>p</i> -value		
Under ideal conditions						
Tchebychef	0.103	0.102	0.126	0.007	0.000	0.000
1	0.284	0.214	0.371	0.000	0.000	0.001
2	0.065	0.053	0.073	0.000	0.000	0.115
3	0.424	0.377	0.501	0.000	0.000	0.000
4	0.150	0.154	0.168	0.739	0.000	0.000
With simulation of inconsistencies in DM's decisions						
Tchebychef	0.129	0.107	0.141	0.000	0.000	0.000
1	0.330	0.215	0.404	0.000	0.000	0.000
2	0.082	0.054	0.089	0.000	0.000	0.148
3	0.459	0.380	0.542	0.000	0.000	0.000
4	0.161	0.156	0.168	0.021	0.000	0.000

MOA and iTDEA in all tests except three instances which included DTLZ2 with $m = 2$, and DTLZ7 with $m = 2$ where BCEMOA performs slightly better than DTEMOA. All of these exceptions, as illustrated in Figure 5.4 are simulations of DM type 3 which indicates DTEMOA's performance declines when dealing with this DM type.

DM types 1 and 3 seem to be the most difficult for DTEMOA. The same is true to a lesser degree for BCEMOA and iTDEA. The MPS is found by all algorithms over problems DTLZ1 and DTLZ2 with $m = 2$ and DTLZ2 with $m = 10$ with DM type 1 and over problems DTLZ1($m = 2$) and DTLZ2 ($m = 2$ and $m = 1$) and DTLZ7 ($m = 2$ and $m = 4$) all with

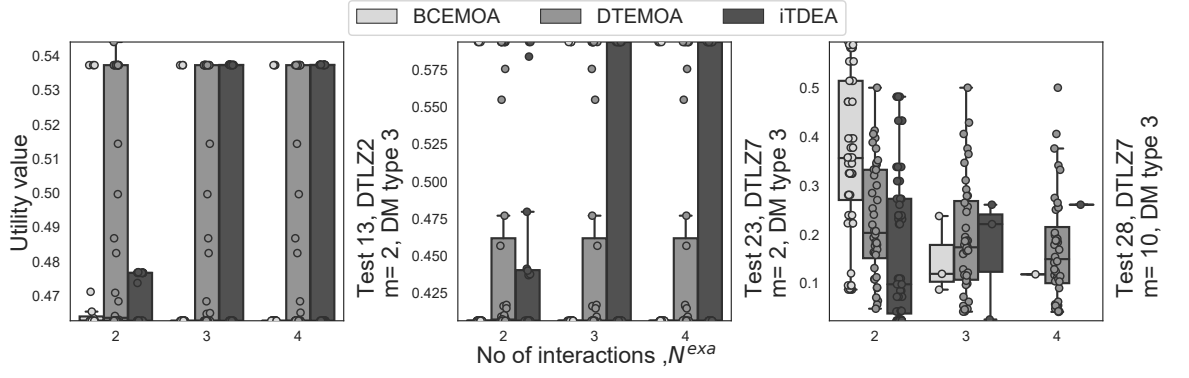


Figure 5.4. The worst performances of DTEMOA in comparison with BCEMOA and iDEA under ideal conditions with Stewart UF.

DM type 2. DTEMOA also finds MPS in tests DTLZ1($m = 2$, DM type 4), DTLZ2($m = 10$, DM type 1), and DTLZ7($m = 4$, DM type 1).

The worst performance of both algorithms is over problem DTLZ1 ($m = 2$) and DM type=3. Some of the best and worst performances of the algorithms are illustrated in Figure 5.5.

Finally, the location of the returned points for DTLZ7 with 2 objectives where DM Type 2 is used to simulate DM preferences is provided in Figure 5.6. Each algorithm is run 20 times with different random seeds and for each run, the final solution is depicted in the figure. The points returned by the DTEMOA are consistently closer to MPS. In contrast, the solutions returned by other algorithms are scattered away. It can be noted that all algorithms find points almost located on the PF but with less desirable utility.

5.6 Conclusions

There have been many improvements in the field of interactive methods, and yet it seems there is room for more. Recent research on benchmarking of these methods has revealed that they may not perform as expected when confronted with non-ideal conditions that were not anticipated in their development (Afsar et al., 2021; Shavarani et al., 2021). At present, the field has little idea of how to explain or predict which components of iEMOA methods are most vulnerable to non-idealities or what conditions affect their performance the most. However, we believe the core of any interactive method is the preference model and the accuracy of these models in reflecting the DM's preferences is crucial in the optimization process. With these observations, this study proposed an innovative preference model and learning approach using decision trees to predict the result of pairwise comparisons from the DM's perspective to manage the convergence direction of the population. The method's performance is found to be stable in many-objective problems because more objectives translate to more attributes that can participate in classification. Further, when the number of objectives

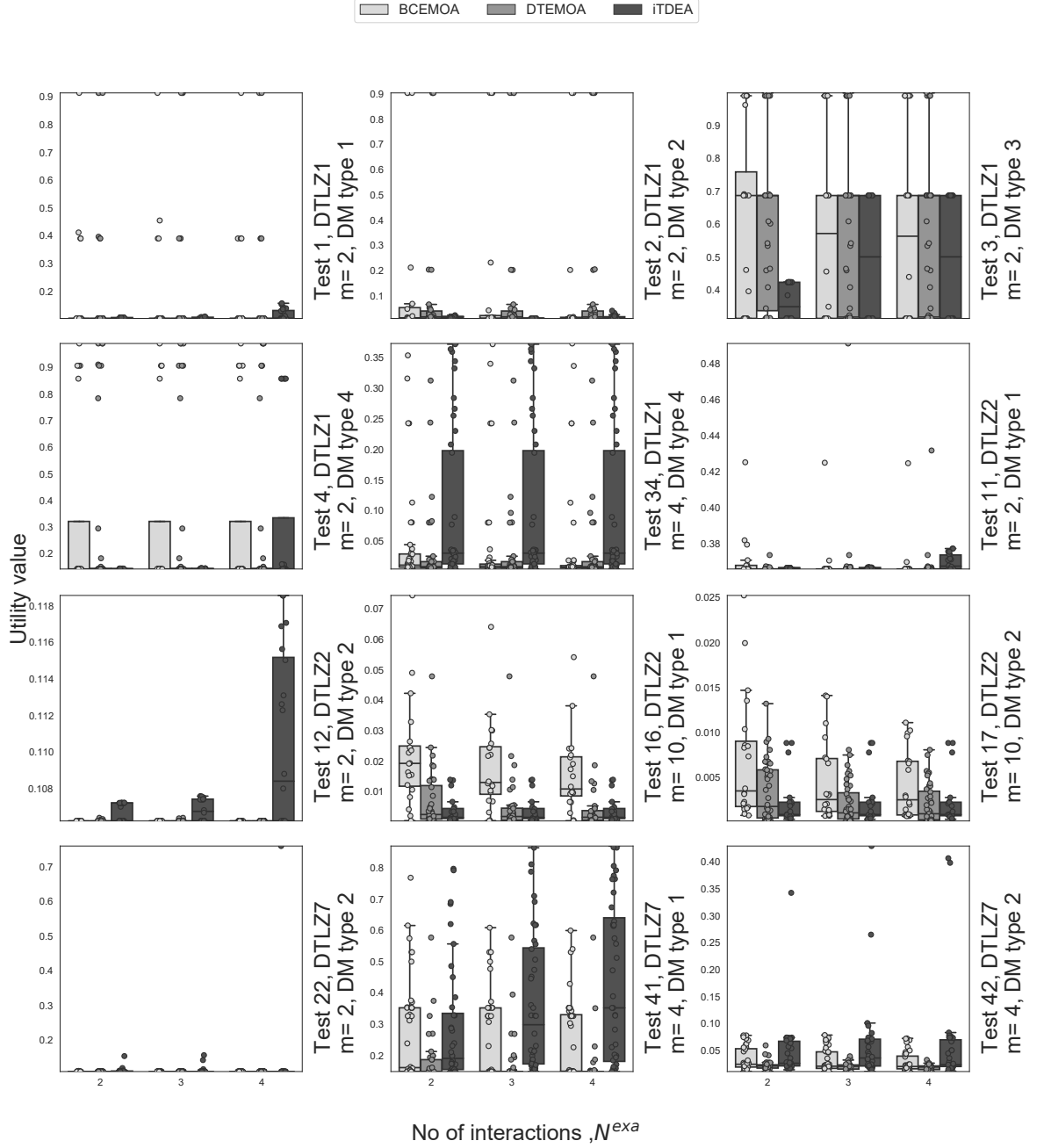


Figure 5.5. Some of the best and worst performances of the algorithms under ideal conditions with Stewart UF.

increases, DTs can naturally identify the most important ones. Our method is also found to be more robust to biases as we make holistic comparisons to cancel out any estimation errors in single predictions. In contrast, other learning methods such as SVM rely on kernel performance, and complications such as higher dimensions, non-linearity, and biases make kernel selection an intensive task and generally less accurate. The performance of DTEMOA is also more efficient as it reaches better results in terms of the utility value of the final solution in less interactions.

Another critical advantage of DTs in this context is their intuitive interpretation and ease of visualization, both being essential elements in gaining the DM's trust in the process and the results. We used the suggested experimental design of (Shavarani et al., 2021) to evaluate

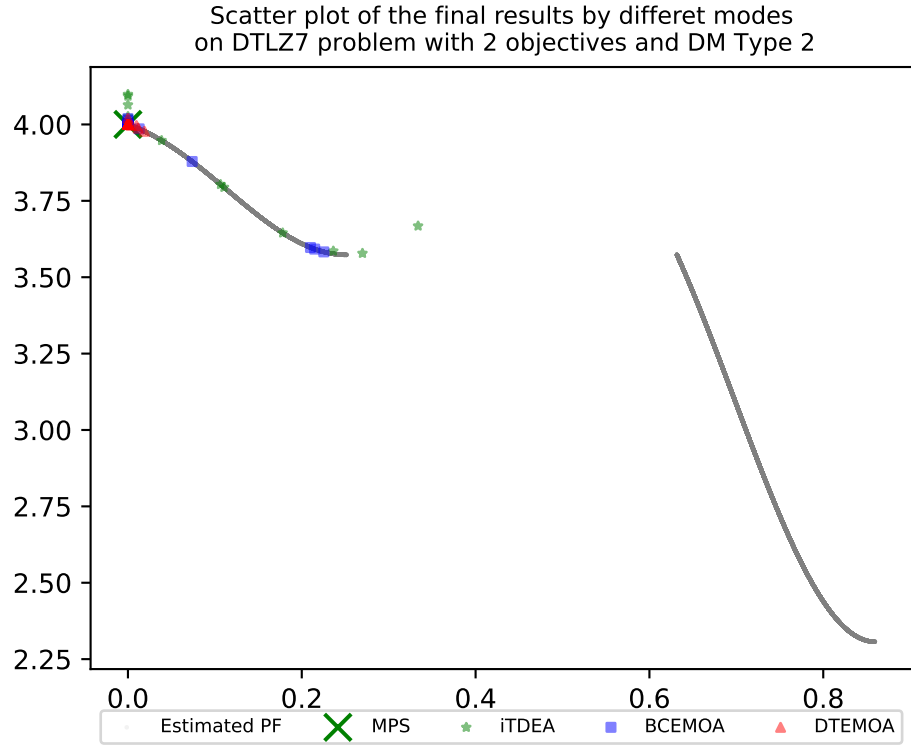


Figure 5.6. Location of the returned solutions with different algorithms on DTLZ7 problem with $m = 2$ objectives and DM Type 2. The horizontal axis indicates the first objective (f_1) and the vertical axis indicates the second objective (f_2). The results returned by DTEMOA (indicated in red) are close to MPS(depicted in green “X”).

the performance of the algorithm and examine the efficiency of the method. Further research may extend this study by investigating the application of weighted decision trees to emphasize the most recently elicited information. Another interesting research direction is to explore the use of decision tree regressors instead of classifiers in the preference model.

Reproducibility. We make source code and data for reproducing our results publicly available as supplementary materials (Shavarani et al., 2022) to motivate further research in this direction.

Chapter 6

Conclusions & Future Study

The objective of this thesis was to develop tools and measures for benchmarking and improving iEMOAs (Interactive Evolutionary Multi-Objective Optimization Algorithms). Previous researchers in the field have emphasized the need for proper measures and tools to benchmark iEMOAs, as the lack thereof hinders our understanding of how these algorithms perform under real-life conditions and delays progress in the field (López-Ibáñez & Knowles, 2015; Afsar et al., 2021). In response, this thesis proposed improvements to the design of experiments for iEMOAs. Specifically, it built upon an existing MDM and incorporated a realistic utility function (UF) while also extending the list of biases and non-idealities that the MDM simulates.

Such a realistic simulation allowed us to evaluate the performance of well-known iEMOAs and illustrate how their performance can be biased under realistic conditions. The results uncovered some weaknesses in the iEMOAs we studied, which inspired us to propose improvements to the iEMOAs. We have organized our conclusions into two sections. The first section briefly revisits the research’s main findings, limitations, and implications. The second section proposes the directions for future research, which mainly targets the limitations of our research and is inspired by the lessons we have learned from such extensive analysis of iEMOAs.

6.1 Findings

In the first step of this research, we investigated key problems associated with measuring and improving iEMOAs, identified the gaps in the literature, and addressed some issues in this field.

Building on a research by (López-Ibáñez & Knowles, 2015), we proposed a machine DM utilising the realistic Stewart UF that can mimic different decision-making behaviors. The results of the experiments presented in Chapter 3 indicate how different simulated behaviors

can affect the performance of the iEMOA. We also noticed that for some parameter combinations of the Stewart UF, the utility value of the non-dominated solutions fall into a narrow interval. As a result, any point on the PF returned by the algorithm will have almost the same utility, making it challenging to discriminate any significant observations. A mathematical model was proposed in Chapter 3 that can be used to simulate a specific behavior and impose desired characteristics, such as the place of the MPS on the PF, and preserving the diversity of the utility values of the non-dominated solutions.

We extended the list of non-idealities and biases discussed in the original study to take another step toward a more realistic and reliable MDM. The results indicated that the performance of the algorithms studied is not robust towards non-idealities, and there are considerable deviations from the MPS when such non-idealities exist. These results confirm the need for such extensive experiments in the field, performing statistical analyses and benchmarking of interactive methods under realistic conditions, scrutinizing different elements of interactive methods (particularly preference learning and modeling), and exploring ideas for improvement of iEMOAs. We used the MDM to experiment with two iEMOAs in Chapter 3. The results indicated how experimenting under realistic conditions would affect the performance of iEMOAs and possibly reveal their weaknesses, which will open the way for more improvements in the field. We also observed how the performance of iEMOAs can be affected not only by the non-idealities but also by the number of objectives which can significantly deteriorate the effectiveness of these methods. One of the main goals of iEMOAs is to address problems with many-objective problems; thus, such a decline in performance is not expected with the increased number of objectives. In some cases, even the design of the algorithm will not allow its application to many-objective problems.

For instance, as reviewed in Chapter 2.2.2 some studies estimate the UF by solving a linear programming model or running an EMOA which becomes time-consuming and complex with the increase in the number of objectives. During our experiment, we noticed a similar problem with iTDEA when it came to selecting algorithm parameters, specifically the parameter responsible for controlling the density of solutions in objective space (called territory size in iTDEA). A small value for this parameter would overpopulate the population and make the algorithm computationally expensive. On the other hand, large values will result in poor diversity and exploration. Another problem with similar methods is the need for normalization and scaling objectives, a process which is not straightforward and susceptible to the parameters used for normalization. Such scaling is predominantly done by using ideal and nadir points, the estimation of which is not a trivial task in real-life problems (Szczepański & Wierzbicki, 2003; Afsar et al., 2021).

We then elaborated and provided formal definitions for two non-idealities; irrelevant and

hidden objectives. Although these objectives were discussed in practical experiments in the literature, no formal definitions were available. We proposed an approach to simulate and integrate hidden and irrelevant objectives into any existing multi-objective benchmark problem in Chapter 4. To make iEMOAs robust toward such non-idealities, we designed an efficient method based on uni-variate feature selection and recursive feature elimination to automatically identify hidden and irrelevant objectives. The method can change the set of active objective functions during the optimization process to optimize the set of objective functions that align with DM's preferences. We also discussed applying the proposed feature selection methods as an objective reduction technique to reduce those objectives that are not (or less) important to the DM and increase computational efficiency. The results of our experiments indicated that objective evaluations could be dropped by 80%. This reduction in objective evaluations is significant in agile real-world problems where an evaluation of complex objectives is time-consuming. Detecting relevant objectives would also benefit preference learning as the estimated model is not biased or complicated by irrelevant objectives. The study illustrated that the valuable DM's preferences elicited in iEMOAs for directing the search could also be exploited in other ways to address the problems associated with many-objective problems.

Considering how the discrepancy between the set of objective functions and DM's preferences degrades the learned model, a novel iEMOA was proposed in Chapter 5, which is based on decision trees that naturally deploy feature selection. The learning scheme in the proposed iEMOA is differentiated by several important characteristics. The DT is trained on the trade-offs of some small set of ranked solutions and is later used to predict the preferred solution in pairwise comparisons based on their trade-offs. The proposed preference learning scheme was an effort to avoid problems with other preference learning methods; (i) we do not assume any specific shape for the DM's preference model. (ii) we do not aim for learning and estimating a complete (or partial) ranking of the solutions through an ensemble of estimators, which is prone to biases, or by estimating a UF that can not adequately capture the complicated preferences of the DM. (iii) Owing to their mechanism and use of information gain, DTs can detect irrelevant objectives and make predictions based on the important ones. (iv) DTs can be easily depicted and understood by the DM, and the DM can revisit his decisions if required. The algorithm is thus explainable, which is proved to be an essential feature in facilitating decision-making and gaining the DM's trust in the final results. (vi) Like any other learning algorithm, the accuracy of DTs is never 100%, and they are prone to errors. To minimize the effect of such errors, we make comprehensive comparisons and compare each solution with all of the population to get a better estimate of the solution's quality compared to the current population. To evaluate our DT-based preference learning, we compared our proposed preference learning scheme with one based on support vector machines used in

(Battiti & Passerini, 2010), and the results indicated the superiority of our scheme in terms of accuracy and robustness to non-idealities. This experiment showcases the experimental settings that the researchers can employ to investigate an isolated part of their method rather than evaluating the algorithms' performance based on the final results, which is not so much informative and does not explain why a particular result is observed.

The contributions and findings of this research can be summarized as follows. We have improved the MDM framework proposed in (López-Ibáñez & Knowles, 2015) with the Stewart UF, which is used for the first time in this field. We have introduced and provided formal definitions for hidden objectives, irrelevant objectives and simulated several other non-idealities. The usage of the MDM is showcased through the research on experimenting on two existing algorithms and benchmarking the performance of a novel algorithm by comparing it with other well-known algorithms.

We discussed how the algorithms can be improved with the detection and handling of hidden and irrelevant objectives, which improves the performance in terms of the utility value of the final results as well as the computational efficiency. We also introduced a new objective reduction approach that considers the DM's preferences rather than the problem's structure. The results indicated that the number of objective evaluations is reduced up to 80% while improving the quality of the returned solutions.

Finally, a preference learning technique was proposed based on decision trees that can handle irrelevant objectives and inconsistencies in the DM's decisions and can adapt to complex preference models of the DM. This learning technique was used in the development of a novel iEMOA (DTEMOA) introduced in Chapter 5. Using the MDM framework implemented in Chapter 3, we conducted experiments to compare the performance of DTEMOA with that of other algorithms. The results demonstrate the algorithm's superiority over two other iEMOAs, as well as its robustness in the face of noise and non-idealities.

6.2 Future Study

The advancements to the MDM discussed here are just another step towards improving such frameworks. Many other non-idealities in the literature can be integrated into the MDM to make the simulation of decision-making behaviors more realistic and the results more reliable. Such non-idealities include but are not limited to fatigue (Zujevs & Eiduks, 2008), hesitation (Huber et al., 2015), lack of complete information (Lerch & Harter, 2001), cognitive bottleneck (Tversky & Kahneman, 1974; Stewart, 2005) and anchoring (Buchanan & Corner, 1997). We share our implementation of the MDM and welcome the contributions of other scientists in the field.

It is also desirable to study the improvements of an existing iEMOA by varying different elements and observing the effects on the results. These elements might include selecting solutions for DM's evaluations, preference modeling, preference elicitation styles, and stopping criteria.

Our experiments were limited to several well-known benchmark problems from the literature. These problems provide the opportunity to test the algorithms under specific difficulties. However, they also impose some undesired characteristics. For instance, the objectives are highly correlated in some cases, and optimizing any of the correlated objectives will result in optimized values of other objectives. Another issue we encountered was that the projection of some of the problems into lower dimensions would impose undesired characteristics (Lygoe et al., 2010), such as the scenario with DTLZ problems where the PF collapses to one point or a few points. Another problem that was specifically troublesome when experimenting with objective reduction methods in Chapter 4, was that the problems seemed too easy for the algorithms. It is desirable to use representative of real-life applications with different computational challenges in experimenting on iEMOAs.

The feature selection used for updating the set of active objectives in Chapter 4 was based on uni-variate feature selection and recursive feature elimination. These methods do not consider the correlation among objectives. On the other hand, other algorithms that do so incur higher computational time, which is not desirable during interaction with the DM. Future research can build on our proposed method to find a solution with a proper trade-off between computational cost and accuracy in detecting any correlations among objectives. We will also pursue applying feature extraction methods instead of feature selection techniques in our future study. It was also observed that the detection method successfully identifies the relevant objectives. However, it might happen that after optimizing the relevant ones, the algorithm starts exploring the feasible space by deactivating the optimized relevant objectives and optimizing other irrelevant objectives. Proper mechanisms can be incorporated to stop optimization when the relevant objectives are optimized to increase computational efficiency.

We have introduced objective reduction techniques that target irrelevant objectives. It might be beneficial to investigate the combination of reduction techniques that target irrelevant and redundant objectives, particularly for many-objective problems.

In our study, we limited our experiments to two algorithms that had publicly shared their source code. On the other hand, implementing other algorithms that do not provide the source code will be complicated and time-consuming, even if the full details were provided in the original studies, which is not the case in most of them. Thus, we encourage the research community to share their source code to facilitate experimenting on iEMOAs.

We have used DTs in Chapter 5 to model the DM's preferences. The robustness of the DTs in learning preferences motivates research toward using random forests or boosted DTs, which tend to be more accurate and may result in better performance. Also, one may explore the effect of using a temporally weighted DT, where recent preference information is weighted more to enable the algorithm to cope with learning and the change in DM's preferences. It is also possible to use the value acquired from DT regressors to rank the solutions in the population directly. This possibility should be explored in future research. The information gain that is used for deciding on the root node and its subsequent nodes, can also be used as a feature selection technique to decide on the objectives that are relevant and should be optimized by the iEMOA.

To motivate further contributions and facilitate improvements in the field, We make all the codes used in this study publicly available at <https://github.com/ShavaraniMahdi/MachineDM>.

References

- Afsar, B., Miettinen, K., & Ruiz, F. (2021). Assessing the Performance of Interactive Multiobjective Optimization Methods: A Survey. *ACM Computing Surveys*, 54(4). doi:[10.1145/3448301](https://doi.org/10.1145/3448301)
- Afsar, B., Ruiz, A. B., & Miettinen, K. (2021). Comparing interactive evolutionary multiobjective optimization methods with an artificial decision maker. *Complex & Intelligent Systems*. doi:[10.1007/s40747-021-00586-5](https://doi.org/10.1007/s40747-021-00586-5)
- Agrell, P. J. (1997). On redundancy in multi criteria decision making. *European Journal of Operational Research*, 98(3), 571–586. doi:[10.1016/0377-2217\(95\)00340-1](https://doi.org/10.1016/0377-2217(95)00340-1)
- Aguirre, H. E., & Tanaka, K. (2007). Working principles, behavior, and performance of MOEAs on MNK-landscapes. *European Journal of Operational Research*, 181(3), 1670–1690. doi:[10.1016/j.ejor.2006.08.004](https://doi.org/10.1016/j.ejor.2006.08.004)
- Allen, J., Moussa, A., & Liu, X. (2019). Human-in-the-Loop Learning of Qualitative Preference Models. In R. Barták & K. W. Brawner (Eds.), *Proceedings of the Thirty-Second International Florida Artificial Intelligence Research Society Conference, Sarasota, Florida, USA, May 19-22 2019* (pp. 108–111). AAAI Press.
- Allmendinger, R. (2012). *Tuning Evolutionary Search for Closed-Loop Optimization* (Doctoral dissertation, The University of Manchester, UK).
- Allmendinger, R., Jaszekiewicz, A., Liefooghe, A., & Tammer, C. (2022). What if we increase the number of objectives? Theoretical and empirical implications for many-objective combinatorial optimization. *Computers & Operations Research*, 145, 105857. doi:[10.1016/j.cor.2022.105857](https://doi.org/10.1016/j.cor.2022.105857)
- Allmendinger, R., & Knowles, J. (2021). Heterogeneous objectives: State-of-the-art and future research. *arXiv preprint arXiv:2103.15546*.
- Allmendinger, R., & Knowles, J. D. (2010). Evolutionary Optimization on Problems Subject to Changes of Variables. In R. Schaefer, C. Cotta, J. Kołodziej, & G. Rudolph (Eds.), *Parallel Problem Solving from Nature, PPSN XI* (Vol. 6238, pp. 151–160). Heidelberg: Springer.

- Barba-González, C., Ojalehto, V., García-Nieto, J., Nebro, A. J., Miettinen, K., & Aldana-Montes, J. F. (2018). Artificial Decision Maker Driven by PSO: An Approach for Testing Reference Point Based Interactive Methods. In A. Auger, C. M. Fonseca, N. Lourenço, P. Machado, L. Paquete, & D. Whitley (Eds.), *Parallel Problem Solving from Nature – PPSN XV* (pp. 274–285). Cham: Springer International Publishing.
- Battiti, R., & Campigotto, P. (2010). Reactive search optimization: Learning while optimizing. An experiment in interactive multi-objective optimization. In M. Caserta & S. Voß (Eds.), *Proceedings of MIC 2009, the 8th Metaheuristics International Conference*, Hamburg, Germany: University of Hamburg.
- Battiti, R., & Passerini, A. (2010). Brain-Computer Evolutionary Multiobjective Optimization: A Genetic Algorithm Adapting to the Decision Maker. *IEEE Transactions on Evolutionary Computation*, 14(5), 671–687. doi:[10.1109/TEVC.2010.2058118](https://doi.org/10.1109/TEVC.2010.2058118)
- Belton, V., Branke, J., Eskelinen, P., Greco, S., Molina, J., Ruiz, F., & Słowiński, R. (2008). Interactive Multiobjective Optimization from a Learning Perspective. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Vol. 5252, pp. 405–433). doi:[10.1007/978-3-540-88908-3_15](https://doi.org/10.1007/978-3-540-88908-3_15)
- Belton, V., & Elder, M. D. (1994). Decision support systems: Learning from visual interactive modelling. *Decision Support Systems*, 12(4), 355–364. doi:[10.1016/0167-9236\(94\)90052-3](https://doi.org/10.1016/0167-9236(94)90052-3)
- Ben Said, L., Bechikh, S., & Ghedira, K. (2010). The r-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making. *IEEE Transactions on Evolutionary Computation*, 14(5), 801–818.
- Benabbou, N., Leroy, C., & Lust, T. (2020). An Interactive Regret-Based Genetic Algorithm for Solving Multi-Objective Combinatorial Optimization Problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03), 2335–2342. doi:[10.1609/aaai.v34i03.5612](https://doi.org/10.1609/aaai.v34i03.5612)
- Benayoun, R., De Montgolfier, J., Tergny, J., & Laritchev, O. (1971). Linear programming with multiple objective functions: Step method (STEM). *Mathematical programming*, 1(1), 366–375.

- Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L., & Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, 13(5), 1075–1082. doi:[10.1109/TEVC.2009.2015575](https://doi.org/10.1109/TEVC.2009.2015575)
- Bezerra, L. C. T., López-Ibáñez, M., & Stützle, T. (2018). A Large-Scale Experimental Evaluation of High-Performing Multi- and Many-Objective Evolutionary Algorithms. *Evolutionary Computation*, 26(4), 621–656. doi:[10.1162/evco_a_00217](https://doi.org/10.1162/evco_a_00217)
- Biscani, F., Izzo, D., & Yam, C. H. (2010a). A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation. *Arxiv preprint arXiv:1004.3824*. Retrieved from <http://arxiv.org/abs/1004.3824>
- Biscani, F., Izzo, D., & Yam, C. H. (2010b). A Global Optimisation Toolbox for Massively Parallel Engineering Optimisation. In *Astrodynamics Tools and Techniques (ICATT 2010), 4th International Conference on*. Retrieved from <http://arxiv.org/abs/1004.3824>
- Bishop, C. (2006a). *Pattern Recognition and Machine Learning*. Springer. Retrieved from <https://books.google.co.uk/books?id=kTNoQgAACAAJ>
- Bishop, C. M. (2006b). *Pattern recognition and machine learning*. Springer.
- Branke, J., Corrente, S., Greco, S., Słowiński, R., & Zielniewicz, P. (2016). Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research*, 250(3), 884–901. doi:[10.1016/j.ejor.2015.10.027](https://doi.org/10.1016/j.ejor.2015.10.027)
- Branke, J., & Deb, K. (2005). Integrating User Preferences into Evolutionary Multi-Objective Optimization. In Y. Jin (Ed.), *Knowledge Incorporation in Evolutionary Computation* (pp. 461–477). doi:[10.1007/978-3-540-44511-1_21](https://doi.org/10.1007/978-3-540-44511-1_21)
- Branke, J., Deb, K., Miettinen, K., & Słowiński, R. (Eds.). (2008). *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Heidelberg: Springer.
- Branke, J., Greco, S., Słowiński, R., & Zielniewicz, P. (2010). Interactive evolutionary multi-objective optimization driven by robust ordinal regression. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 58(3), 347–358. doi:[10.2478/v10175-010-0033-3](https://doi.org/10.2478/v10175-010-0033-3)
- Branke, J., Greco, S., Słowiński, R., & Zielniewicz, P. (2015). Learning Value Functions in Interactive Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 19(1), 88–102.
- Branke, J., Kaussler, T., & Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32, 499–507.

- Breiman, L., Friedman, J., Stone, C. J., & Olshen, R. A. (1984). *Classification and regression trees*. CRC press.
- Brockhoff, D., Hamadi, Y., & Kaci, S. (2014). Using Comparative Preference Statements in Hypervolume-Based Interactive Multiobjective Optimization. In P. M. Pardalos, M. G. Resende, C. Vogiatzis, & J. L. Walteros (Eds.), *Learning and Intelligent Optimization* (pp. 121–136). Cham: Springer International Publishing.
- Brockhoff, D., & Zitzler, E. (2006). Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In T. P. Runarsson, H.-G. Beyer, E. K. Burke, J.-J. Merelo, D. Whitley, & X. Yao (Eds.), *Parallel Problem Solving from Nature – PPSN IX* (Vol. 4193, pp. 533–542). Heidelberg: Springer.
- Brockhoff, D., & Zitzler, E. (2007a). Dimensionality Reduction in Multiobjective Optimization: The Minimum Objective Subset Problem. In K.-H. Waldmann & U. M. Stocker (Eds.), *Operations Research Proceedings 2006* (pp. 423–429). doi:[10.1007/978-3-540-69995-8_68](https://doi.org/10.1007/978-3-540-69995-8_68)
- Brockhoff, D., & Zitzler, E. (2007b). Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, 25-28 September 2007, Singapore* (pp. 2086–2093). doi:[10.1109/CEC.2007.4424730](https://doi.org/10.1109/CEC.2007.4424730)
- Brockhoff, D., & Zitzler, E. (2009). Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications. *Evolutionary Computation*, 17(2), 135–166. doi:[10.1162/evco.2009.17.2.135](https://doi.org/10.1162/evco.2009.17.2.135)
- Buchanan, J. T., & Corner, J. (1997). The effects of anchoring in interactive MCDM solution methods. *Computers & operations research*, 24(10), 907–918.
- Buchanan, J. T. (1994). An experimental evaluation of interactive MCDM methods and the decision making process. *Journal of the Operational Research Society*, 45(9), 1050–1059.
- Campigotto, P., & Passerini, A. (2010). Adapting to a realistic decision maker: experiments towards a reactive multi-objective optimizer. In C. Blum & R. Battiti (Eds.), *Learning and Intelligent Optimization, 4th International Conference, LION 4* (Vol. 6073, pp. 338–341). doi:[10.1007/978-3-642-13800-3](https://doi.org/10.1007/978-3-642-13800-3)
- Chen, L., Xin, B., Chen, J., & Li, J. (2017). A virtual-decision-maker library considering personalities and dynamically changing preference structures for interactive multiobjective

- optimization. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 636–641). doi:[10.1109/CEC.2017.7969370](https://doi.org/10.1109/CEC.2017.7969370)
- Cheng, W., Hühn, J., & Hüllermeier, E. (2009). Decision tree and instance-based learning for label ranking. In *Proceedings of the 26th annual international conference on machine learning* (pp. 161–168).
- Chrysostomou, K., Chen, S. Y., & Liu, X. (2011). Identifying user preferences with Wrapper-based Decision Trees. *Expert Systems with Applications*, 38(4), 3294–3303. doi:[10.1016/j.eswa.2010.08.115](https://doi.org/10.1016/j.eswa.2010.08.115)
- Chu, J., Zhang, C., Fu, G., Li, Y., & Zhou, H. (2015). Improving multi-objective reservoir operation optimization with sensitivity-informed dimension reduction. *Hydrology and Earth System Sciences*, 19(8), 3557–3570. doi:[10.5194/hess-19-3557-2015](https://doi.org/10.5194/hess-19-3557-2015)
- Chugh, T., Sindhya, K., Hakanen, J., & Miettinen, K. (2015). An Interactive Simple Indicator-Based Evolutionary Algorithm (I-SIBEA) for Multiobjective Optimization Problems. In A. Gaspar-Cunha, C. Henggeler Antunes, & C. C. Coello (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 277–291). Cham: Springer International Publishing.
- Coello Coello, C. A. (2002). Evolutionary Multi-Objective Optimization: A Critical Review. In *Evolutionary Optimization* (pp. 117–146). doi:[10.1007/0-306-48041-7_5](https://doi.org/10.1007/0-306-48041-7_5)
- Coello Coello, C. A. (2006). Evolutionary multi-objective optimization: a historical view of the field. *IEEE Computational Intelligence Magazine*, 1(1), 28–36.
- Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, NY: Springer.
- Corne, D. W., Jerram, N. R., Knowles, J. D., & Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 3rd annual conference on genetic and evolutionary computation* (pp. 283–290).
- Costa, L., & Oliveira, P. (2010). Biplots in offline multiobjective reduction. In *IEEE Congress on Evolutionary Computation* (pp. 1–8). doi:[10.1109/CEC.2010.5585997](https://doi.org/10.1109/CEC.2010.5585997)
- D’Agostino, R., & Pearson, E. S. (1973). Tests for Departure from Normality. Empirical Results for the Distributions of b_2 and $\sqrt{b_1}$. *Biometrika*, 60(3), 613–622. doi:[10.2307/2335012](https://doi.org/10.2307/2335012)
- Dalip, D. H., Gonçalves, M. A., Cristo, M., & Calado, P. (2013). Exploiting User Feedback to Learn to Rank Answers in Q&a Forums: A Case Study with Stack Overflow. In *Proceed-*

- ings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 543–552). doi:[10.1145/2484028.2484072](https://doi.org/10.1145/2484028.2484072)
- Deb, K., Sinha, A., Korhonen, P. J., & Wallenius, J. (2010a). An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions. *IEEE Transactions on Evolutionary Computation*, 14(5), 723–739. doi:[10.1109/TEVC.2010.2064323](https://doi.org/10.1109/TEVC.2010.2064323)
- Deb, K. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, UK: Wiley.
- Deb, K. (2005a). Multi-objective Optimization. In E. K. Burke & G. Kendall (Eds.), *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques* (pp. 403–449). doi:[10.1007/978-1-4614-6940-7_15](https://doi.org/10.1007/978-1-4614-6940-7_15)
- Deb, K. (2005b). Multi-objective optimization. In E. K. Burke & G. Kendall (Eds.), *Search Methodologies* (pp. 273–316). doi:[10.1007/0-387-28356-0_10](https://doi.org/10.1007/0-387-28356-0_10)
- Deb, K. (2008). Introduction to evolutionary multiobjective optimization. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Vol. 5252, pp. 59–96). doi:[10.1007/978-3-540-88908-3_3](https://doi.org/10.1007/978-3-540-88908-3_3)
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197. doi:[10.1109/4235.996017](https://doi.org/10.1109/4235.996017)
- Deb, K., & Saxena, D. K. (2006). Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)* (pp. 3353–3360).
- Deb, K., Sinha, A., Korhonen, P. J., & Wallenius, J. (2010b). An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions. *IEEE Transactions on Evolutionary Computation*, 14(5), 723–739. doi:[10.1109/TEVC.2010.2064323](https://doi.org/10.1109/TEVC.2010.2064323)
- Deb, K., & Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In M. Cattolico et al. (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2006* (pp. 635–642). doi:[10.1145/1143997.1144112](https://doi.org/10.1145/1143997.1144112)

- Deb, K., Tewari, R., Dixit, M., & Dutta, J. (2007). Finding trade-off solutions close to KKT points using evolutionary multi-objective optimization. In *Proceedings of the 2007 Congress on Evolutionary Computation (CEC 2007)* (pp. 2109–2116). Piscataway, NJ: IEEE Press.
- Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, & R. Goldberg (Eds.), *Evolutionary Multiobjective Optimization* (pp. 105–145). doi:[10.1007/1-84628-137-7_6](https://doi.org/10.1007/1-84628-137-7_6)
- Deb, Kalyanmoy. (2007). Current trends in evolutionary multi-objective optimization. *Int. J. Simul. Multidisci. Des. Optim.*, 1(1), 1–8. doi:[10.1051/ijsmdo:2007001](https://doi.org/10.1051/ijsmdo:2007001)
- Debreu, G. (1987). *Theory of value: an axiomatic analysis of economic equilibrium* (19. Dr.). OCLC: 33936319. New Haven: Yale Univ. Press.
- de Freitas, A. R. R., Fleming, P. J., & Guimarães, F. G. (2015). Aggregation trees for visualization and dimension reduction in many-objective optimization. *Information Sciences*, 298, 288–314.
- Dekel, O., Singer, Y., & Manning, C. D. (2003). Log-Linear Models for Label Ranking. In S. Thrun, L. Saul, & B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems* (Vol. 16), MIT Press. Retrieved from <https://proceedings.neurips.cc/paper/2003/file/217c0e01c1828e7279051f1b6675745d-Paper.pdf>
- DeShazo, J., & Fermo, G. (2002). Designing Choice Sets for Stated Preference Methods: The Effects of Complexity on Choice Consistency. *Journal of Environmental Economics and Management*, 44(1), 123–143. doi:[10.1006/jeem.2001.1199](https://doi.org/10.1006/jeem.2001.1199)
- Destrero, A., Mosci, S., Mol, C., Verri, A., & Odone, F. (2009). Feature selection for high-dimensional data. *Computational Management Science*, 6, 25–40. doi:[10.1007/s10287-008-0070-7](https://doi.org/10.1007/s10287-008-0070-7)
- Farrugia, V. E., Martínez, H. P., & Yannakakis, G. N. (2015). The Preference Learning Toolbox.
- Fieldsend, J. E., & Everson, R. M. (2013). Visualising high-dimensional Pareto relationships in two-dimensional scatterplots. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, & J. Shaw (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2013* (Vol. 7811, pp. 558–572). doi:[10.1007/978-3-642-37140-0_42](https://doi.org/10.1007/978-3-642-37140-0_42)

- Figueira, J., Liefooghe, A., Talbi, E.-G., & Wierzbicki, A. (2010). A parallel multiple reference point approach for multi-objective optimization. *European Journal of Operational Research*, 205(2), 390–400. doi:[10.1016/j.ejor.2009.12.027](https://doi.org/10.1016/j.ejor.2009.12.027)
- Fleming, P. J., Purshouse, R. C., & Lygoe, R. J. (2005). Many-objective optimization: An engineering design perspective. In C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2005* (Vol. 3410, pp. 14–32). Heidelberg: Springer.
- Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In S. Forrest (Ed.), *ICGA* (pp. 416–423). Morgan Kaufmann Publishers.
- Fonseca, C. M., & Fleming, P. J. (1998). Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms (II): Application Example. *IEEE Transactions on Systems, Man, and Cybernetics – Part A*, 28(1), 38–44. doi:[10.1109/3468.650320](https://doi.org/10.1109/3468.650320)
- Fowler, J. W., Gel, E. S., Köksalan, M. M., Korhonen, P., Marquis, J. L., & Wallenius, J. (2010a). Interactive evolutionary multi-objective optimization for quasi-concave preference functions. *European Journal of Operational Research*, 206(2), 417–425.
- Fowler, J. W., Gel, E. S., Köksalan, M., Korhonen, P., Marquis, J. L., & Wallenius, J. (2010b). Interactive evolutionary multi-objective optimization for quasi-concave preference functions. *European Journal of Operational Research*, 206(2), 417–425. doi:[10.1016/j.ejor.2010.02.027](https://doi.org/10.1016/j.ejor.2010.02.027)
- Fox, S., Karnawat, K., Mydland, M., Dumais, S., & White, T. (2005). Evaluating Implicit Measures to Improve Web Search. *ACM Trans. Inf. Syst.*, 23(2), 147–168. doi:[10.1145/1059981.1059982](https://doi.org/10.1145/1059981.1059982)
- Freitas, A. R. R., Fleming, P. J., & Guimarães, F. G. (2013). A Non-parametric Harmony-Based Objective Reduction Method for Many-Objective Optimization. In *IEEE International Conference on Systems, Man, and Cybernetics, SMC 2013, Manchester, United Kingdom, October 13-16, 2013* (pp. 651–656). doi:[10.1109/SMC.2013.116](https://doi.org/10.1109/SMC.2013.116)
- French, S. (1984). Interactive Multi-Objective Programming: Its Aims, Applications and Demands. *Journal of the Operational Research Society*, 35(9), 827–834. doi:[10.1057/jors.1984.165](https://doi.org/10.1057/jors.1984.165)

- Fürnkranz, J., & Hüllermeier, E. (2003). Pairwise Preference Learning and Ranking. In N. Lavrač, D. Gamberger, H. Blockeel, & L. Todorovski (Eds.), *Machine Learning: ECML 2003* (pp. 145–156). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Gal, T., & Leberling, H. (1977). Redundant objective functions in linear vector maximum problems and their determination. *European Journal of Operational Research*, 1(3), 176–184. doi:[10.1016/0377-2217\(77\)90025-X](https://doi.org/10.1016/0377-2217(77)90025-X)
- Geiger, M. J., & Wenger, W. (2007). On the Interactive Resolution of Multi-objective Vehicle Routing Problems. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 687–699). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Giagkiozis, I., Purshouse, R. C., & Fleming, P. J. (2013). Generalized Decomposition. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, & J. Shaw (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 428–442). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Goel, T., Vaidyanathan, R., Haftka, R. T., Shyy, W., Queipo, N. V., & Tucker, K. (2007). Response surface approximation of Pareto optimal front in multi-objective optimization. *Computer Methods in Applied Mechanics and Engineering*, 196(4), 879–893. doi:[10.1016/j.cma.2006.07.010](https://doi.org/10.1016/j.cma.2006.07.010)
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley.
- Gong, Z., Zhang, N., & Chiclana, F. (2018). The optimization ordering model for intuitionistic fuzzy preference relations with utility functions. *Knowledge-Based Systems*, 162, 174–184. doi:[10.1016/j.knosys.2018.07.012](https://doi.org/10.1016/j.knosys.2018.07.012)
- González-Gallardo, S., Saborido, R., Ruiz, A. B., & Luque, M. (2021). Preference-Based Evolutionary Multiobjective Optimization Through the Use of Reservation and Aspiration Points. *IEEE Access*, 9, 108861–108872. doi:[10.1109/ACCESS.2021.3101899](https://doi.org/10.1109/ACCESS.2021.3101899)
- Greco, S., Matarazzo, B., & Słowiński, R. (2010). Interactive evolutionary multiobjective optimization using dominance-based rough set approach. In H. Ishibuchi et al. (Eds.), *Proceedings of the 2010 Congress on Evolutionary Computation (CEC 2010)* (pp. 1–8). Piscataway, NJ: IEEE Press.
- Greenwood, G. W., Hu, X., & D'Ambrosio, J. G. (1996). Fitness functions for multiple objective optimization problems: Combining preferences with Pareto rankings. In R. K.

- Belew & M. D. Vose (Eds.), *Foundations of Genetic Algorithms (FOGA)* (pp. 437–455). Morgan Kaufmann Publishers.
- Guillén-Gosálbez, G. (2011a). A novel MILP-based objective reduction method for multi-objective optimization: Application to environmental problems. *Computers and Chemical Engineering*, 35(8), 1469–1477. Energy & Sustainability. doi:[10.1016/j.compchemeng.2011.02.001](https://doi.org/10.1016/j.compchemeng.2011.02.001)
- Guillén-Gosálbez, G. (2011b). A novel MILP-based objective reduction method for multi-objective optimization: Application to environmental problems. *Computers & Chemical Engineering*, 35(8), 1469–1477. doi:[10.1016/j.compchemeng.2011.02.001](https://doi.org/10.1016/j.compchemeng.2011.02.001)
- Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1), 389–422.
- Hakanen, J., Chugh, T., Sindhya, K., Jin, Y., & Miettinen, K. (2016). Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1–8). doi:[10.1109/SSCI.2016.7850220](https://doi.org/10.1109/SSCI.2016.7850220)
- Har-Peled, S., Roth, D., & Zimak, D. (2002). Constraint Classification for Multiclass Classification and Ranking. In S. Becker, S. Thrun, & K. Obermayer (Eds.), *Advances in Neural Information Processing Systems* (Vol. 15), MIT Press. Retrieved from <https://proceedings.neurips.cc/paper/2002/file/16026d60ff9b54410b3435b403afd226-Paper.pdf>
- Heiman, G. W. (2000). *Understanding Research Methods and Statistics: An Integrated Introduction for Psychology* (2nd ed.). Wadsworth Publishing Company.
- Herbrich, R., Graepel, T., & Obermayer, K. (1999). Support vector learning for ordinal regression. In *Proceedings of the 9th International Conference on Artificial Neural Networks: ICANN '99, Location: Edinburgh, UK, 7-10 Sept. 1999* (pp. 97–102). doi:[10.1049/cp:19991091](https://doi.org/10.1049/cp:19991091)
- Horn, J. (1997). Multicriterion decision making, *Handbook of Evolutionary Computation*, Volume 1, F1. 9: 1–F1. 9: 15. IOP Publishing Ltd. and Oxford University Press.
- Huber, S., Geiger, M. J., & Sevaux, M. (2015). Simulation of Preference Information in an Interactive Reference Point-Based Method for the Bi-Objective Inventory Routing Problem. *Journal of Multi-Criteria Decision Analysis*, 22(1-2), 17–35. doi:[10.1002/mcda.1534](https://doi.org/10.1002/mcda.1534)

- Hughes, E. (2005). Evolutionary many-objective optimisation: many once or one many? In *Proceedings of the 2005 Congress on Evolutionary Computation (CEC 2005)* (222–227 Vol.1). doi:[10.1109/CEC.2005.1554688](https://doi.org/10.1109/CEC.2005.1554688)
- Hüllermeier, E., Fürnkranz, J., Cheng, W., & Brinker, K. (2008). Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16), 1897–1916. doi:[10.1016/j.artint.2008.08.002](https://doi.org/10.1016/j.artint.2008.08.002)
- Ishibuchi, H., Tsukamoto, N., & Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China* (pp. 2419–2426). doi:[10.1109/CEC.2008.4631121](https://doi.org/10.1109/CEC.2008.4631121)
- Jacquet-Lagrange, E., & Siskos, Y. (2001). Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research*, 130(2), 233–245.
- Jaimes, A. L., Montaña, A. A., & Coello, C. A. C. (2011). Preference incorporation to solve many-objective airfoil design problems. In *2011 IEEE Congress of Evolutionary Computation (CEC)* (pp. 1605–1612).
- Jain, A., & Zongker, D. (1997). Feature selection: evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2), 153–158. doi:[10.1109/34.574797](https://doi.org/10.1109/34.574797)
- Jaszkiewicz, A. (2004). Interactive multiple objective optimization with the pareto memetic algorithm. In *4th EU/ME workshop: design and evaluation of advanced hybrid meta-heuristics, Nottingham, UK*. Citeseer.
- Jaszkiewicz, A., & Branke, J. (2008). Interactive Multiobjective Evolutionary Algorithms. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Vol. 5252, pp. 179–193). doi:[10.1007/978-3-540-88908-3_7](https://doi.org/10.1007/978-3-540-88908-3_7)
- Jensen, M. T. (2003). Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. *IEEE Transactions on Evolutionary Computation*, 7(5), 503–515.
- Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2), 263–291. doi:[10.2307/1914185](https://doi.org/10.2307/1914185)
- Karahan, İ., & Köksalan, M. (2010). A territory defining multiobjective evolutionary algorithms and preference incorporation. *IEEE Transactions on Evolutionary Computation*, 14(4), 636–664. doi:[10.1109/TEVC.2009.2033586](https://doi.org/10.1109/TEVC.2009.2033586)

- Keeney, R. L. (1981). Analysis of preference dependencies among objectives. *Operations Research*, 29, 1105–1120.
- Keeney, R., & Raiffa, H. (1993). *Decisions with multiple objectives: Preferences and value tradeoffs*. Cambridge University Press.
- Khare, V., Yao, X., & Deb, K. (2003). Performance Scaling of Multi-objective Evolutionary Algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2003* (Vol. 2632, pp. 376–390). Heidelberg: Springer.
- Knowles, J. D., & Corne, D. W. (2000a). M-PAES: A memetic algorithm for multiobjective optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)* (Vol. 1, pp. 325–332). IEEE.
- Knowles, J. D. (2002). *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization* (Doctoral dissertation, University of Reading, UK).
- Knowles, J. D. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1), 50–66.
- Knowles, J. D., & Corne, D. (2000b). Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2), 149–172. doi:[10.1162/106365600568167](https://doi.org/10.1162/106365600568167)
- Knowles, J. D., & Corne, D. (2002). On Metrics for Comparing Non-Dominated Sets. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'02)* (pp. 711–716). Piscataway, NJ: IEEE Press.
- Knowles, J. D., & Corne, D. (2003). Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2), 100–116.
- Knowles, J. D., Corne, D., & Fleischer, M. (2003). Bounded archiving using the Lebesgue measure. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'03)* (pp. 2490–2497). Piscataway, NJ: IEEE Press.
- Knowles, J. D., Thiele, L., & Zitzler, E. (2006). *A tutorial on the performance assessment of stochastic multiobjective optimizers* (TIK-Report No. 214). Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zürich, Switzerland. Revised version.

- Köksalan, M., & Karahan, İ. (2010). An Interactive Territory Defining Evolutionary Algorithm: iTDEA. *IEEE Transactions on Evolutionary Computation*, 14(5), 702–722. doi:[10.1109/TEVC.2010.2070070](https://doi.org/10.1109/TEVC.2010.2070070)
- Köksalan, M., & Phelps, S. (2007). An evolutionary metaheuristic for approximating preference-nondominated solutions. *INFORMS Journal on Computing*, 19(2), 291–301.
- Koppen, M., & Yoshida, K. (2007). Visualization of Pareto-sets in evolutionary multi-objective optimization. In *7th International Conference on Hybrid Intelligent Systems (HIS 2007)* (pp. 156–161). IEEE.
- Korhonen, P. J., & Laakso, J. (1986). A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 24(2), 277–287. Mathematical Programming Multiple Criteria Decision Making. doi:[10.1016/0377-2217\(86\)90050-0](https://doi.org/10.1016/0377-2217(86)90050-0)
- Korn, F., Pagel, B.-U., & Faloutsos, C. (2001). On the “dimensionality curse” and the “self-similarity blessing”. *IEEE Transactions on Knowledge and Data Engineering*, 13(1), 96–111. doi:[10.1109/69.908983](https://doi.org/10.1109/69.908983)
- Kornbluth, J. (1985). Sequential multi-criterion decision making. *Omega*, 13(6), 569–574. doi:[10.1016/0305-0483\(85\)90045-3](https://doi.org/10.1016/0305-0483(85)90045-3)
- Kraskov, A., Stögbauer, H., & Grassberger, P. (2011). Erratum: Estimating mutual information [Phys. Rev. E 69, 066138 (2004)]. *Phys. Rev. E*, 83, 019903. doi:[10.1103/PhysRevE.83.019903](https://doi.org/10.1103/PhysRevE.83.019903)
- Kryston, A., Müller, M. F., Penny, G., Bolster, D., Tank, J. L., & Mondal, M. S. (2022). Addressing climate uncertainty and incomplete information in transboundary river treaties: A scenario-neutral dimensionality reduction approach. *Journal of Hydrology*, 612, 128004. doi:[10.1016/j.jhydrol.2022.128004](https://doi.org/10.1016/j.jhydrol.2022.128004)
- Larichev, O. I. (1992). Cognitive validity in design of decision-aiding techniques. *Journal of Multi-Criteria Decision Analysis*, 1(3), 127–138. doi:[10.1002/mcda.4020010303](https://doi.org/10.1002/mcda.4020010303)
- Lerch, F. J., & Harter, D. E. (2001). Cognitive Support for Real-Time Dynamic Decision Making. *Information Systems Research*, 12(1), 63–82. doi:[10.1287/isre.12.1.63.9717](https://doi.org/10.1287/isre.12.1.63.9717)
- Li, B., Li, J., Tang, K., & Yao, X. (2015). Many-Objective Evolutionary Algorithms: A Survey. *ACM Computing Surveys*, 48(1), 1–35.

- Li, K., Chen, R., Savić, D., & Yao, X. (2019). Interactive Decomposition Multiobjective Optimization Via Progressively Learned Value Functions. *IEEE Transactions on Fuzzy Systems*, 27(5), 849–860. doi:[10.1109/TFUZZ.2018.2880700](https://doi.org/10.1109/TFUZZ.2018.2880700)
- Li, K., Deb, K., & Yao, X. (2018). R-Metric: Evaluating the performance of preference-based evolutionary multiobjective optimization using reference points. *IEEE Transactions on Evolutionary Computation*, 22(6), 821–835. doi:[10.1109/TEVC.2017.2737781](https://doi.org/10.1109/TEVC.2017.2737781)
- Liu, H., & Kešelj, V. (2007). Combined mining of Web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data & Knowledge Engineering*, 61(2), 304–330. doi:[10.1016/j.datak.2006.06.001](https://doi.org/10.1016/j.datak.2006.06.001)
- López Jaimes, A., Coello Coello, C. A., & Chakraborty, D. (2008). Objective reduction using a feature selection technique. In C. Ryan (Ed.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2008* (pp. 673–680). New York, NY: ACM Press.
- López Jaimes, A., Coello Coello, C. A., & Urías Barrientos, J. E. (2009). Online Objective Reduction to Deal with Many-Objective Problems. In M. Ehrgott, C. M. Fonseca, X. Gandibleux, J.-K. Hao, & M. Sevaux (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2009* (Vol. 5467, pp. 423–437). Heidelberg: Springer.
- López-Ibáñez, M., & Knowles, J. D. (2015). Machine Decision Makers as a Laboratory for Interactive EMO. In A. Gaspar-Cunha, C. H. Antunes, & C. A. Coello Coello (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2015 Part II* (Vol. 9019, pp. 295–309). doi:[10.1007/978-3-319-15892-1_20](https://doi.org/10.1007/978-3-319-15892-1_20)
- López-Jaimes, A., & Coello Coello, C. A. (2014). Including preferences into a multiobjective evolutionary algorithm to deal with many-objective engineering optimization problems. *Information Sciences*, 277, 1–20. doi:[10.1016/j.ins.2014.04.023](https://doi.org/10.1016/j.ins.2014.04.023)
- Luque, M. (2015). Modified interactive chebyshev algorithm (MICA) for non-convex multi-objective programming. *Optimization Letters*, 9(1), 173–187. doi:[10.1007/s11590-014-0743-9](https://doi.org/10.1007/s11590-014-0743-9)
- Luque, M., Caballero, R., Molina, J., & Ruiz, F. (2007). Equivalent Information for Multi-objective Interactive Procedures. *Management Science*, 53(1), 125–134. doi:[10.1287/mnsc.1060.0595](https://doi.org/10.1287/mnsc.1060.0595)

- Luque, M., Miettinen, K., Eskelinen, P., & Ruiz, F. (2009). Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2), 450–462. doi:[10.1016/j.omega.2007.06.001](https://doi.org/10.1016/j.omega.2007.06.001)
- Lygoe, R. J., Cary, M., & Fleming, P. J. (2010). A Many-Objective Optimisation Decision-Making Process Applied to Automotive Diesel Engine Calibration. In K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakroborty, S. Das, J. Dutta, ... K. C. Tan (Eds.), *Simulated Evolution and Learning* (pp. 638–646). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Mas-Colell, A., Whinston, M. D., Green, J. R., et al. (1995). *Microeconomic theory*. Oxford university press New York.
- Mattson, C., Mullur, A., & Messac, A. (2002). Minimal representation of multiobjective design space using a smart Pareto filter. In *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization* (p. 5458).
- Mattson, C. A., & Messac, A. (2003). Concept Selection Using s-Pareto Frontiers. *AIAA Journal*, 41(6), 1190–1198. doi:[10.2514/2.2063](https://doi.org/10.2514/2.2063)
- Miettinen, K. (1999). *Nonlinear Multiobjective Optimization*. Boston, MA: Kluwer Academic Publishers.
- Miettinen, K. (2001). Some Methods for Nonlinear Multi-objective Optimization. In E. Zitzler, L. Thiele, K. Deb, C. A. Coello Coello, & D. Corne (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 1–20). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Miettinen, K. (2008). Introduction to Multiobjective Optimization: Noninteractive Approaches. In *Multiobjective Optimization - Interactive and Evolutionary Approaches* (Chap. 1, pp. 1–26). Springer.
- Miettinen, K., & Kirilov, L. (2005). Interactive Reference Direction Approach Using Implicit Parametrization for Nonlinear Multiobjective Optimization. *Journal of Multi-Criteria Decision Analysis*, 13(2-3), 115–123. doi:[10.1002/mcda.377](https://doi.org/10.1002/mcda.377)
- Miettinen, K., & Mäkelä, M. M. (2000). Interactive multiobjective optimization system on the Internet. *Computers & Operations Research*, 27(7-8), 709–723.
- Miettinen, K., & Mäkelä, M. M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, 170(3), 909–922.
- Miettinen, K., Mustajoki, J., & Stewart, T. J. (2014). Interactive multiobjective optimization with NIMBUS for decision making under uncertainty. *OR Spectrum*, 36(1), 39–56.

- Miettinen, K., Podkopaev, D., Ruiz, F., & Luque, M. (2015). A new preference handling technique for interactive multiobjective optimization without trading-off. *Journal Of Global Optimization*, 63(4), 633–652. doi:[10.1007/s10898-015-0301-8](https://doi.org/10.1007/s10898-015-0301-8)
- Miettinen, K., Ruiz, F., & Wierzbicki, A. P. (2008). Introduction to Multiobjective Optimization: Interactive Approaches. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Vol. 5252). doi:[10.1007/978-3-540-88908-3_2](https://doi.org/10.1007/978-3-540-88908-3_2)
- Misitano, G., Afsar, B., Larraga, G., & Miettinen, K. (2022). Towards explainable interactive multiobjective optimization: R-XIMO. *AUTONOMOUS AGENTS AND MULTI-AGENT SYSTEMS*, 36(2). doi:[10.1007/s10458-022-09577-3](https://doi.org/10.1007/s10458-022-09577-3)
- Mitchell, T. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*.
- Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3), 301–312. doi:[10.1109/34.990133](https://doi.org/10.1109/34.990133)
- Mor, E., & Minguillon, J. (2003). An empirical evaluation of classifier combination schemes for predicting user navigational behavior. In *Proceedings ITCC 2003. International Conference on Information Technology: Coding and Computing* (pp. 467–471). doi:[10.1109/ITCC.2003.1197574](https://doi.org/10.1109/ITCC.2003.1197574)
- Mote, J., Olson, D. L., & Venkataramanan, M. A. (1988). A comparative multiobjective programming study. *Mathematical and Computer Modelling*, 10(10), 719–729. doi:[10.1016/0895-7177\(88\)90085-4](https://doi.org/10.1016/0895-7177(88)90085-4)
- Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Narukawa, K., Setoguchi, Y., Tanigaki, Y., Olhofer, M., Sendhoff, B., & Ishibuchi, H. (2016a). Preference representation using Gaussian functions on a hyperplane in evolutionary multi-objective optimization. *Soft Computing*, 20(7), 2733–2757. doi:[10.1007/s00500-015-1674-9](https://doi.org/10.1007/s00500-015-1674-9)
- Narukawa, K., Setoguchi, Y., Tanigaki, Y., Olhofer, M., Sendhoff, B., & Ishibuchi, H. (2016b). Preference representation using Gaussian functions on a hyperplane in evolutionary multi-objective optimization. *Soft Computing*, 20(7), 2733–2757.
- Obayashi, S., & Sasaki, D. (2003). Visualization and data mining of Pareto solutions using self-organizing map. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele

- (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2003* (Vol. 2632, pp. 796–809). Heidelberg: Springer.
- Ojalehto, V., Podkopaev, D., & Miettinen, K. (2016). Towards Automatic Testing of Reference Point Based Interactive Methods. In J. Handl, E. Hart, P. R. Lewis, M. López-Ibáñez, G. Ochoa, & B. Paechter (Eds.), *Parallel Problem Solving from Nature – PPSN XIV* (Vol. 9921, pp. 483–492). doi:[10.1007/978-3-319-45823-6_45](https://doi.org/10.1007/978-3-319-45823-6_45)
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Pedro, L. R., & Takahashi, R. H. C. (2013). Decision-Maker Preference Modeling in Interactive Multiobjective Optimization. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, & J. Shaw (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2013* (Vol. 7811, pp. 811–824). doi:[10.1007/978-3-642-37140-0_60](https://doi.org/10.1007/978-3-642-37140-0_60)
- Pedro, L. R., & Takahashi, R. H. (2014). INSPM: An interactive evolutionary multi-objective algorithm with preference model. *Information Sciences*, 268, 202–219. New Sensing and Processing Technologies for Hand-based Biometrics Authentication. doi:[10.1016/j.ins.2013.12.045](https://doi.org/10.1016/j.ins.2013.12.045)
- Phelps, S., & Köksalan, M. (2003). An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12), 1726–1738.
- Pozo, C., Ruíz-Femenia, R., Caballero, J., Guillén-Gosálbez, G., & Jiménez, L. (2012). On the use of Principal Component Analysis for reducing the number of environmental objectives in multi-objective optimization: Application to the design of chemical supply chains. *Chemical Engineering Science*, 69(1), 146–158. doi:[10.1016/j.ces.2011.10.018](https://doi.org/10.1016/j.ces.2011.10.018)
- Pu, P., & Chen, L. (2008). User-Involved Preference Elicitation for Product Search and Recommender Systems. *AI Magazine*, 29(4), 93. doi:[10.1609/aimag.v29i4.2200](https://doi.org/10.1609/aimag.v29i4.2200)
- Purshouse, R. C., Jalbă, C., & Fleming, P. J. (2011). Preference-driven co-evolutionary algorithms show promise for many-objective optimisation. In *International Conference on Evolutionary Multi-Criterion Optimization* (pp. 136–150). Springer.
- Purshouse, R. C., Deb, K., Mansor, M. M., Mostaghim, S., & Wang, R. (2014). *A review of hybrid evolutionary multiple criteria decision making methods* (COIN Report No. 2014005). Computational Optimization and Innovation (COIN) Laboratory, University of Michigan, USA.

- Purshouse, R. C., & Fleming, P. J. (2003a). Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-criterion Optimisation. In C. M. Fonseca, P. J. Fleming, E. Zitzler, L. Thiele, & K. Deb (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 16–30). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Purshouse, R. C., & Fleming, P. J. (2003b). Evolutionary many-objective optimisation: an exploratory analysis. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC 2003)*, Canberra, ACT, Australia (pp. 2066–2073). doi:[10.1109/CEC.2003.1299927](https://doi.org/10.1109/CEC.2003.1299927)
- Purshouse, R. C., & Fleming, P. J. (2007). On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Computation*, 11(6), 770–784. doi:[10.1109/TEVC.2007.910138](https://doi.org/10.1109/TEVC.2007.910138)
- Quan, G., Greenwood, G. W., Liu, D., & Hu, S. (2007a). Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms. *European Journal of Operational Research*, 177(3), 1969–1984. doi:[10.1016/j.ejor.2005.12.015](https://doi.org/10.1016/j.ejor.2005.12.015)
- Quan, G., Greenwood, G. W., Liu, D., & Hu, S. (2007b). Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms. *European Journal of Operational Research*, 177(3), 1969–1984. doi:[10.1016/j.ejor.2005.12.015](https://doi.org/10.1016/j.ejor.2005.12.015)
- Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning*, 1, 81–106.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Rachmawati, L., & Srinivasan, D. (2005). A hybrid fuzzy evolutionary algorithm for a multi-objective resource allocation problem. In *Fifth International Conference on Hybrid Intelligent Systems (HIS'05)*. doi:[10.1109/ICHIS.2005.10](https://doi.org/10.1109/ICHIS.2005.10)
- Rachmawati, L., & Srinivasan, D. (2006). Preference incorporation in multiobjective evolutionary algorithms: A survey. In *Proceedings of the 2006 Congress on Evolutionary Computation (CEC 2006)* (pp. 3385–3391). Piscataway, NJ: IEEE Press.
- Ramos-Pérez, J.-M., Miranda, G., Segredo, E., León, C., & Rodríguez-León, C. (2021). Application of Multi-Objective Evolutionary Algorithms for Planning Healthy and Balanced School Lunches. *Mathematics*, 9(1), 80. doi:[10.3390/math9010080](https://doi.org/10.3390/math9010080)

- Reeves, G. R., & Gonzalez, J. J. (1989). A comparison of two interactive MCDM procedures. *European Journal of Operational Research*, 41(2), 203–209. doi:[10.1016/0377-2217\(89\)90385-8](https://doi.org/10.1016/0377-2217(89)90385-8)
- Rokach, L., & Maimon, O. (2007). *Data Mining With Decision Trees: Theory And Applications*. World Scientific Publishing Company.
- Roos, D. S. (2001). Bioinformatics—trying to swim in a sea of data. *Science*, 291(5507), 1260–1261.
- Ross, B. C. (2014). Mutual Information between Discrete and Continuous Data Sets. *PLoS One*, 9(2), 1–5. doi:[10.1371/journal.pone.0087357](https://doi.org/10.1371/journal.pone.0087357)
- Roy, B., & Mousseau, V. (1996). A Theoretical Framework for Analysing the Notion of Relative Importance of Criteria. *Journal of Multi-Criteria Decision Analysis*, 5(2), 145–159. doi:[10.1002/\(SICI\)1099-1360\(199606\)5:2<145::AID-MCDA99>3.0.CO;2-5](https://doi.org/10.1002/(SICI)1099-1360(199606)5:2<145::AID-MCDA99>3.0.CO;2-5)
- Ruiz, A. B., Ruiz, F., Miettinen, K., Delgado-Antequera, L., & Ojalehto, V. (2019). NAUTILUS Navigator: free search interactive multiobjective optimization without trading-off. *Journal of Global Optimization*, 74(2), 213–231.
- Ruiz, A. B., Cabello, J. M., Platero, C. A., & Blázquez, F. (2015a). Multicriteria optimization of the investment in the auxiliary services of thermal power plants: A case study. *Energy Conversion and Management*, 93, 339–348. doi:[10.1016/j.enconman.2015.01.030](https://doi.org/10.1016/j.enconman.2015.01.030)
- Ruiz, A. B., Luque, M., Miettinen, K., & Saborido, R. (2015b). An Interactive Evolutionary Multiobjective Optimization Method: Interactive WASF-GA. In A. Gaspar-Cunha, C. H. Antunes, & C. A. Coello Coello (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2015 Part II* (Vol. 9019, pp. 249–263). doi:[10.1007/978-3-319-15892-1_17](https://doi.org/10.1007/978-3-319-15892-1_17)
- Sakawa, M. (1982). Interactive multiobjective decision making by the sequential proxy optimization technique: SPOT. *European Journal of Operational Research*, 9(4), 386–396. doi:[10.1016/0377-2217\(82\)90183-7](https://doi.org/10.1016/0377-2217(82)90183-7)
- Saxena, D. K., Ray, T., Deb, K., & Tiwari, A. (2009). Constrained many-objective optimization: A way forward. In *2009 IEEE Congress on Evolutionary Computation* (pp. 545–552). doi:[10.1109/CEC.2009.4982993](https://doi.org/10.1109/CEC.2009.4982993)
- Saxena, D. K., & Deb, K. (2007a). Non-linear Dimensionality Reduction Procedures for Certain Large-Dimensional Multi-objective Optimization Problems: Employing Correntropy and a Novel Maximum Variance Unfolding. In S. Obayashi et al. (Eds.), *Evo-*

- lutionary Multi-criterion Optimization, EMO 2007* (Vol. 4403, pp. 772–787). doi:[10.1007/978-3-540-70928-2_58](https://doi.org/10.1007/978-3-540-70928-2_58)
- Saxena, D. K., & Deb, K. (2007b). Trading on infeasibility by exploiting constraint’s criticality through multi-objectivization: A system design perspective. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, 25-28 September 2007, Singapore* (pp. 919–926). doi:[10.1109/CEC.2007.4424568](https://doi.org/10.1109/CEC.2007.4424568)
- Saxena, D. K., & Deb, K. (2008). Dimensionality reduction of objectives and constraints in multi-objective optimization problems: A system design perspective. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2008, June 1-6, 2008, Hong Kong, China* (pp. 3204–3211). doi:[10.1109/CEC.2008.4631232](https://doi.org/10.1109/CEC.2008.4631232)
- Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., & Zhang, Q. (2013). Objective Reduction in Many-Objective Optimization: Linear and Nonlinear Algorithms. *IEEE Transactions on Evolutionary Computation*, 17(1), 77–99. doi:[10.1109/TEVC.2012.2185847](https://doi.org/10.1109/TEVC.2012.2185847)
- Shavarani, S. M., López-Ibáñez, M., & Allmendinger, R. (2022). Detection of Hidden Objectives and Interactive Objective Reduction. doi:[10.48550/ARXIV.2206.01320](https://doi.org/10.48550/ARXIV.2206.01320)
- Shavarani, S. M., López-Ibáñez, M., Allmendinger, R., & Knowles, J. (2023). An Interactive Decision Tree-Based Evolutionary Multi-objective Algorithm. In M. Emmerich, A. Deutz, H. Wang, A. V. Kononova, B. Naujoks, K. Li, ... I. Yevseyeva (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 620–634). Springer Nature Switzerland.
- Shavarani, S. M., López-Ibáñez, M., Allmendinger, R., & Knowles, J. D. (2022). An Interactive Decision Tree-Based Evolutionary Multi-Objective Algorithm: Supplementary material. <https://doi.org/10.5281/zenodo.7429806>.
- Shavarani, S. M., López-Ibáñez, M., & Knowles, J. D. (2021). Realistic Utility Functions Prove Difficult for State-of-the-Art Interactive Multiobjective Optimization Algorithms. In F. Chicano & K. Krawiec (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2021* (pp. 457–465). doi:[10.1145/3449639.3459373](https://doi.org/10.1145/3449639.3459373)
- Shen, X., Guo, Y., Chen, Q., & Hu, W. (2010). A multi-objective optimization evolutionary algorithm incorporating preference information based on fuzzy logic. *Computational Optimization and Applications*, 46(1), 159–188. doi:[10.1007/s10589-008-9189-2](https://doi.org/10.1007/s10589-008-9189-2)
- Sindhya, K., Ruiz, A. B., & Miettinen, K. (2011). A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE. In R. H. C. Takahashi, K. Deb,

- E. F. Wanner, & S. Greco (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 212–225). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Singh, H. K., Isaacs, A., & Ray, T. (2011). A Pareto Corner Search Evolutionary Algorithm and Dimensionality Reduction in Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 15(4), 539–556. doi:[10.1109/TEVC.2010.2093579](https://doi.org/10.1109/TEVC.2010.2093579)
- Sinha, A., Malo, P., & Kallio, M. (2018). Convex preference cone-based approach for many objective optimization problems. *Computers & Operations Research*, 95, 1–11.
- Sinha, A., Saxena, D. K., Deb, K., & Tiwari, A. (2013). Using objective reduction and interactive procedure to handle many-objective optimization problems. *Applied Soft Computing*, 13(1), 415–427. doi:[10.1016/j.asoc.2012.08.030](https://doi.org/10.1016/j.asoc.2012.08.030)
- Steuer, R. E. (1986). *Multiple Criteria Optimization: Theory, Computation and Application*. New York, NY: John Wiley & Sons.
- Steuer, R. E., & Gardiner, L. (1991). On the Computational Testing of Procedures for Interactive Multiple Objective Linear Programming. In G. Fandel & H. Gehring (Eds.), *Operations Research* (pp. 121–131). doi:[10.1007/978-3-642-76537-7_8](https://doi.org/10.1007/978-3-642-76537-7_8)
- Stewart, T. J. (1996). Robustness of Additive Value Function Methods in MCDM. *Journal of Multi-Criteria Decision Analysis*, 5(4), 301–309.
- Stewart, T. J. (1999). Evaluation and refinement of aspiration-based methods in MCDM. *European Journal of Operational Research*, 113(3), 643–652.
- Stewart, T. J. (2005). Goal programming and cognitive biases in decision-making. *Journal of the Operational Research Society*, 56(10), 1166–1175. doi:[10.1057/palgrave.jors.2601948](https://doi.org/10.1057/palgrave.jors.2601948)
- Svore, K. M., & Burges, C. J. C. (2011). Large-Scale Learning to Rank Using Boosted Decision Trees. In R. Bekkerman, M. Bilenko, & J. Langford (Eds.), *Scaling up Machine Learning: Parallel and Distributed Approaches* (pp. 148–169). doi:[10.1017/CBO9781139042918.009](https://doi.org/10.1017/CBO9781139042918.009)
- Szczepański, M., & Wierzbicki, A. (2003). Application of multiple criteria evolutionary algorithms to vector optimisation, decision support and reference point approaches. *Journal of Telecommunications and Information Technology*, (3), 16–33.

- Thiele, L., Miettinen, K., Korhonen, P. J., & Molina, J. (2009). A Preference-Based Evolutionary Algorithm for Multi-Objective Optimization. *Evolutionary Computation*, 17(3), 411–436. doi:[10.1162/evco.2009.17.3.411](https://doi.org/10.1162/evco.2009.17.3.411)
- Tomczyk, M. K., & Kadziński, M. (2019a). Decomposition-based interactive evolutionary algorithm for multiple objective optimization. *IEEE Transactions on Evolutionary Computation*, 24(2), 320–334. doi:[10.1109/TEVC.2019.2915767](https://doi.org/10.1109/TEVC.2019.2915767)
- Tomczyk, M. K., & Kadziński, M. (2019b). EMOSOR: Evolutionary multiple objective optimization guided by interactive stochastic ordinal regression. *Computers & Operations Research*, 108, 134–154. doi:[10.1016/j.cor.2019.04.008](https://doi.org/10.1016/j.cor.2019.04.008)
- Tomczyk, M. K., & Kadziński, M. (2019c). Robust Indicator-Based Algorithm for Interactive Evolutionary Multiple Objective Optimization. In M. López-Ibáñez, A. Auger, & T. Stützle (Eds.), *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2019* (pp. 629–637). doi:[10.1145/3321707.3321742](https://doi.org/10.1145/3321707.3321742)
- Tomczyk, M. K., & Kadziński, M. (2020a). On the elicitation of indirect preferences in interactive evolutionary multiple objective optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (pp. 569–577).
- Tomczyk, M. K., & Kadziński, M. (2021a). Decomposition-based co-evolutionary algorithm for interactive multiple objective optimization. *Information Sciences*, 549, 178–199. doi:[10.1016/j.ins.2020.11.030](https://doi.org/10.1016/j.ins.2020.11.030)
- Tomczyk, M. K., & Kadziński, M. (2019d). Robust Indicator-Based Algorithm for Interactive Evolutionary Multiple Objective Optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference* (pp. 629–637). doi:[10.1145/3321707.3321742](https://doi.org/10.1145/3321707.3321742)
- Tomczyk, M. K., & Kadziński, M. (2020b). On the Elicitation of Indirect Preferences in Interactive Evolutionary Multiple Objective Optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (pp. 569–577). doi:[10.1145/3377930.3389826](https://doi.org/10.1145/3377930.3389826)
- Tomczyk, M. K., & Kadziński, M. (2021b). Decomposition-based co-evolutionary algorithm for interactive multiple objective optimization. *Information Sciences*, 549, 178–199. doi:[10.1016/j.ins.2020.11.030](https://doi.org/10.1016/j.ins.2020.11.030)
- Trianni, V., & López-Ibáñez, M. (2015). Advantages of Task-Specific Multi-Objective Optimisation in Evolutionary Robotics. *PLoS One*, 10(8), e0136406. doi:[10.1371/journal.pone.0136406](https://doi.org/10.1371/journal.pone.0136406)

- Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *Science*, 185(4157), 1124–1131.
- Van Veldhuizen, D. A., & Lamont, G. B. (2000). Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-art. *Evolutionary Computation*, 8(2), 125–147. doi:[10.1162/106365600568158](https://doi.org/10.1162/106365600568158)
- Verel, S., Liefvooghe, A., Jourdan, L., & Dhaenens, C. (2013). On the Structure of Multi-objective Combinatorial Search Space: MNK-landscapes with Correlated Objectives. *European Journal of Operational Research*, 227(2), 331–342. doi:[10.1016/j.ejor.2012.12.019](https://doi.org/10.1016/j.ejor.2012.12.019)
- Waheed, T., Bonnell, R., Prasher, S., & Paulet, E. (2006). Measuring performance in precision agriculture: CART—A decision tree approach. *Agricultural Water Management*, 84(1), 173–185. doi:[10.1016/j.agwat.2005.12.003](https://doi.org/10.1016/j.agwat.2005.12.003)
- Wallenius, J. (1975). Comparative Evaluation of Some Interactive Approaches to Multicriterion Optimization. *Management Science*, 21(12), 1387–1396.
- Wang, F., Li, Y., Zhang, H., Hu, T., & Shen, X.-L. (2019). An adaptive weight vector guided evolutionary algorithm for preference-based multi-objective optimization. *Swarm and Evolutionary Computation*, 49, 220–233. doi:[10.1016/j.swevo.2019.06.009](https://doi.org/10.1016/j.swevo.2019.06.009)
- Wang, R., Purshouse, R. C., & Fleming, P. J. (2013). “Whatever Works Best for You”- A New Method for a Priori and Progressive Multi-objective Optimisation. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, & J. Shaw (Eds.), *Evolutionary Multi-Criterion Optimization* (pp. 337–351). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Wang, R., Zhang, Q., & Zhang, T. (2016). Decomposition-Based Algorithms Using Pareto Adaptive Scalarizing Methods. *IEEE Transactions on Evolutionary Computation*, 20(6), 821–837. doi:[10.1109/TEVC.2016.2521175](https://doi.org/10.1109/TEVC.2016.2521175)
- Wang, S., Tang, J., & Liu, H. (2017). Feature Selection.
- Weber, M. (1987). Decision making with incomplete information. *European Journal of Operational Research*, 28(1), 44–57. doi:[10.1016/0377-2217\(87\)90168-8](https://doi.org/10.1016/0377-2217(87)90168-8)
- Weinberger, K. Q., & Saul, L. K. (2006). An Introduction to Nonlinear Dimensionality Reduction by Maximum Variance Unfolding. In A. Cohn (Ed.), *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts*,

- USA (Vol. 6, pp. 1683–1686). AAAI Press/MIT Press, Menlo Park, CA. Retrieved from <http://www.aaai.org/Library/AAAI/2006/aaai06-280.php>
- Weinberger, K. Q., Sha, F., & Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction. In C. E. Brodley (Ed.), *Machine Learning, Proceedings of the Twenty-first International Conference, ICML 2004, Banff, Alberta, Canada, July 4–8, 2004*. doi:[10.1145/1015330.1015345](https://doi.org/10.1145/1015330.1015345)
- Wierzbicki, A. P. (1982). A mathematical basis for satisficing decision making. *Mathematical modelling*, 3(5), 391–405.
- Wierzbicki, A. P. (1986). On the completeness and constructiveness of parametric characterizations to vector optimization problems. *Operations-Research-Spektrum*, 8(2), 73–87.
- Wierzbicki, A. P. (1999). Reference point approaches. In *Multicriteria decision making* (pp. 237–275). Springer.
- Wierzbicki, A. P. (1980). The Use of Reference Objectives in Multiobjective Optimisation. In G. Fandel & T. Gal (Eds.), *Multiple Criteria Decision Making Theory and Application* (pp. 468–486). doi:[10.1007/978-3-642-48782-8_32](https://doi.org/10.1007/978-3-642-48782-8_32)
- Xia, F., Zhang, W., Li, F., & Yang, Y. (2008). Ranking with decision tree. *Knowledge and information systems*, 17(3), 381–395. doi:[10.1007/s10115-007-0118-y](https://doi.org/10.1007/s10115-007-0118-y)
- Xin, B., Chen, L., Chen, J., Ishibuchi, H., Hirota, K., & Liu, B. (2018). Interactive Multiobjective Optimization: A Review of the State-of-the-Art. *IEEE Access*, 6, 41256–41279. doi:[10.1109/ACCESS.2018.2856832](https://doi.org/10.1109/ACCESS.2018.2856832)
- Xu, J.-W., Pokharel, P. P., Paiva, A. R. C., & Príncipe, J. C. (2006). Nonlinear Component Analysis Based on Correntropy. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN 2006, part of the IEEE World Congress on Computational Intelligence, WCCI 2006, Vancouver, BC, Canada, 16-21 July 2006* (pp. 1851–1855). doi:[10.1109/IJCNN.2006.246905](https://doi.org/10.1109/IJCNN.2006.246905)
- Yu, P.-L. (1973). A class of solutions for group decision problems. *Management Science*, 19(8), 936–946.
- Yu, P. L. H., Wan, W. M., & Lee, P. H. (2011). Decision Tree Modeling for Ranking Data. In J. Fürnkranz & E. Hüllermeier (Eds.), *Preference Learning* (pp. 83–106). doi:[10.1007/978-3-642-14125-6_5](https://doi.org/10.1007/978-3-642-14125-6_5)
- Zeleny, M. (1973). Compromise programming. *Multiple criteria decision making*.

- Zhang, Q., & Li, H. (2007). MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6), 712–731. doi:[10.1109/TEVC.2007.892759](https://doi.org/10.1109/TEVC.2007.892759)
- Zhao, H., & Ram, S. (2004). Constrained cascade generalization of decision trees. *IEEE Transactions on Knowledge and Data Engineering*, 16(6), 727–739. doi:[10.1109/TKDE.2004.3](https://doi.org/10.1109/TKDE.2004.3)
- Zhao, M., Zhang, K., Chen, G., Zhao, X., Yao, C., Sun, H., ... Yao, J. (2020). A surrogate-assisted multi-objective evolutionary algorithm with dimension-reduction for production optimization. *Journal of Petroleum Science and Engineering*, 192, 107192. doi:[10.1016/j.petrol.2020.107192](https://doi.org/10.1016/j.petrol.2020.107192)
- Zhu, T., Greiner, R., & Häubl, G. (2003). Learning a Model of a Web User's Interests. In P. Brusilovsky, A. Corbett, & F. de Rosis (Eds.), *User Modeling 2003* (pp. 65–75). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Zionts, S. (1981). A multiple criteria method for choosing among discrete alternatives. *European Journal of Operational Research*, 7(2), 143–147. Fourth EURO III Special Issue. doi:[10.1016/0377-2217\(81\)90275-7](https://doi.org/10.1016/0377-2217(81)90275-7)
- Zionts, S., & Wallenius, J. (1976). An interactive programming method for solving the multiple criteria problem. *Management science*, 22(6), 652–663.
- Zionts, S., & Wallenius, J. (1983). An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions. *Management Science*, 29(5), 519–529.
- Zitzler, E., Brockhoff, D., & Thiele, L. (2007). The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In S. Obayashi et al. (Eds.), *Evolutionary Multi-criterion Optimization, EMO 2007* (Vol. 4403, pp. 862–876). doi:[10.1007/978-3-540-70928-2_64](https://doi.org/10.1007/978-3-540-70928-2_64)
- Zitzler, E., Knowles, J. D., & Thiele, L. (2008). Quality Assessment of Pareto Set Approximations. In J. Branke, K. Deb, K. Miettinen, & R. Słowiński (Eds.), *Multiobjective Optimization: Interactive and Evolutionary Approaches* (Vol. 5252, pp. 373–404). doi:[10.1109/TEVC.2009.2016569](https://doi.org/10.1109/TEVC.2009.2016569)
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. (2003). Performance Assessment of Multiobjective Optimizers: an Analysis and Review. *IEEE*

Transactions on Evolutionary Computation, 7(2), 117–132. doi:[10.1109/TEVC.2003.](https://doi.org/10.1109/TEVC.2003.810758)

[810758](https://doi.org/10.1109/TEVC.2003.810758)

Zujevs, A., & Eiduks, J. (2008). Adaptive Multi-objective Optimization Procedure Model. In *Scientific Proceeding of Riga Technical University* (Vol. 5, pp. 46–54).

Zujevs, A., & Eiduks, J. (2011). New decision maker model for multiobjective optimization interactive methods. In *17th International Conference on Information and Software Technologies, Kaunas, Lithuania* (pp. 51–58).

Appendices

Appendix A

Supplementary Material to Chapter 3: On Benchmarking Interactive Evolutionary Multi-Objective Algorithms

This supplementary material provides all the figures required for the analysis of Chapter 3 that had been omitted for brevity.

A.1 Results

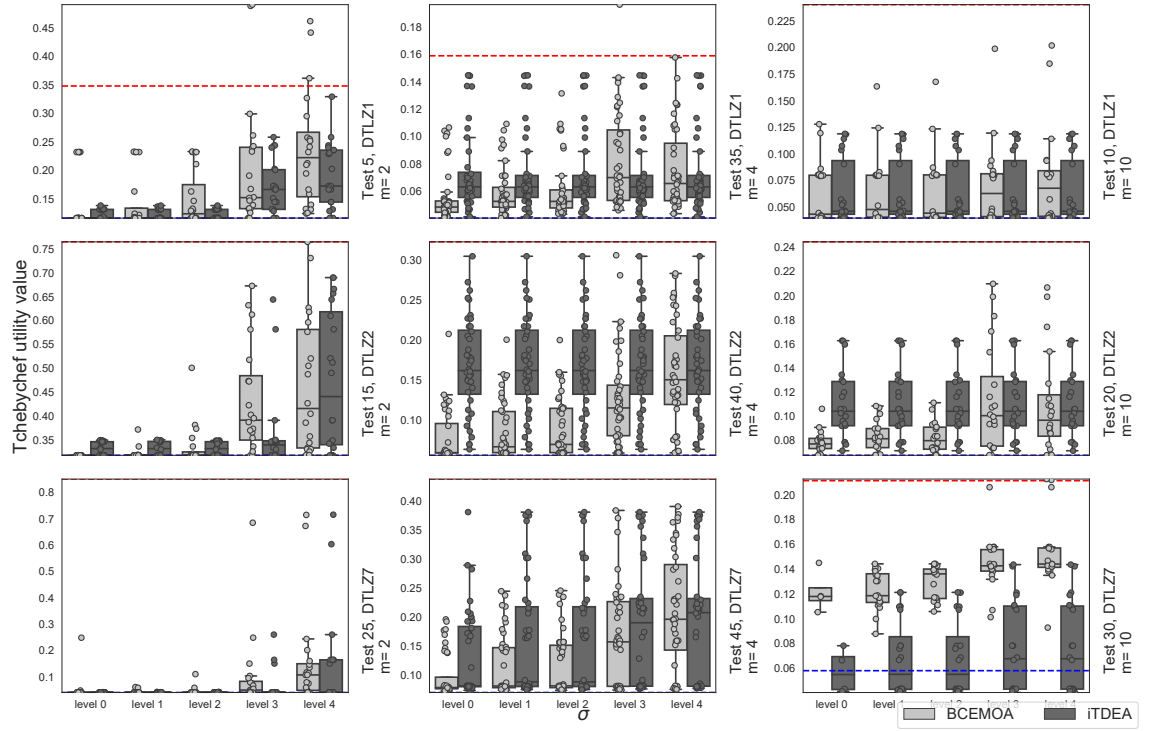


Figure A.1. Comparison of the performance of the algorithms with simulation of inconsistencies and Tchebychef utility function. The blue dashed line represents the utility value of the MPS and the red one illustrates worst utility value among PF solutions. Number of interactions is fixed to 3.

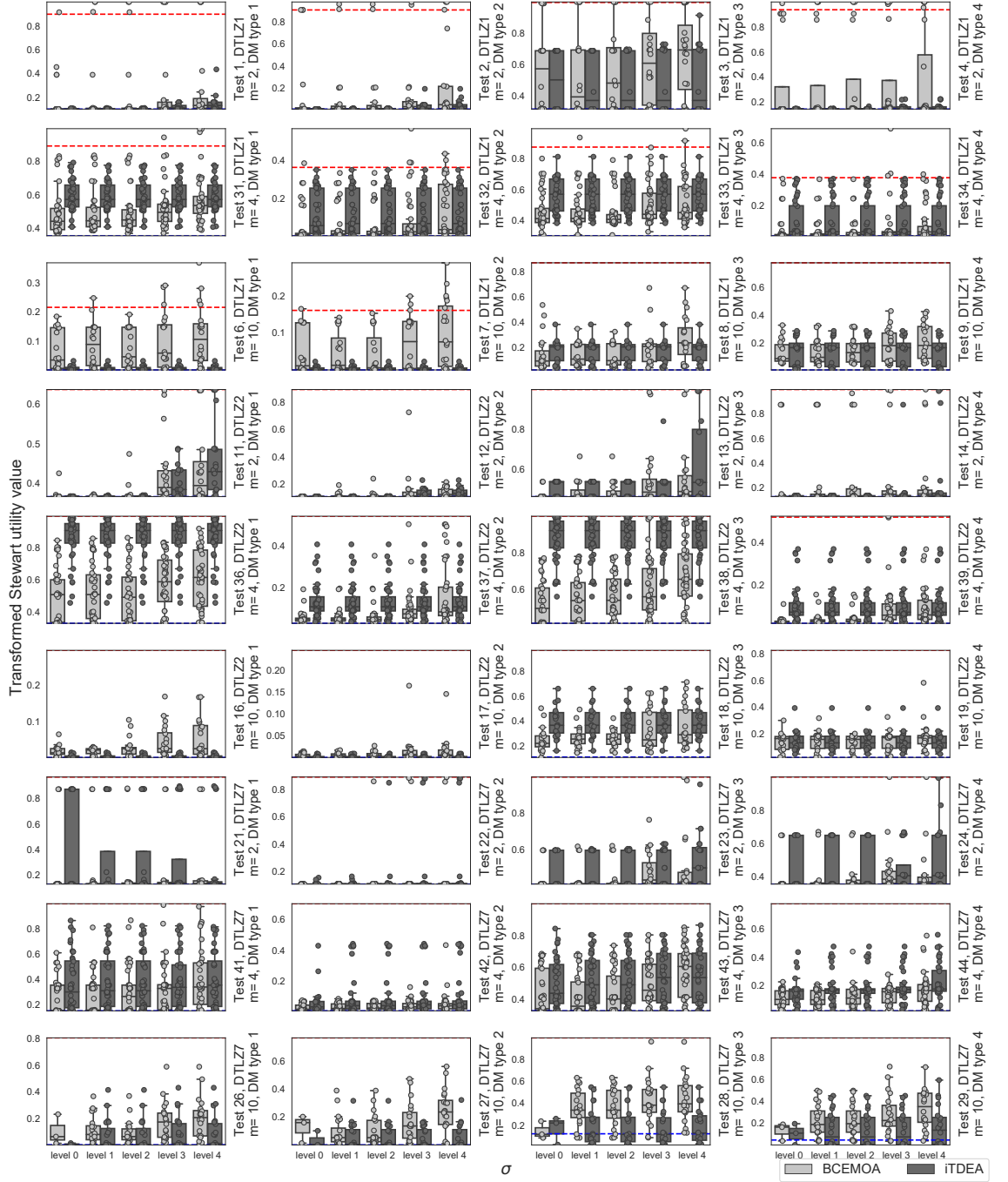


Figure A.2. Comparison of the performance of the algorithms with simulation of inconsistencies and transformed Stewart's utility function. The blue dashed line represents the utility value of the MPS and the red one illustrates worst utility value among PF solutions. Number of interactions is fixed to 3.

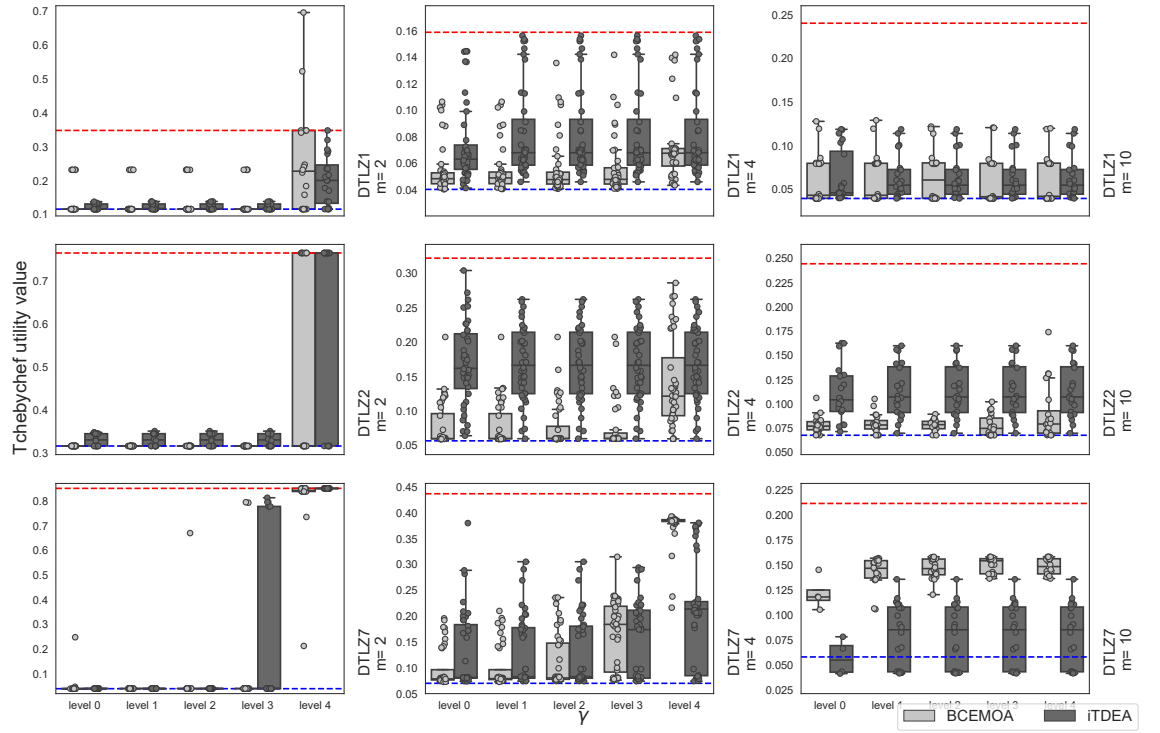


Figure A.3. Comparison of the performance of the algorithms with correlated objectives and Tchebycheff utility function. The blue dashed line represents the utility value of the MPS and the red one illustrates worst utility value among PF solutions. Number of interactions is fixed to 3.

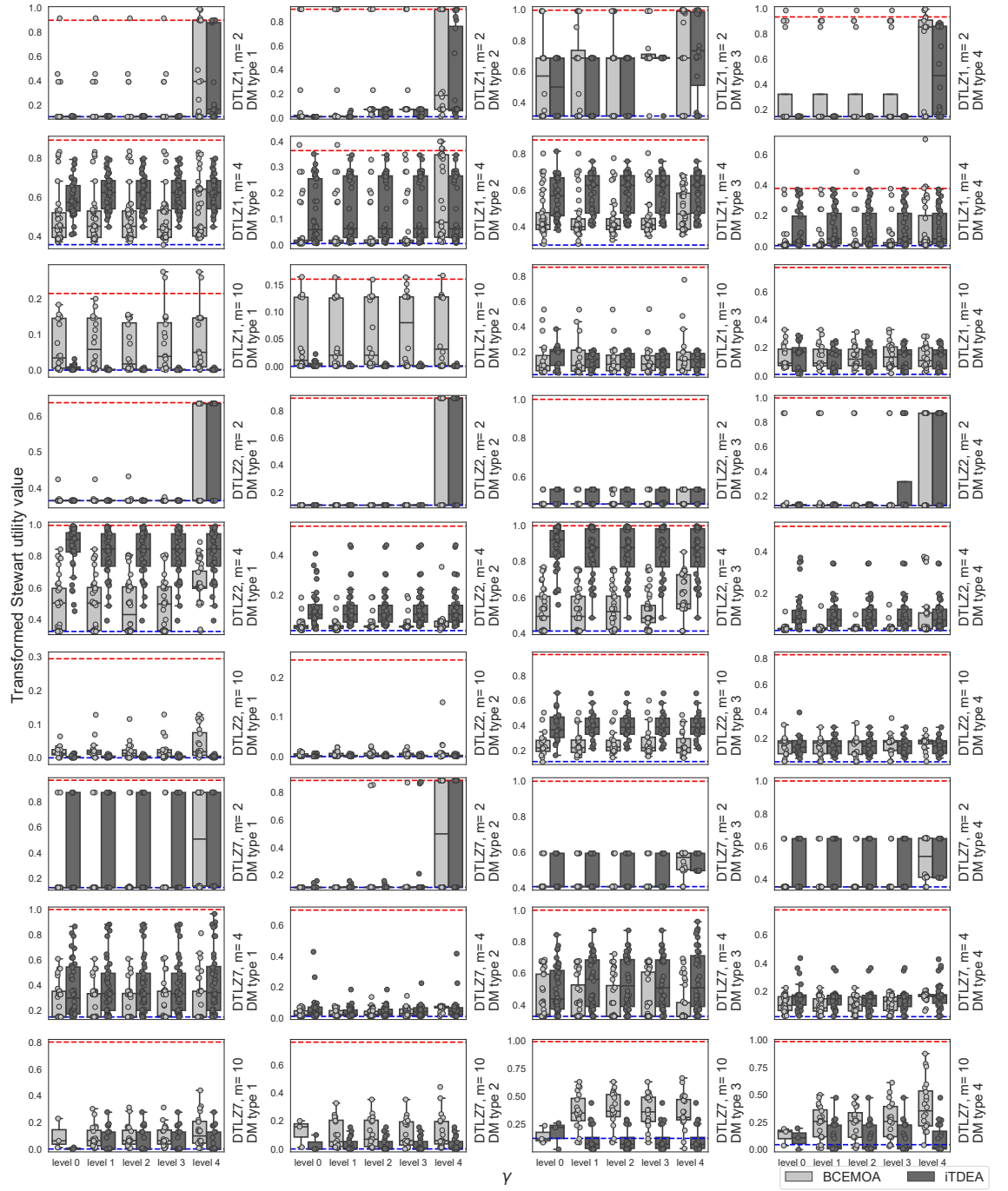


Figure A.4. Comparison of the performance of the algorithms with correlated objectives and the transformed Stewart's utility function. The blue dashed line represents the utility value of the MPS and the red one illustrates worst utility value among PF solutions. Number of interactions is fixed to 3.

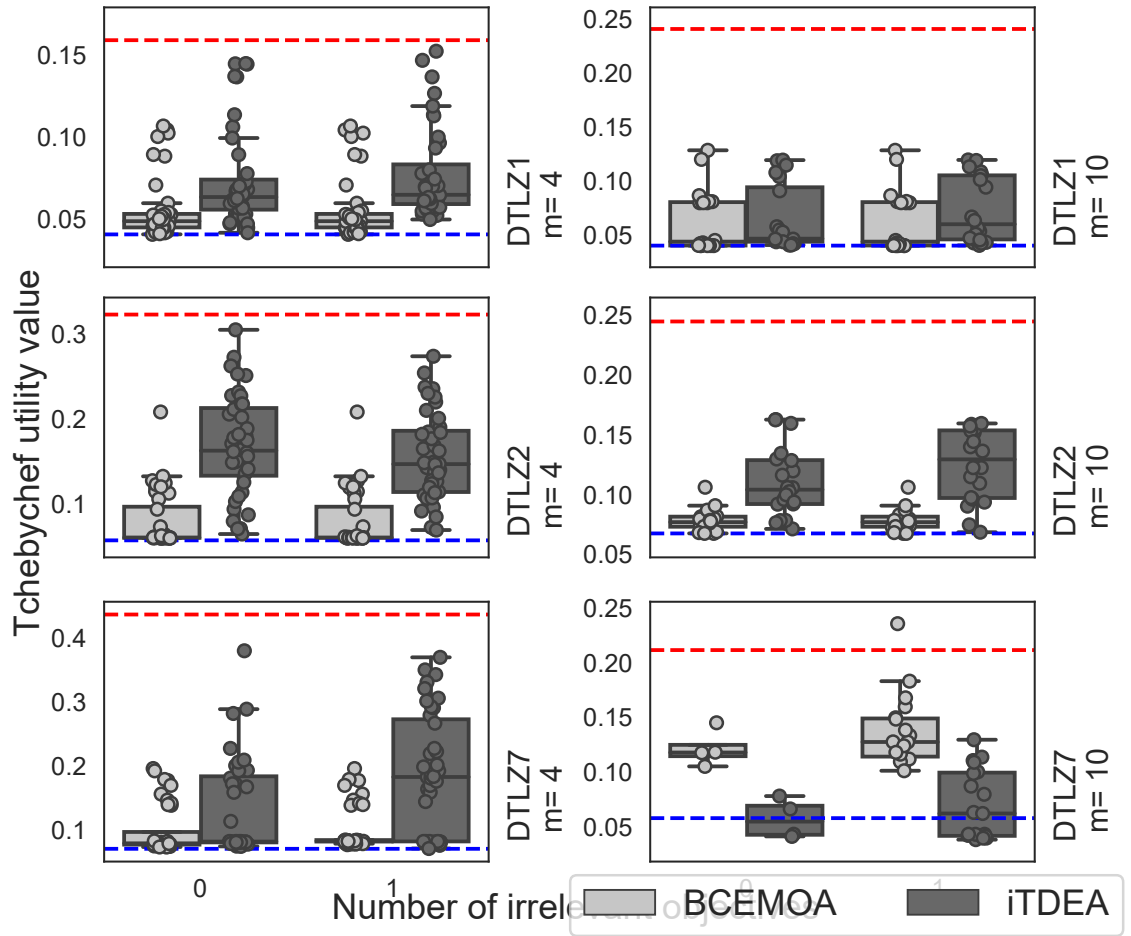


Figure A.5. Comparison of the performance of the algorithms with simulation of an irrelevant objective and the Tchebychev utility function. The blue dashed line represents the utility value of the MPS and the red one illustrates worst utility value among PF solutions. Number of interactions is fixed to 3.

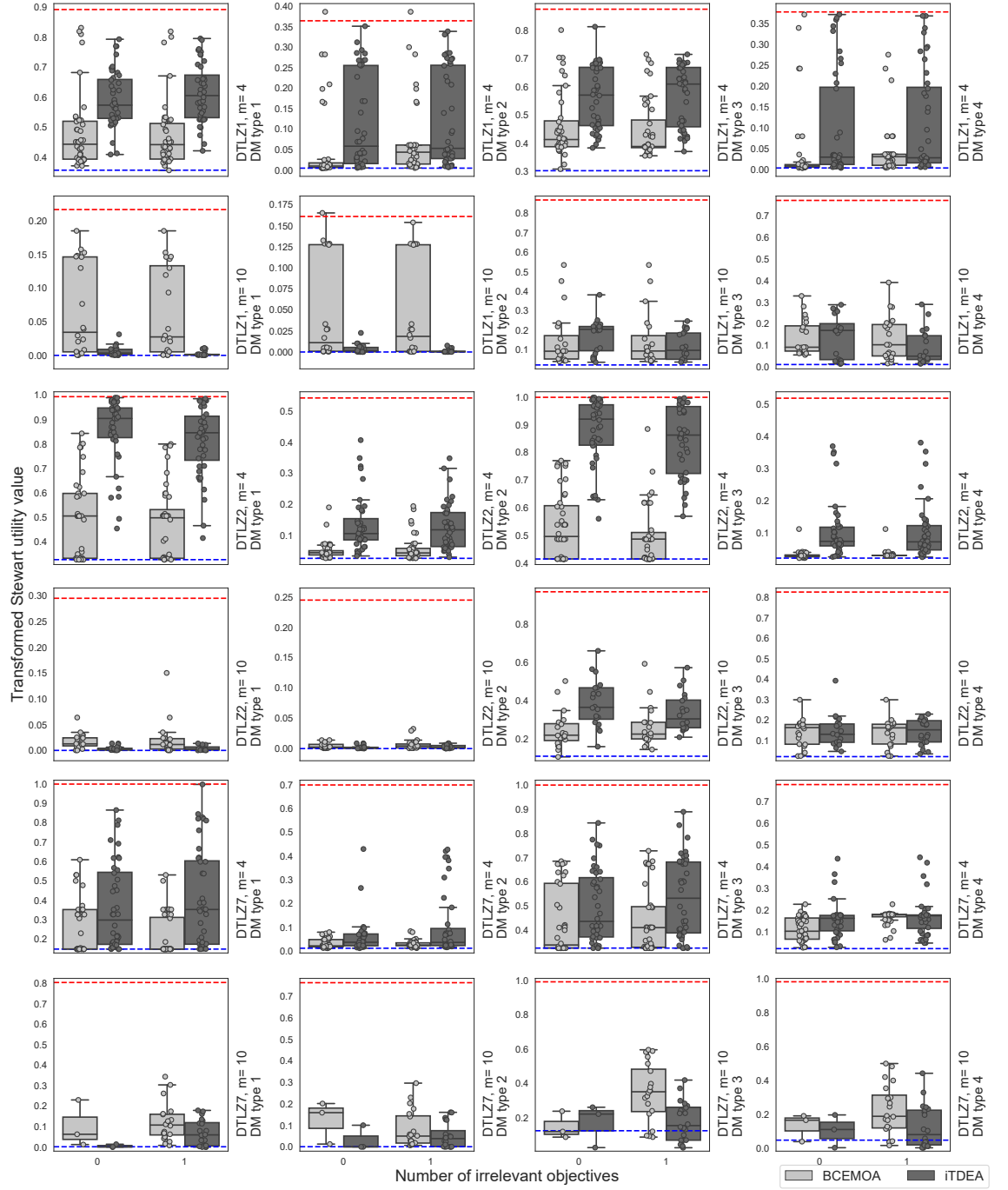


Figure A.6. Comparison of the performance of the algorithms with simulation of a irrelevant objective and the transformed Stewart's utility function. The blue dashed line represents the utility value of the MPS and the red one illustrates worst utility value among PF solutions. Number of interactions is fixed to 3.

Marginal utility functions

The marginal utility functions for tests with 4 objectives are provided in this Section. Objectives are discriminated by 4 different colors. The corresponding parameters of the UFs are depicted in the legend of each figure.

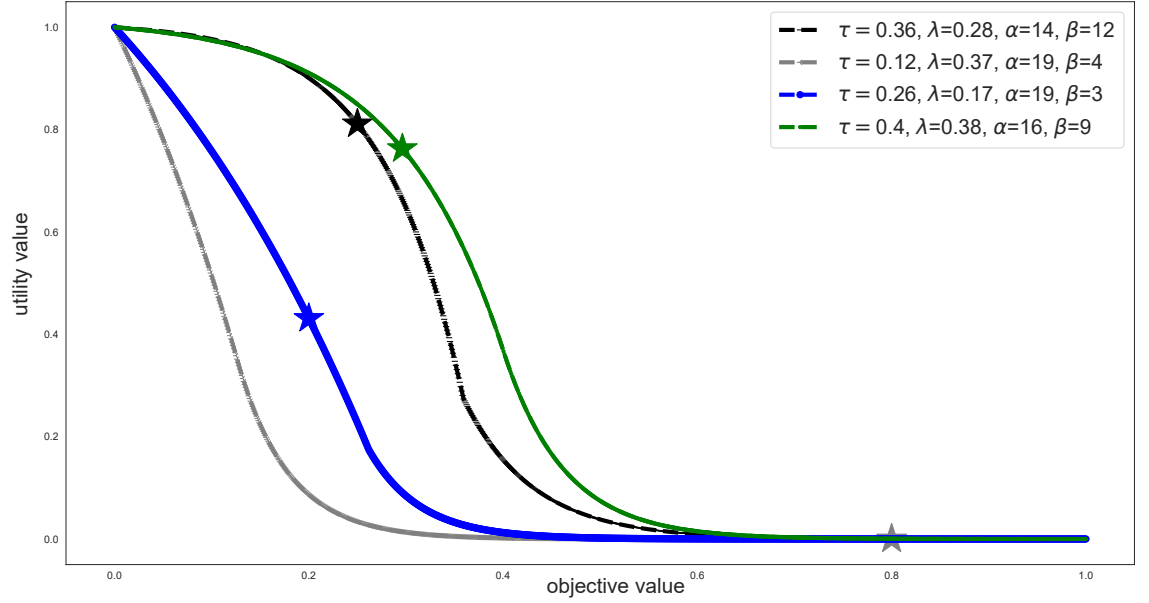


Figure A.7. Marginal utility functions for test 31. The values of the objectives in the MPS is depicted by an asterisk.

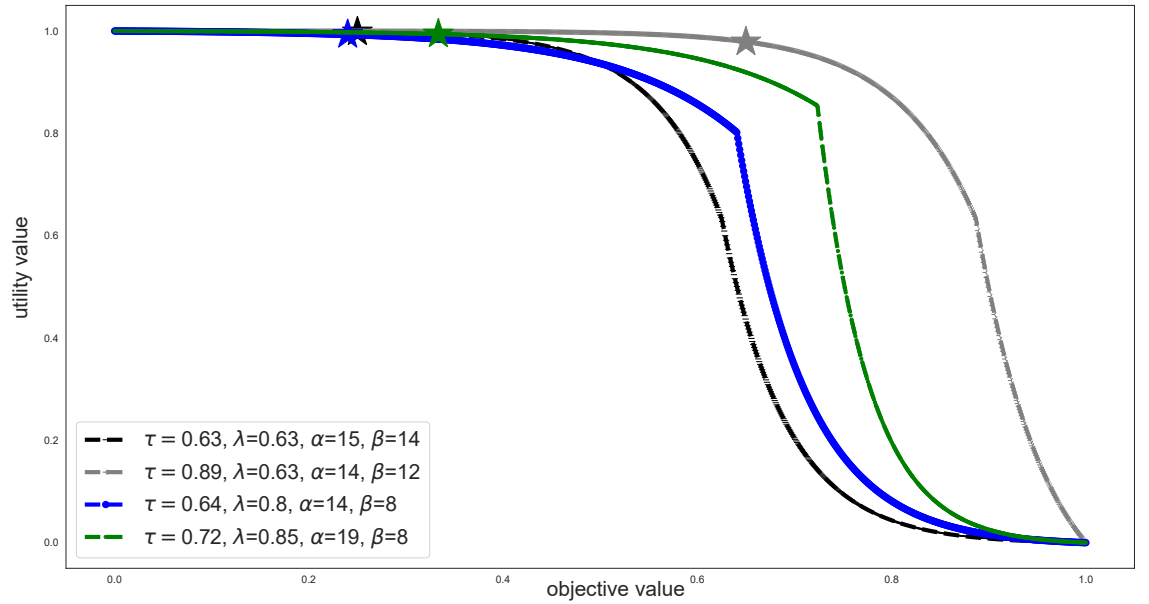


Figure A.8. Marginal utility functions for test 32. The values of the objectives in the MPS is depicted by an asterisk.

A.2 Heat map of utility values vs. the PF for problems with 2 objective

In this section we provide heatmaps of the utility functions for problems with 2 objectives. The effects of simulating different decision-making behaviors can be observed in these plots.

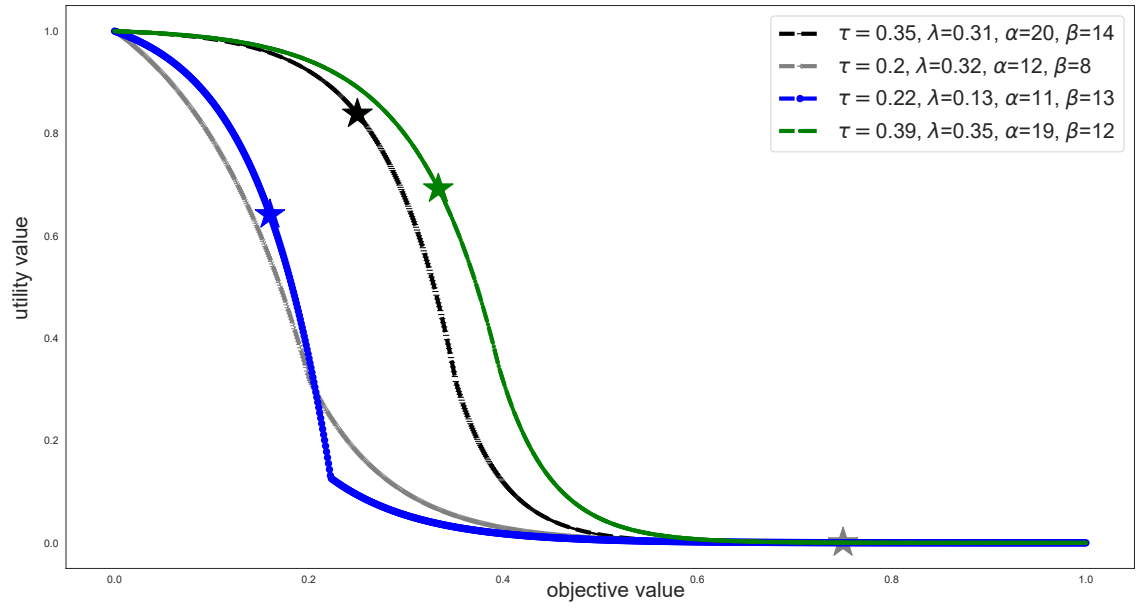


Figure A.9. Marginal utility functions for test 33. The values of the objectives in the MPS is depicted by an asterisk.

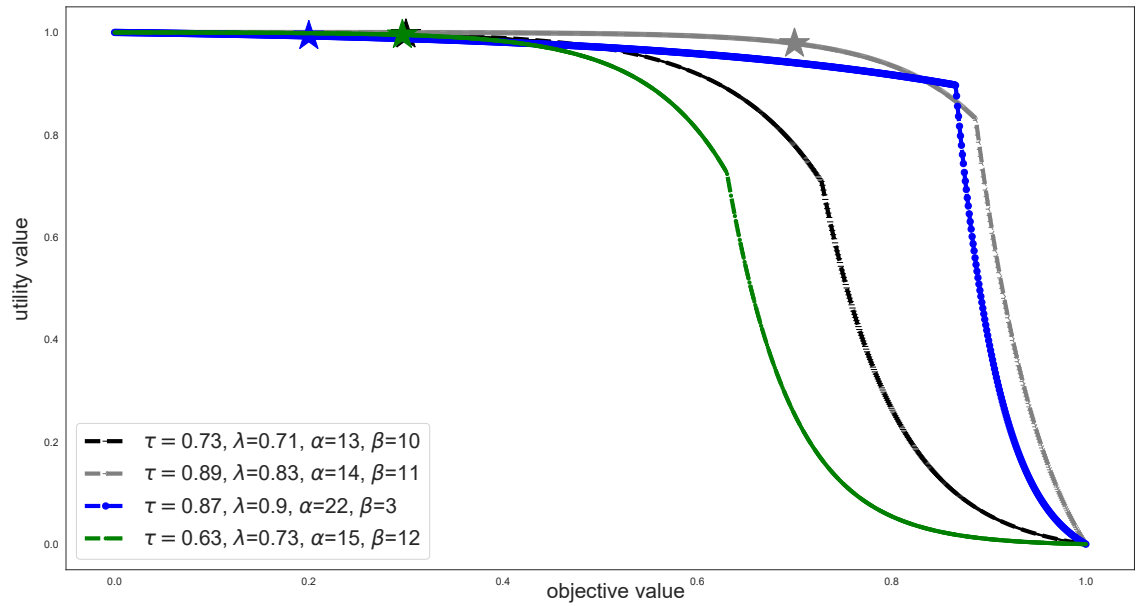


Figure A.10. Marginal utility functions for test 34. The values of the objectives in the MPS is depicted by an asterisk.

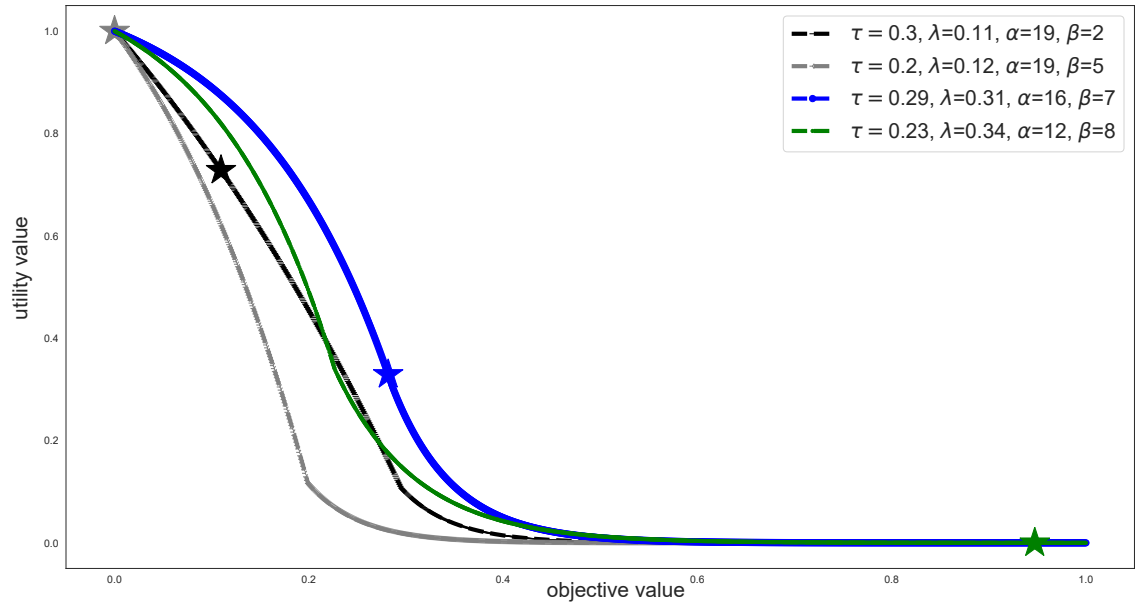


Figure A.11. Marginal utility functions for test 36. The values of the objectives in the MPS is depicted by an asterisk.

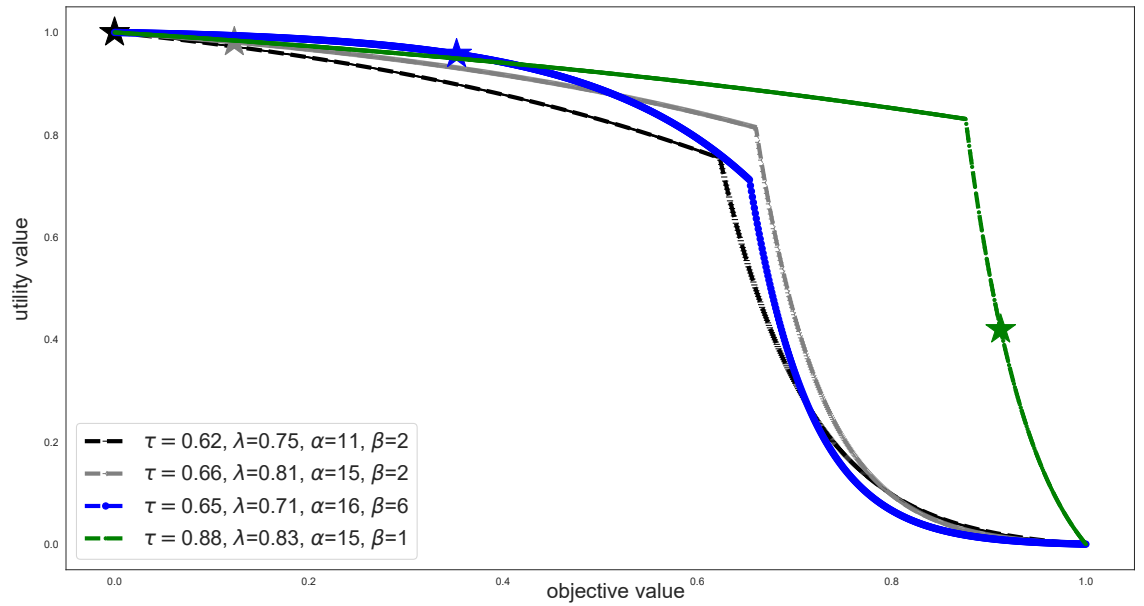


Figure A.12. Marginal utility functions for test 37. The values of the objectives in the MPS is depicted by an asterisk.

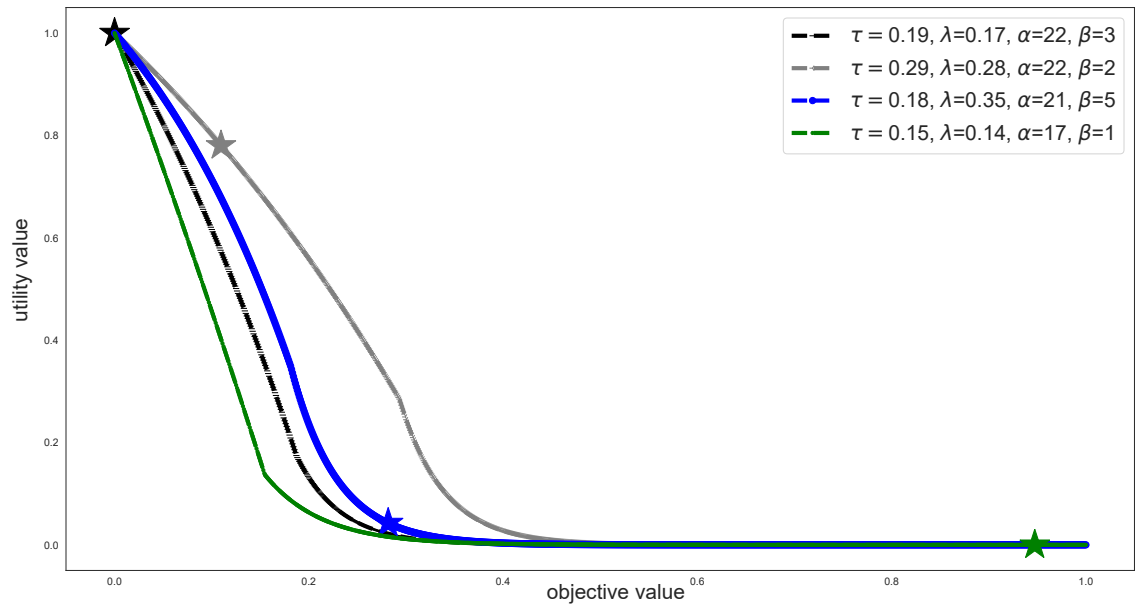


Figure A.13. Marginal utility functions for test 38. The values of the objectives in the MPS is depicted by an asterisk.

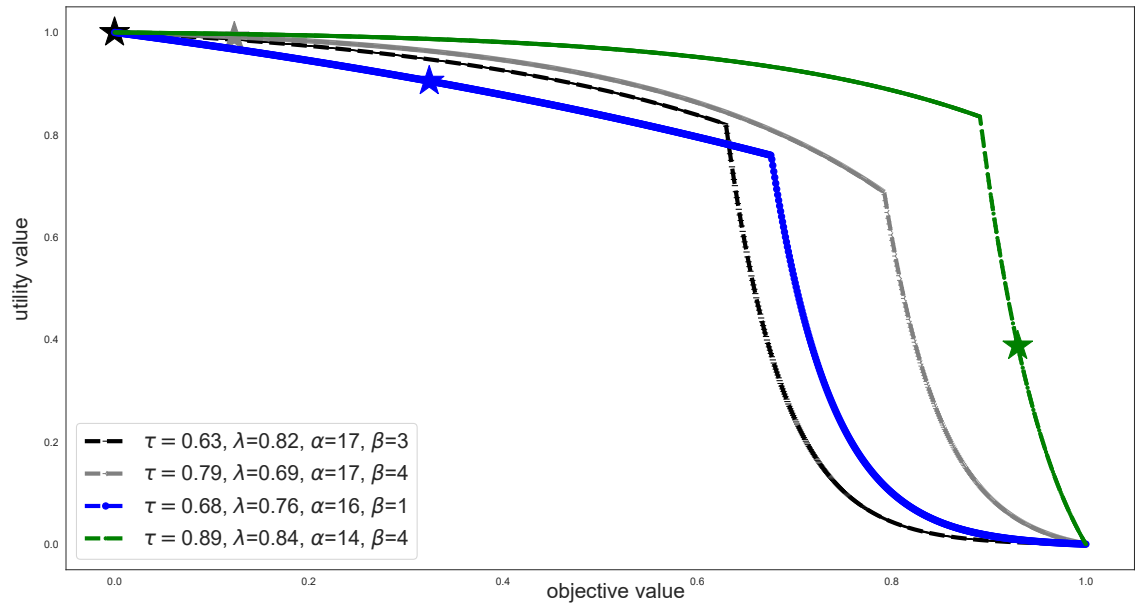


Figure A.14. Marginal utility functions for test 39. The values of the objectives in the MPS is depicted by an asterisk.

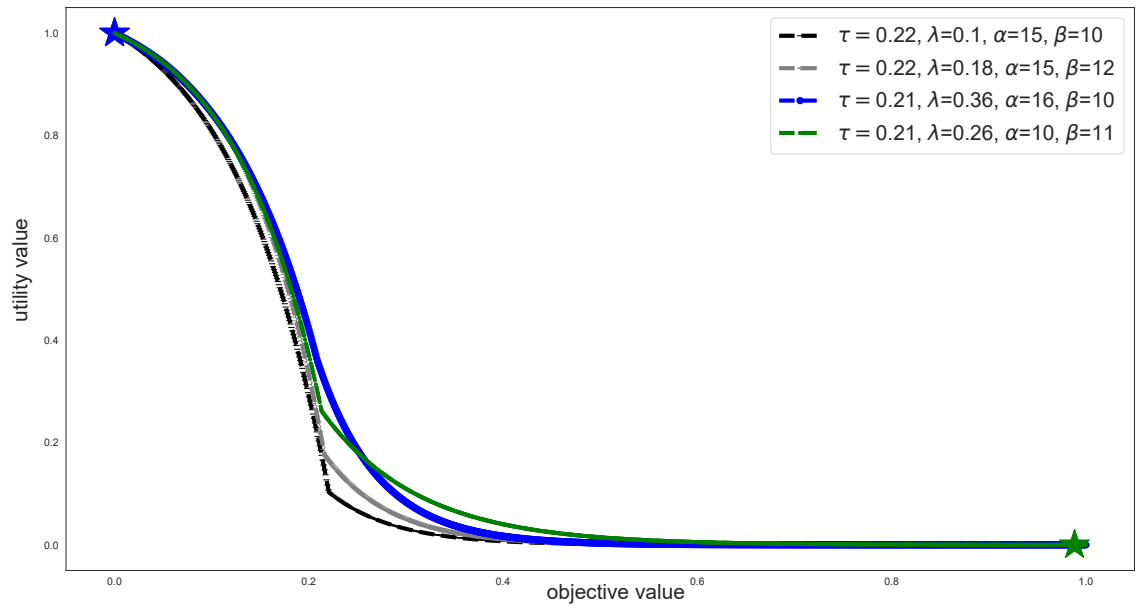


Figure A.15. Marginal utility functions for test 41. The values of the objectives in the MPS is depicted by an asterisk.

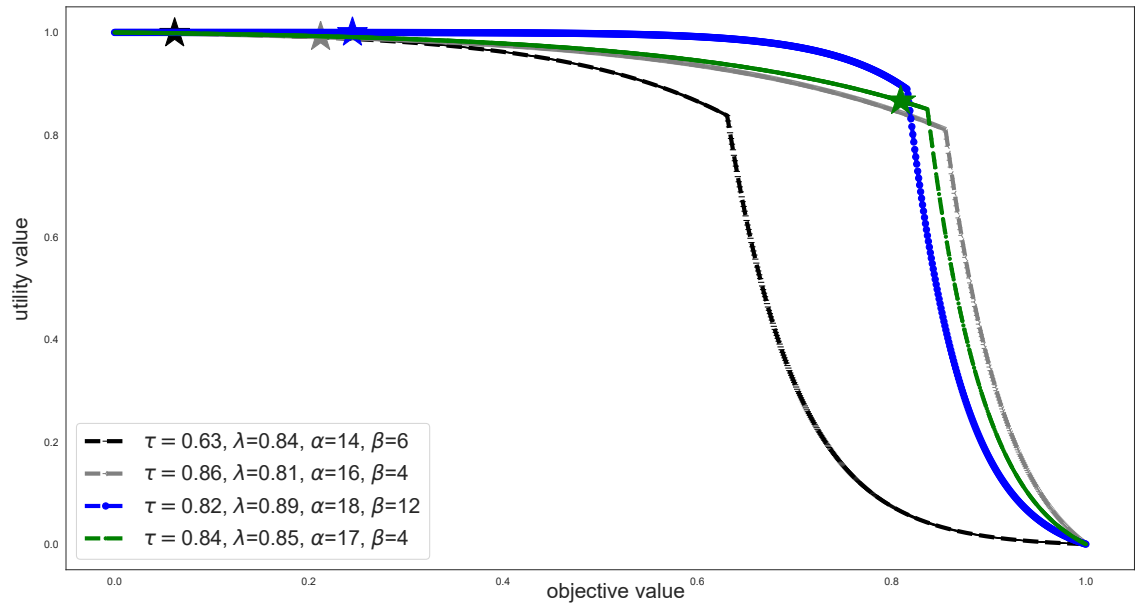


Figure A.16. Marginal utility functions for test 42. The values of the objectives in the MPS is depicted by an asterisk.

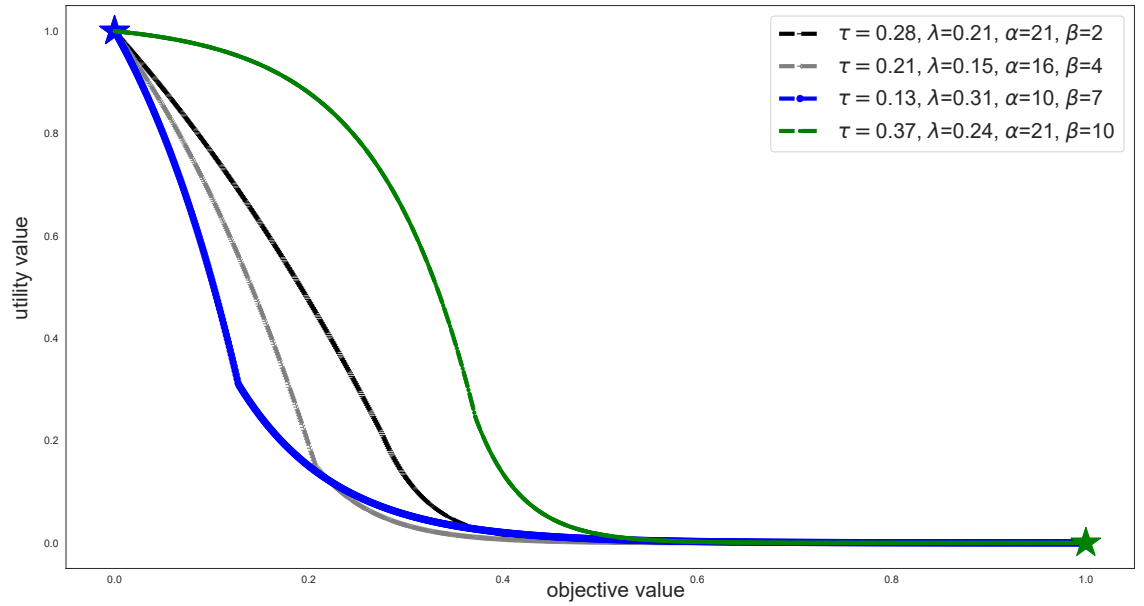


Figure A.17. Marginal utility functions for test 43. The values of the objectives in the MPS is depicted by an asterisk.

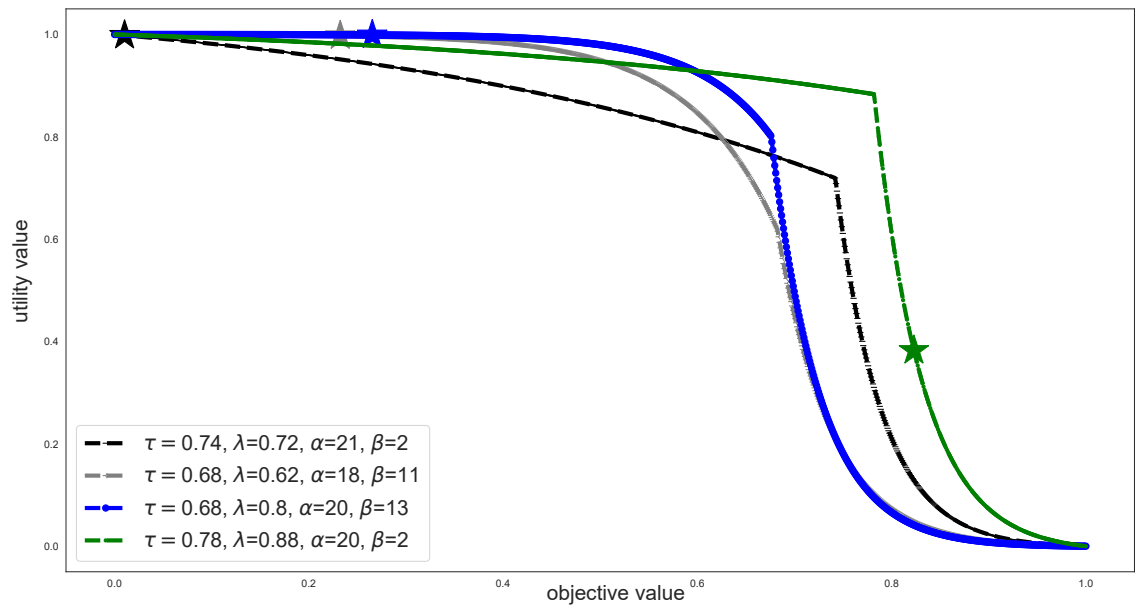


Figure A.18. Marginal utility functions for test 44. The values of the objectives in the MPS is depicted by an asterisk.

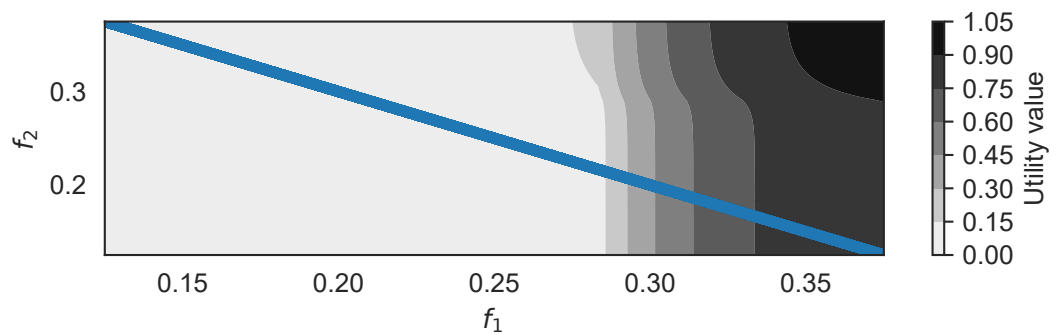


Figure A.19. Heatmap vs. Pf : DTLZ1, DM type 1

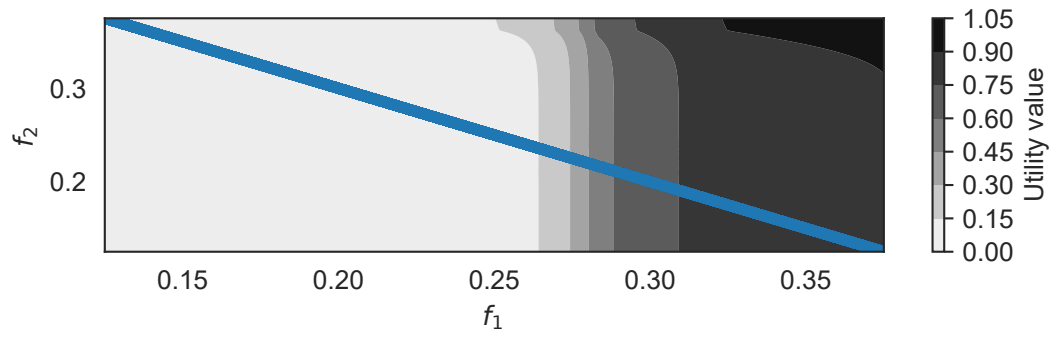


Figure A.20. Heatmap vs. Pf : DTLZ1, DM type 2

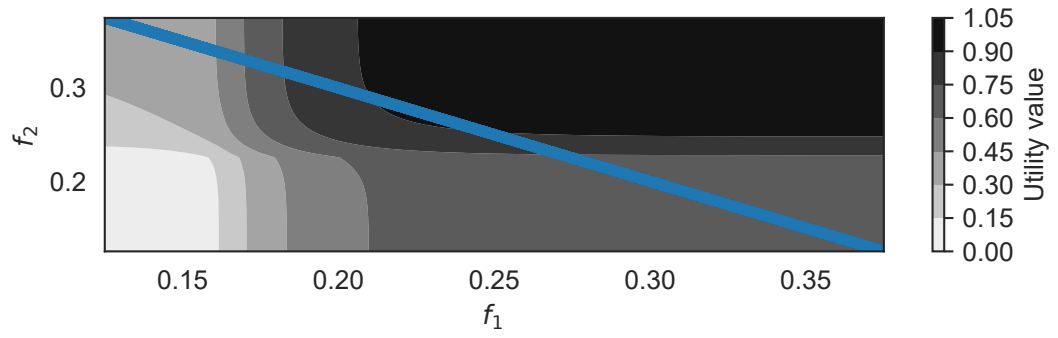


Figure A.21. Heatmap vs. Pf : DTLZ1, DM type 3

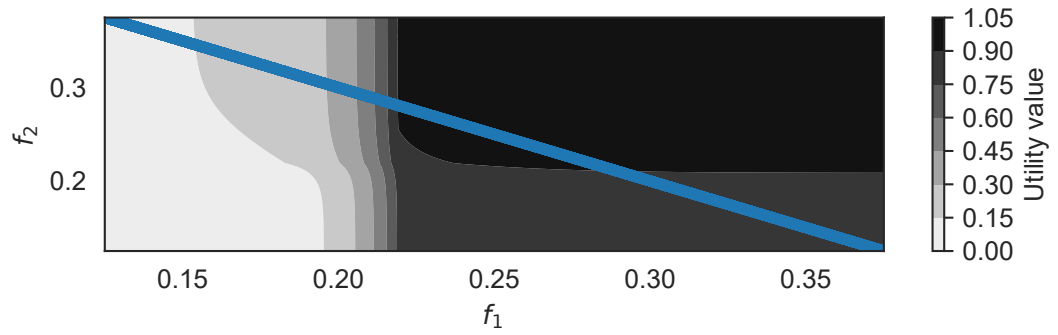


Figure A.22. Heatmap vs. Pf : DTLZ1, DM type 4

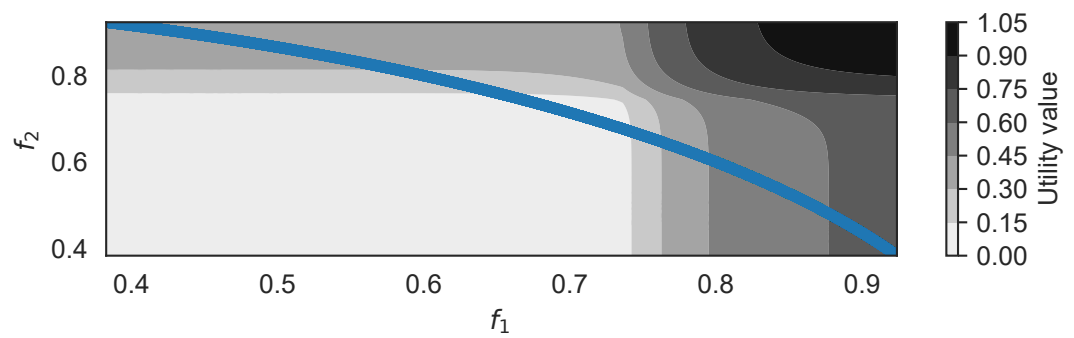


Figure A.23. Heatmap vs. Pf : DTLZ2, DM type 1

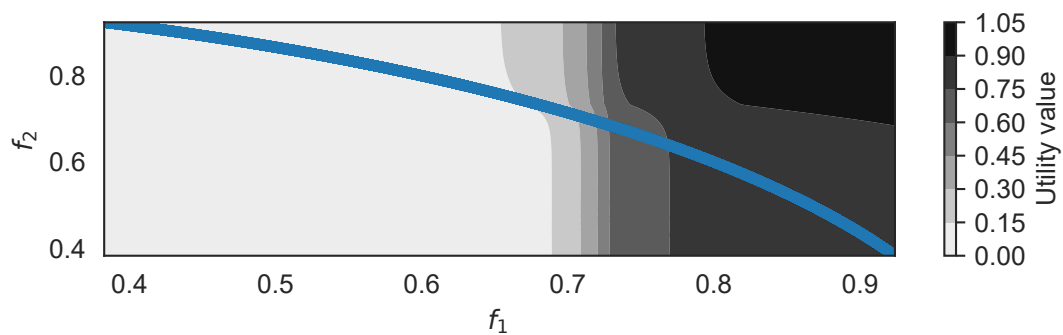


Figure A.24. Heatmap vs. Pf : DTLZ2, DM type 2

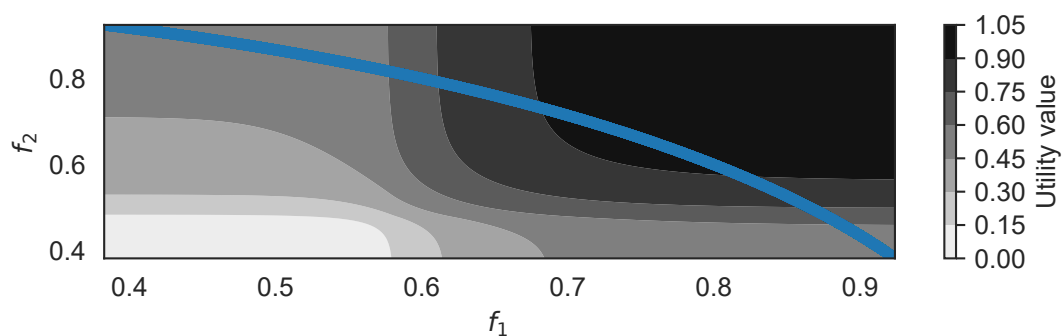


Figure A.25. Heatmap vs. Pf : DTLZ2, DM type 3

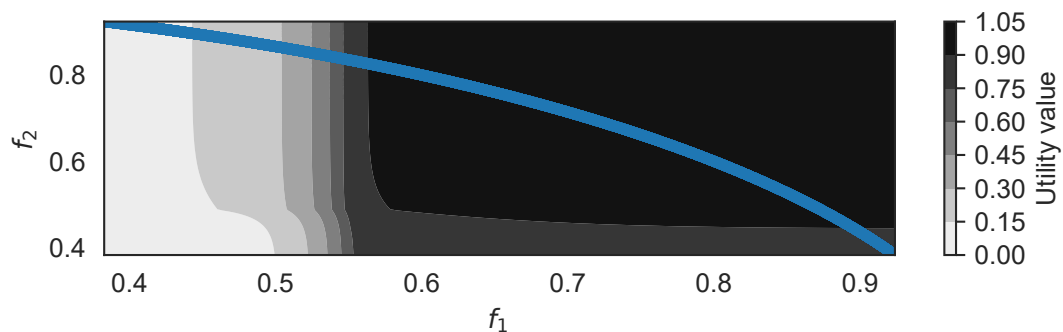


Figure A.26. Heatmap vs. Pf : DTLZ2, DM type 4

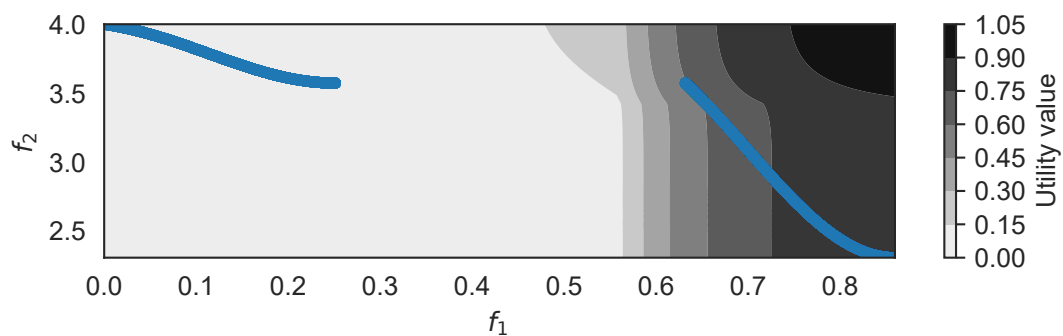


Figure A.27. Heatmap vs. Pf : DTLZ7, DM type 1

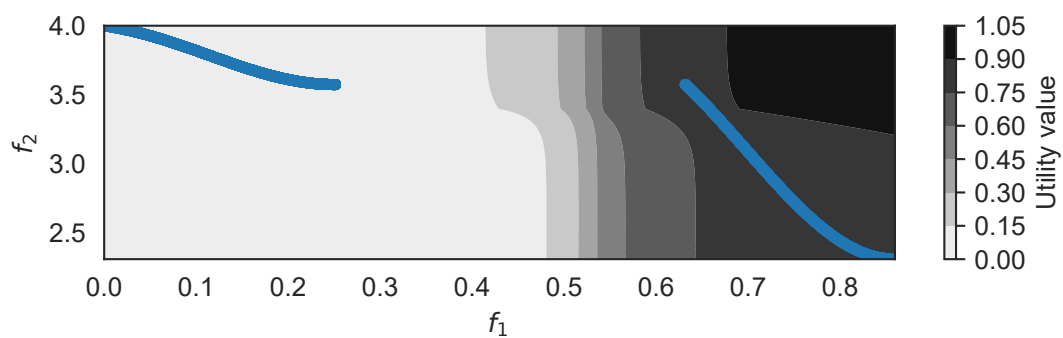


Figure A.28. Heatmap vs. Pf : DTLZ7, DM type 2

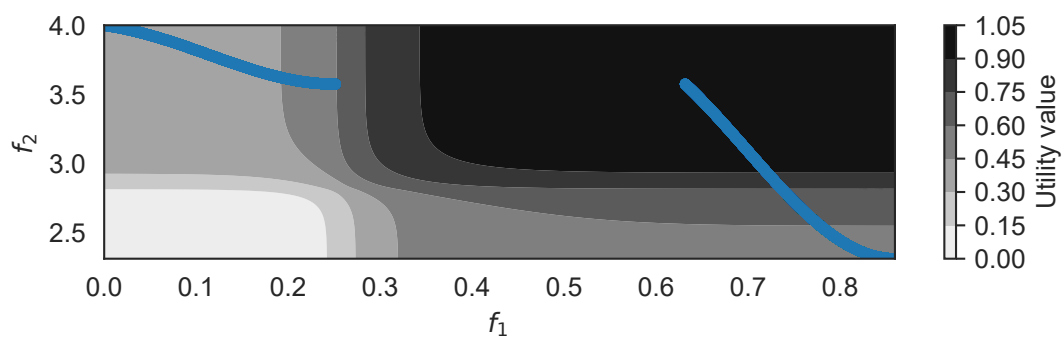


Figure A.29. Heatmap vs. Pf : DTLZ7, DM type 3

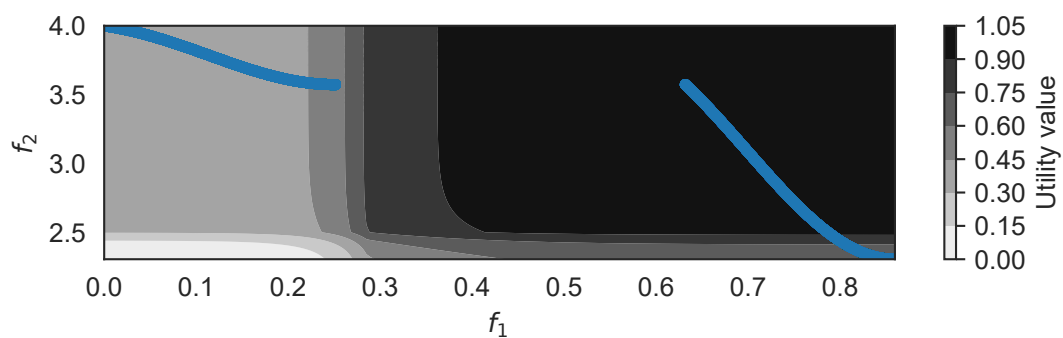


Figure A.30. Heatmap vs. Pf : DTLZ7, DM type 4

Appendix B

Supplementary Material to Chapter 4: Detecting Hidden and Irrelevant Objectives in Interactive Multi-Objective Optimization

This supplementary material provides Figures and Tables that were not included in the main text of Chapter 4 for brevity.

B.1 Additional results

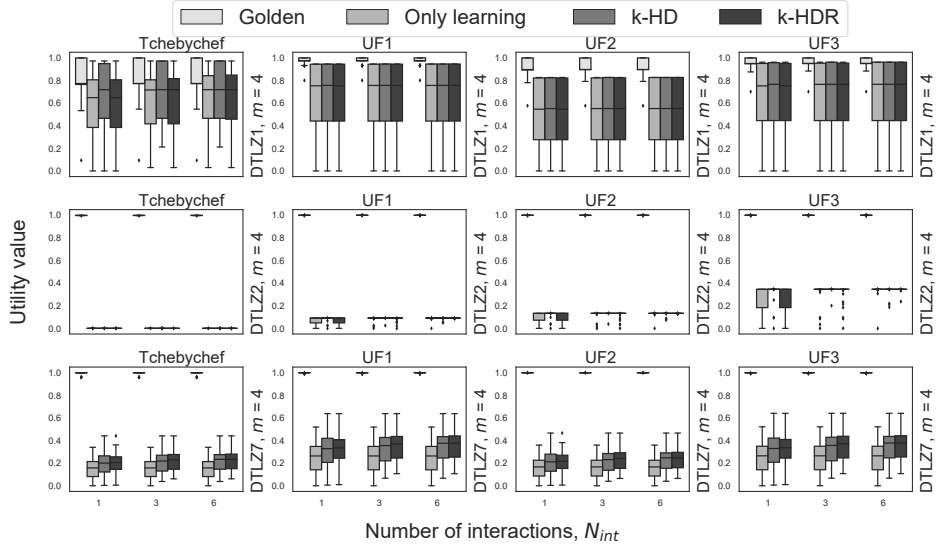


Figure B.1. Comparison of the performance of different modes for DTLZ problems with $m = 4$. The number of active objectives are fixed.

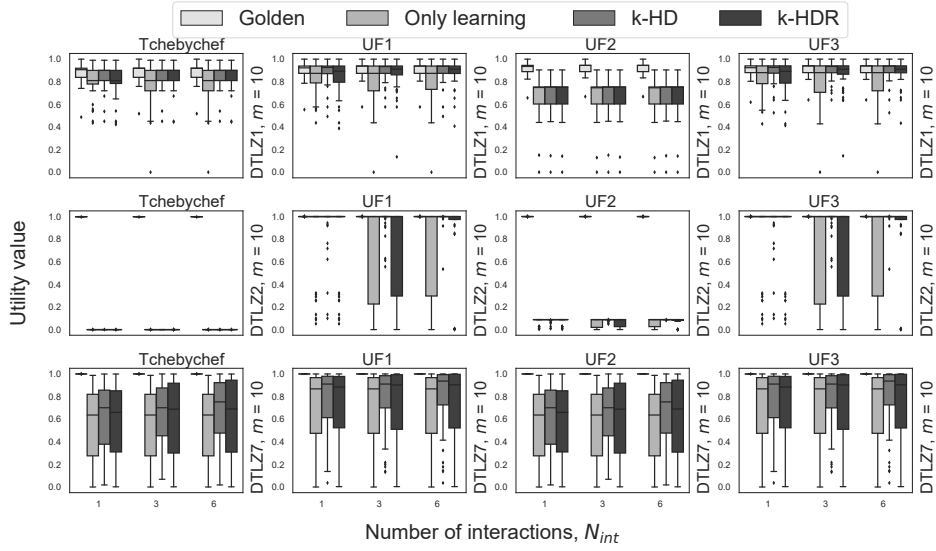


Figure B.2. Comparison of the performance of different modes for DTLZ problems with $m = 10$. The number of active objectives are fixed.

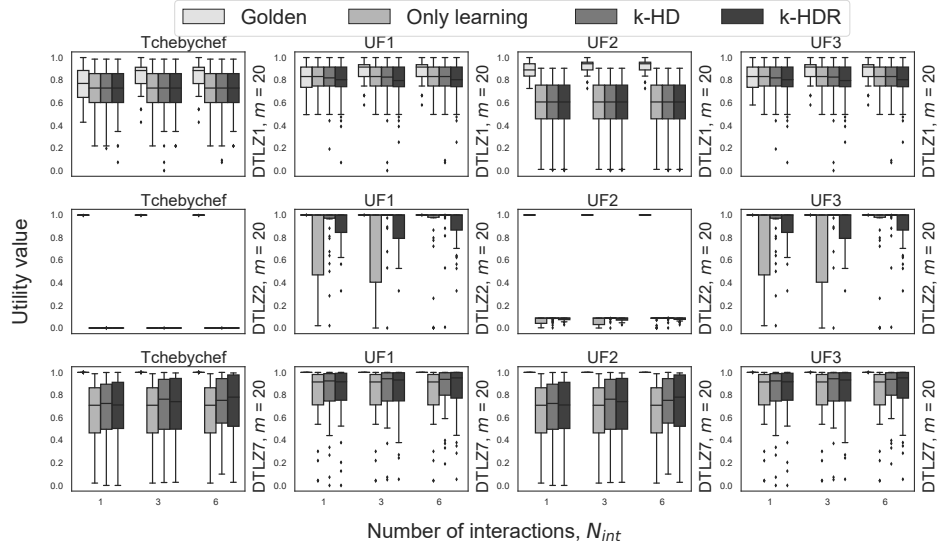


Figure B.3. Comparison of the performance of different modes for DTLZ problems with $m = 20$. The number of active objectives are fixed.

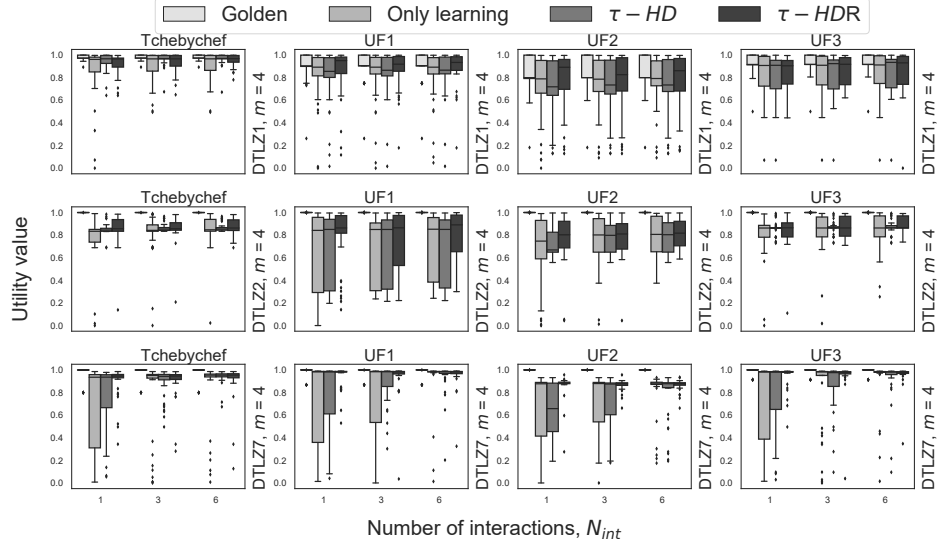


Figure B.4. Comparison of the performance of different modes for DTLZ problems with $m = 4$. The number of active objectives is not fixed and detection mode is used as a mean of objective reduction technique.

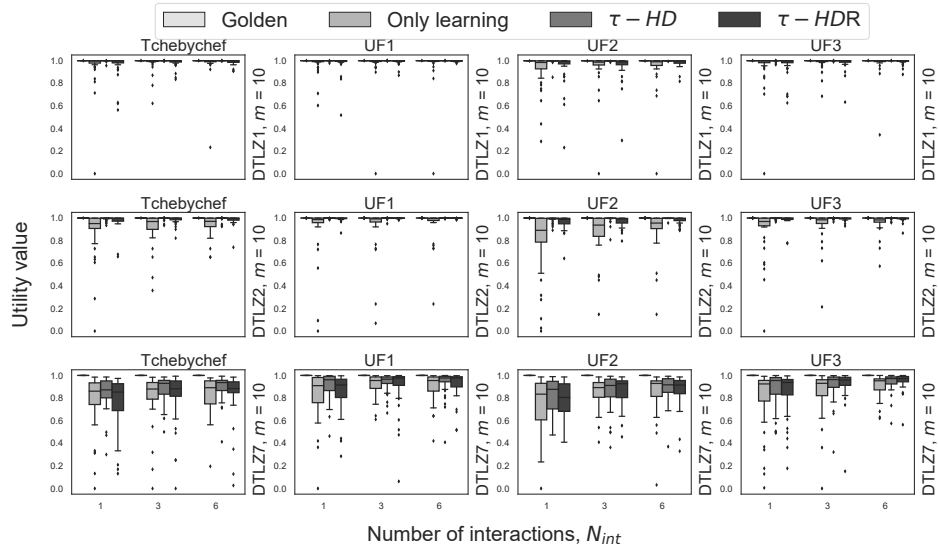


Figure B.5. Comparison of the performance of different modes for DTLZ problems with $m = 10$. The number of active objectives is not fixed and detection mode is used as a mean of objective reduction technique.

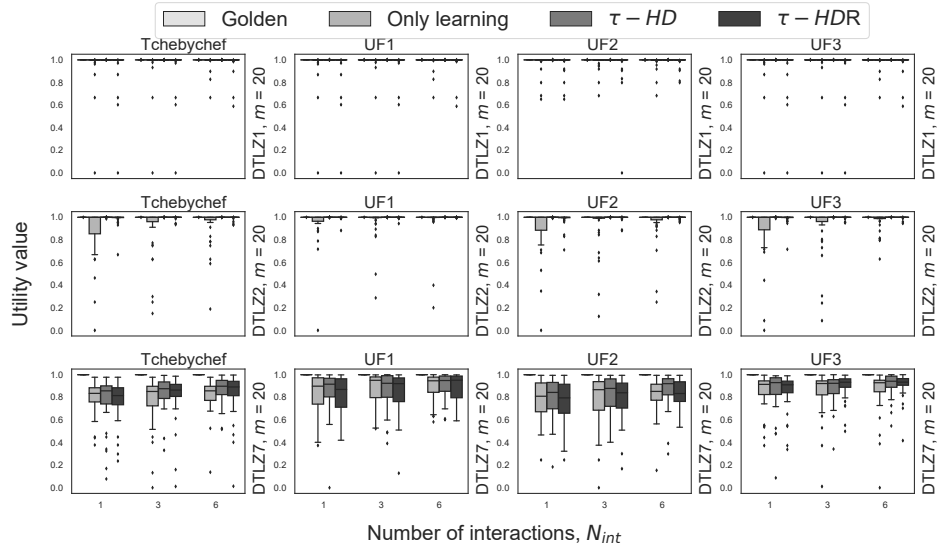


Figure B.6. Comparison of the performance of different modes for DTLZ problems with $m = 20$. The number of active objectives is not fixed and detection mode is used as a mean of objective reduction technique.

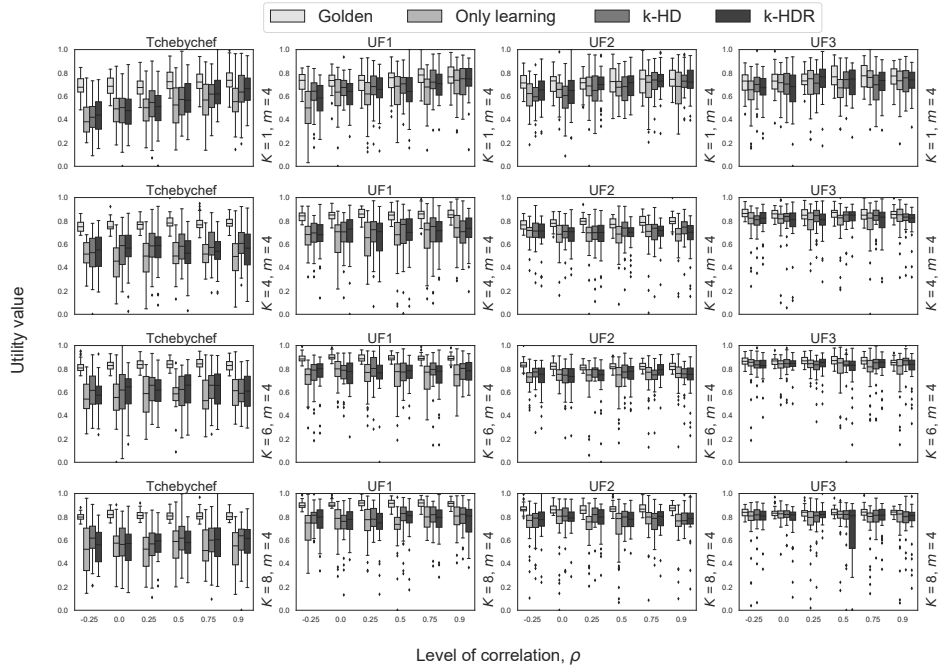


Figure B.7. Comparison of the performance of different modes for ρ MKN problems with $m = 4$. Number of active objectives is fixed.

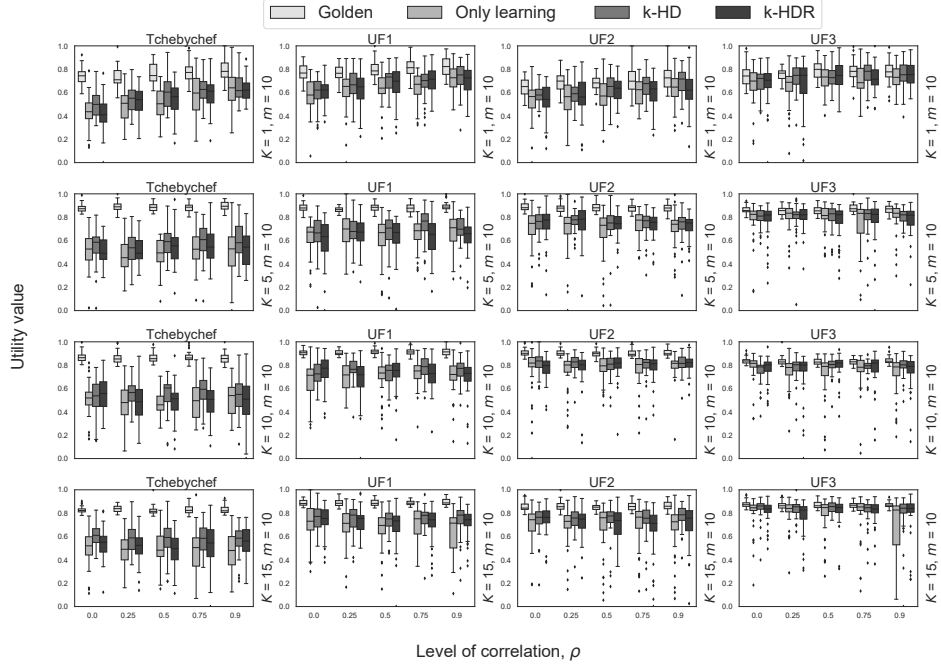


Figure B.8. Comparison of the performance of different modes for ρ MKN problems with $m = 10$. Number of active objectives is fixed.

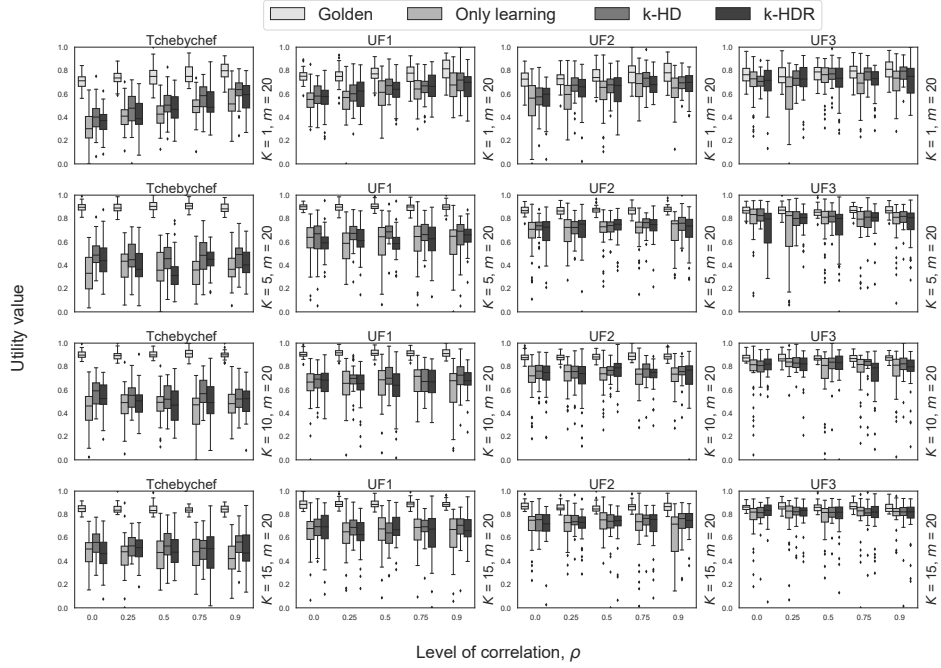


Figure B.9. Comparison of the performance of different modes for ρ MKN problems with $m = 20$. Number of active objectives is fixed.

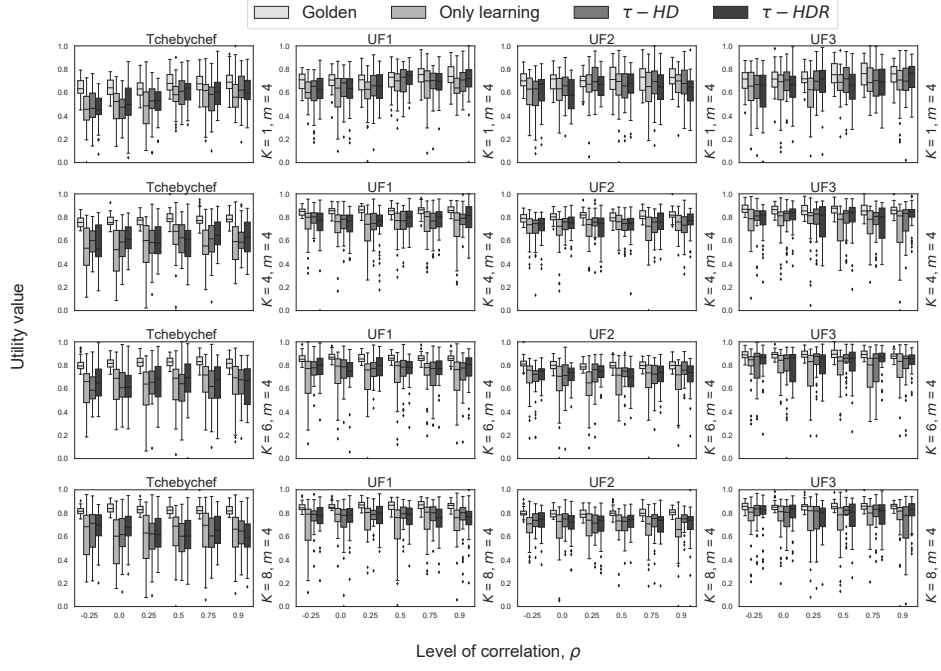


Figure B.10. Comparison of the performance of different modes for ρ MKN problems with $m = 4$. The number of active objectives is not fixed and detection mode is used as a mean of objective reduction technique.

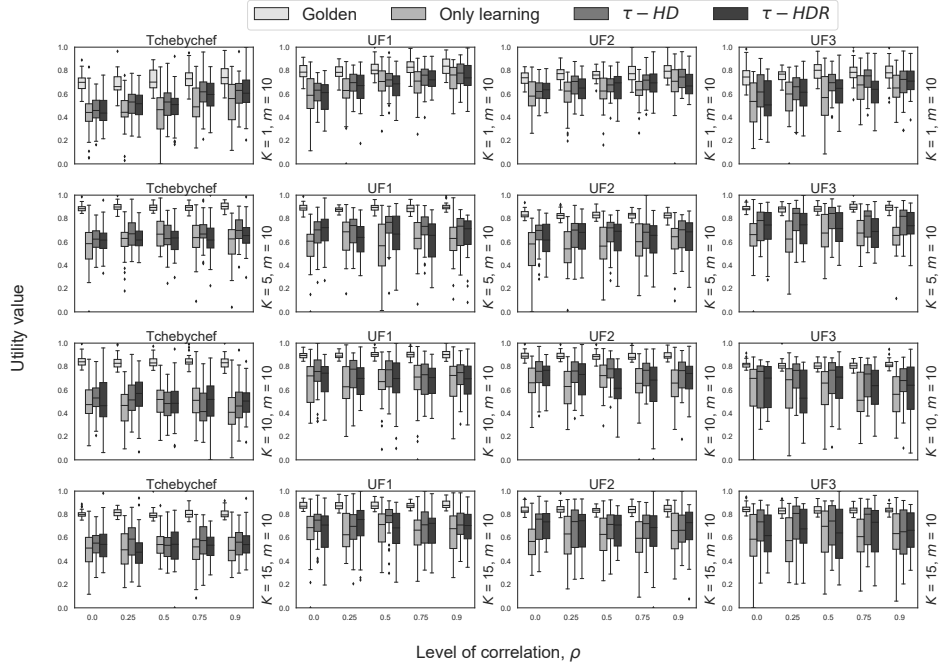


Figure B.11. Comparison of the performance of different modes for ρ MKN problems with $m = 4$. The number of active objectives is not fixed and detection mode is used as a mean of objective reduction technique.

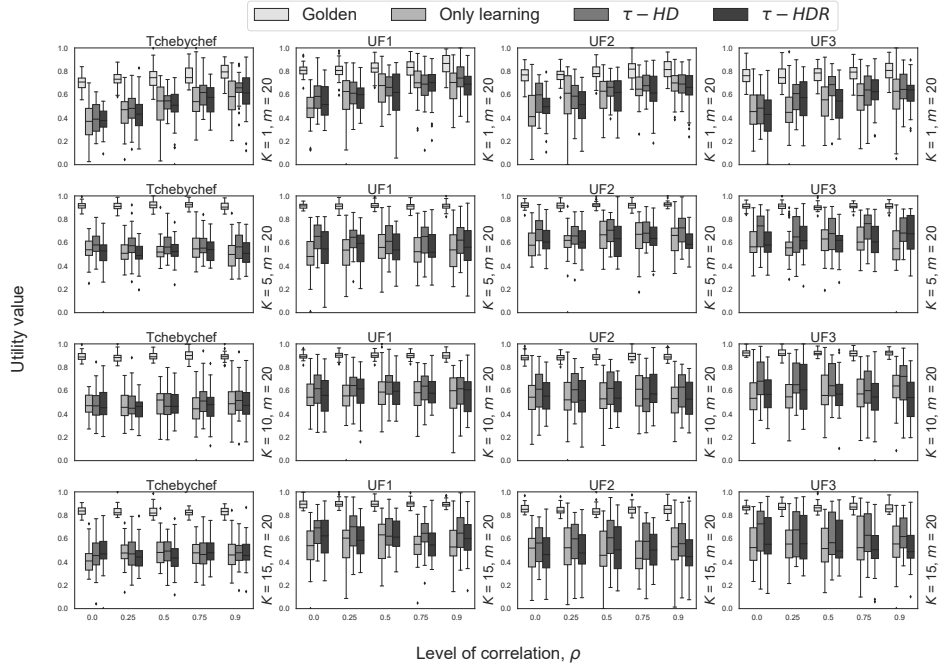


Figure B.12. Comparison of the performance of different modes for ρ MNK problems with $m = 20$. The number of active objectives is not fixed and detection mode is used as a mean of objective reduction technique.

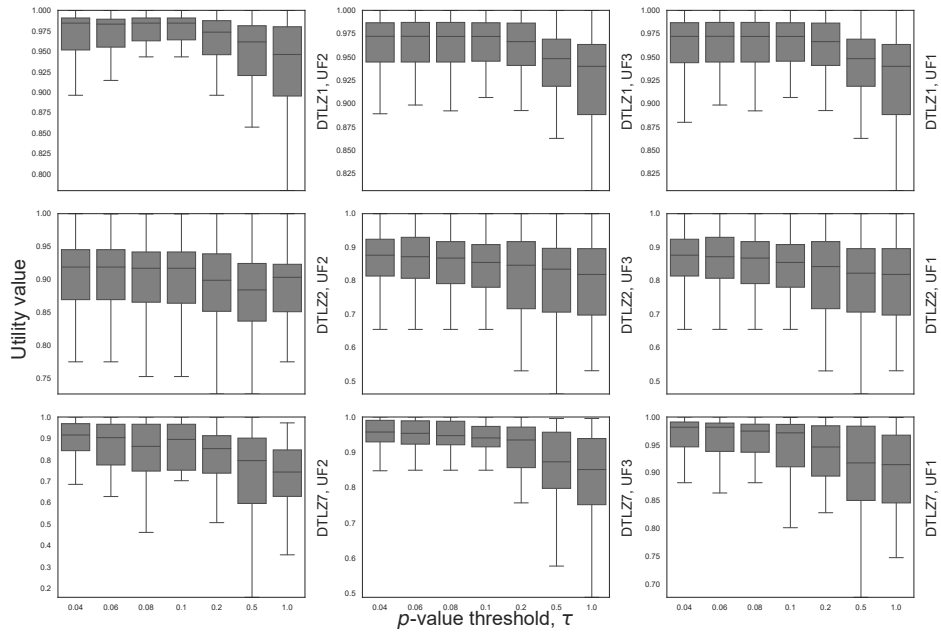


Figure B.13. Performance analysis of τ -HDR with different values of τ for DTLZ problems with $m = 20$. Number of interaction in all runs is fixed to 6.

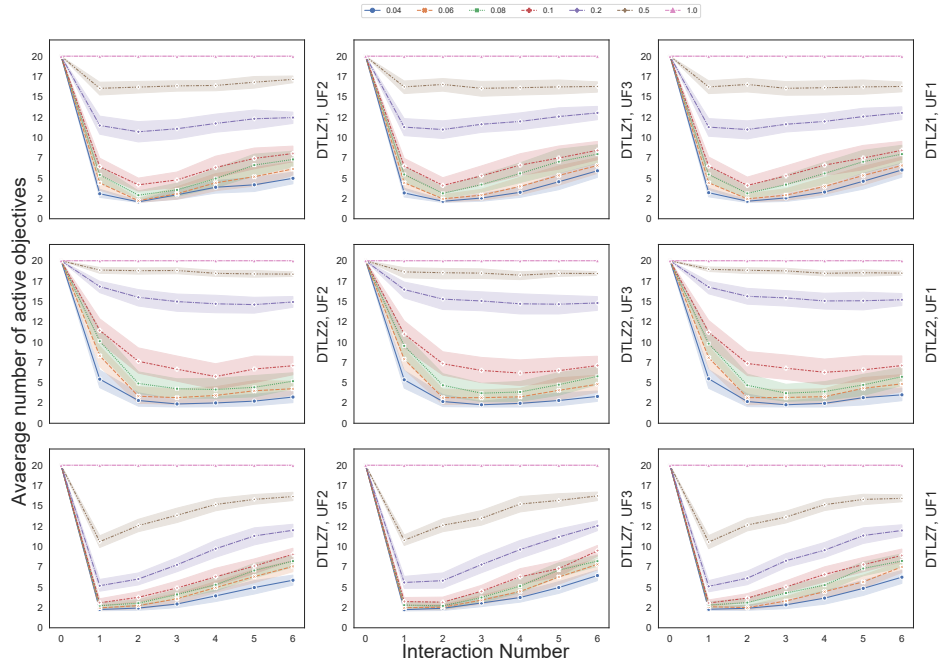


Figure B.14. Number of active objectives after each interaction for different values of τ in τ -HDR mode on DTLZ test problems.

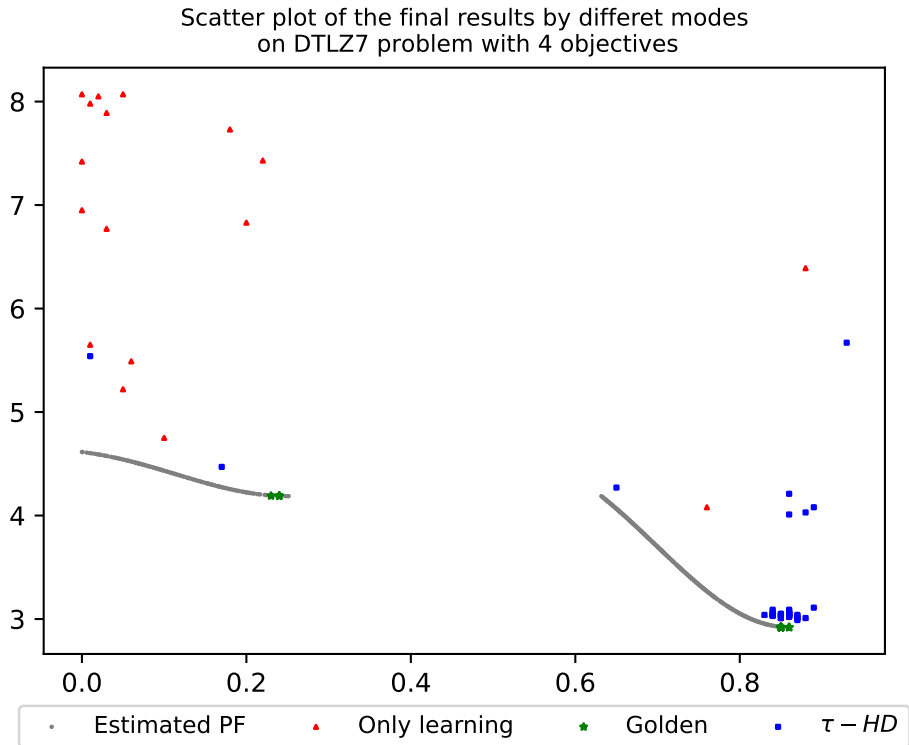


Figure B.15. Location of the returned solutions with different modes compared to golden mode. The horizontal axis indicates the first relevant objective (f_1) and the vertical axis indicates the fourth relevant objective (f_2). The results returned by the τ -HD are close to those returned by Golden mode as can be seen on the bottom right side of the Figure. The solutions by the only learning mode are scattered away from the Golden mode.

B.2 F-test vs. mutual information

In case the normality of the data is not known, the alternative to f-test criteria is the non-parametric *mutual information* metric that can be used for uni-variate feature selection. In our benchmark problems the normality assumptions are verified. However, we ran experiments to compare the accuracy of both methods. In this regard, we have analyzed 5000 data samples to ensure the F-regression test is a better option in our case and the results are laid out in Table B.1 below. The table also indicates that the method can detect the set of important objectives even with a small sample of solutions, and the accuracy increases significantly when the sample size is increased to 10 and 15 solutions.

Table B.1. Comparing the accuracy of F-regression test and mutual information in feature selection.

Method	Size of the training set	Mean Accuracy	Standard Deviation
F-regression	5	71.89	28.48
Mutual information	5	61.99	27.46
F-regression	10	87.8	22.16
Mutual information	10	79.92	26.51
F-regression	15	93.95	16.64
Mutual information	15	88.81	21.52

B.3 Weights of Utility Function in the Experiments

Table B.2 indicates the weights of the active objectives for each set of test problems for different utility functions.

Table B.2. Weights of active objectives used in the Tchebychef function and shape of the UF1-UF3 used in experiments.

Problem	UF	Weight Vector
DTLZ1	Tchebychef	[0.97, 0.03]
DTLZ2	Tchebychef	[0.79835788, 0.20164212]
DTLZ7	Tchebychef	[0.80588792, 0.19411208]
ρ MNK	Tchebychef	[0.5, 0.5]
All DTLZ	UF1	$0.28x_1^2 + 0.29x_1x_4 + 0.38x_4^2 + 0.05x_1$
All DTLZ	UF2	$0.05x_1x_4 + 0.6x_1^2 + 0.38x_4 + 0.23x_1$
All DTLZ	UF3	$0.44x_1^2 + 0.33x_1 + 0.09x_1x_4 + 0.14x_4^2$
All ρ MNK	UF1	$0.28x_1^2 + 0.29x_1x_2 + 0.38x_2^2 + 0.05x_1$
All ρ MNK	UF2	$0.05x_1x_2 + 0.6x_1^2 + 0.38x_2 + 0.23x_1$
All ρ MNK	UF3	$0.44x_1^2 + 0.33x_1 + 0.09x_1x_2 + 0.14x_2^2$

B.4 Comparison of different modes for Figures 3 and 4 in the paper

Table B.3. Comparison of *Only learning* Mode with detection of irrelevant objectives approach in experiments depicted in Fig 4.3. The winner indicates the algorithm with better performance, and p -value is highlighted in the boldface when such a difference is significant (p -value<0.05). Mode 1 and Mode 2: the modes that are being compared. ρ : The correlation among objectives in ρ MNK problem. m : Number of objective functions. winner: the mode with superior performance. p -value: the significance level of the differences.

Mode 1	Mode 2	ρ	m	UF	winner	p -value
Only learning	k -HD	0.25	4	TchebyChef	k -HD	0.080573412
Only learning	k -HD	0.25	4	TchebyChef	k -HD	0.166219599
Only learning	k -HDR	0.25	4	TchebyChef	k -HDR	0.085344126
Only learning	k -HD	0.5	4	TchebyChef	k -HD	0.258868738
Only learning	k -HD	0.75	4	TchebyChef	k -HD	0.080573412
Only learning	k -HDR	0.75	4	TchebyChef	k -HDR	0.112722732
Only learning	k -HD	0.9	4	TchebyChef	k -HD	0.065554509
Only learning	k -HDR	0.9	4	TchebyChef	k -HDR	0.170371114
Only learning	k -HD	0	20	UF3	k -HD	0.667107251
Only learning	k -HD	0.25	20	UF3	k -HD	0.003536861
Only learning	k -HDR	0.25	20	UF3	k -HDR	0.025664219
Only learning	k -HD	0.5	20	UF3	k -HD	0.36781863
Only learning	k -HD	0.75	20	UF3	k -HD	0.001444604
Only learning	k -HD	0.9	20	UF3	k -HD	0.01443244

Table B.4. Comparison of *Only learning* Mode with detection of irrelevant objectives approach in experiments depicted in Fig 4.4. The winner indicates the algorithm with better performance, and p -value is highlighted in the boldface when such a difference is significant (p -value<0.05). Mode 1 and Mode 2: the modes that are being compared. ρ : The correlation among objectives in ρ MNK problem. m : Number of objective functions. winner: the mode with superior performance. p -value: the significance level of the differences.

Mode 1	Mode 2	ρ	m	UF	winner	p -value
Only learning	τ -HD	0	20	UF1	τ -HD	0.00109
Only learning	τ -HDR	0	20	UF1	τ -HDR	0.045204
Only learning	τ -HD	0.25	20	UF1	τ -HD	0.036008
Only learning	τ -HD	0.5	20	UF1	τ -HD	0.178906
Only learning	τ -HD	0.75	20	UF1	τ -HD	0.004197
Only learning	τ -HD	0.9	20	UF1	τ -HD	0.039733
Only learning	τ -HD	0	4	UF2	τ -HD	0.082931
Only learning	τ -HDR	0	4	UF2	τ -HDR	0.109706
Only learning	τ -HD	0.25	4	UF2	τ -HD	0.04378
Only learning	τ -HDR	0.5	4	UF2	Only learning	0.276266
Only learning	τ -HD	0.75	4	UF2	τ -HD	0.404641
Only learning	τ -HDR	0.75	4	UF2	τ -HDR	0.098274
Only learning	τ -HD	0.9	4	UF2	Only learning	0.294445

Appendix C

Supplementary Material to Chapter 5:

Decision-Tree Based interactive

Evolutionary Multi-Objective

Optimization Algorithm

This supplementary material provides all the figures required for the analysis of the Chapter 5 that had been omitted for brevity.

C.1 Samples & Experimental Data

Table C.1. A sample training set acquired from an experiment on DTLZ1 with 2 objectives

	Objective 1	Objective 2	Rank
1	0.224231	0.276240	1
2	0.342230	0.158252	3
3	0.140248	0.360225	4
4	0.270838	0.229634	0
5	0.164741	0.335731	2

Table C.2. Examples used for building the tree constructed using the data in Table C.1

	1	2	Label
1	-0.12	0.12	1
2	0.08	-0.08	1
3	-0.05	0.05	-1
4	0.06	-0.06	1
5	0.20	-0.20	1
6	0.07	-0.07	-1
7	0.18	-0.18	-1
8	-0.13	0.13	-1
9	-0.02	0.02	-1
10	0.11	-0.11	1

Table C.3. Specifications of the tests. **P** indicates the benchmark problem. Type corresponds to the DM types in Table 3.1. n : dimension of the problem. m : dimension of the objective space. $U(\mathbf{z}^{\text{MPS}})$: utility of \mathbf{z}^{MPS} . $U(\mathbf{z}^w)$: utility of the worst PF solution. UF column specifies the utility function used in the test where “tst” stands for the Stewart UF and “tch” for the Tchebychef.

Test	P	n	m	Type	$U(\mathbf{z}^{\text{MPS}})$	$U(\mathbf{z}^w)$	UF
1	1	6	2	1	0.1023	0.8976	tst
2	1	6	2	2	0.0101	0.9038	tst
3	1	6	2	3	0.3128	0.9970	tst
4	1	6	2	4	0.1426	0.9378	tst
5	1	6	2	-	0.1159	0.3476	tch
6	1	14	10	1	0.0002	0.2164	tst
7	1	14	10	2	0.0001	0.1609	tst
8	1	14	10	3	0.0215	0.8690	tst
9	1	14	10	4	0.0108	0.7716	tst
10	1	14	10	-	0.0397	0.2404	tch
11	2	11	2	1	0.3660	0.6364	tst
12	2	11	2	2	0.1062	0.8933	tst
13	2	11	2	3	0.4627	0.9996	tst
14	2	11	2	4	0.1261	0.9977	tst
15	2	11	2	-	0.3170	0.7653	tch
16	2	19	10	1	0.0003	0.2953	tst
17	2	19	10	2	0.0001	0.2448	tst
18	2	19	10	3	0.1117	0.9695	tst
19	2	19	10	4	0.0217	0.8263	tst
20	2	19	10	-	0.0675	0.2446	tch
21	7	21	2	1	0.1259	0.9692	tst
22	7	21	2	2	0.1097	0.8894	tst
23	7	21	2	3	0.4061	1.0000	tst
24	7	21	2	4	0.3525	0.9999	tst
25	7	21	2	-	0.0402	0.8506	tch
26	7	29	10	1	0.0016	0.8045	tst
27	7	29	10	2	0.0007	0.7638	tst
28	7	29	10	3	0.1238	0.9927	tst
29	7	29	10	4	0.0460	0.9842	tst
30	7	29	10	-	0.0580	0.2113	tch
31	1	8	4	1	0.3560	0.8928	tst
32	1	8	4	2	0.0047	0.3643	tst
33	1	8	4	3	0.3019	0.8759	tst
34	1	8	4	4	0.0047	0.3781	tst
35	1	8	4	-	0.0405	0.1589	tch
36	2	13	4	1	0.3267	0.9936	tst
37	2	13	4	2	0.0253	0.5428	tst
38	2	13	4	3	0.4150	0.9989	tst
39	2	13	4	4	0.0214	0.5193	tst
40	2	13	4	-	0.0567	0.3222	tch
41	7	23	4	1	0.1481	0.9993	tst
42	7	23	4	2	0.0111	0.6991	tst
43	7	23	4	3	0.3269	0.9992	tst
44	7	23	4	4	0.0228	0.7759	tst
45	7	23	4	-	0.0699	0.4371	tch

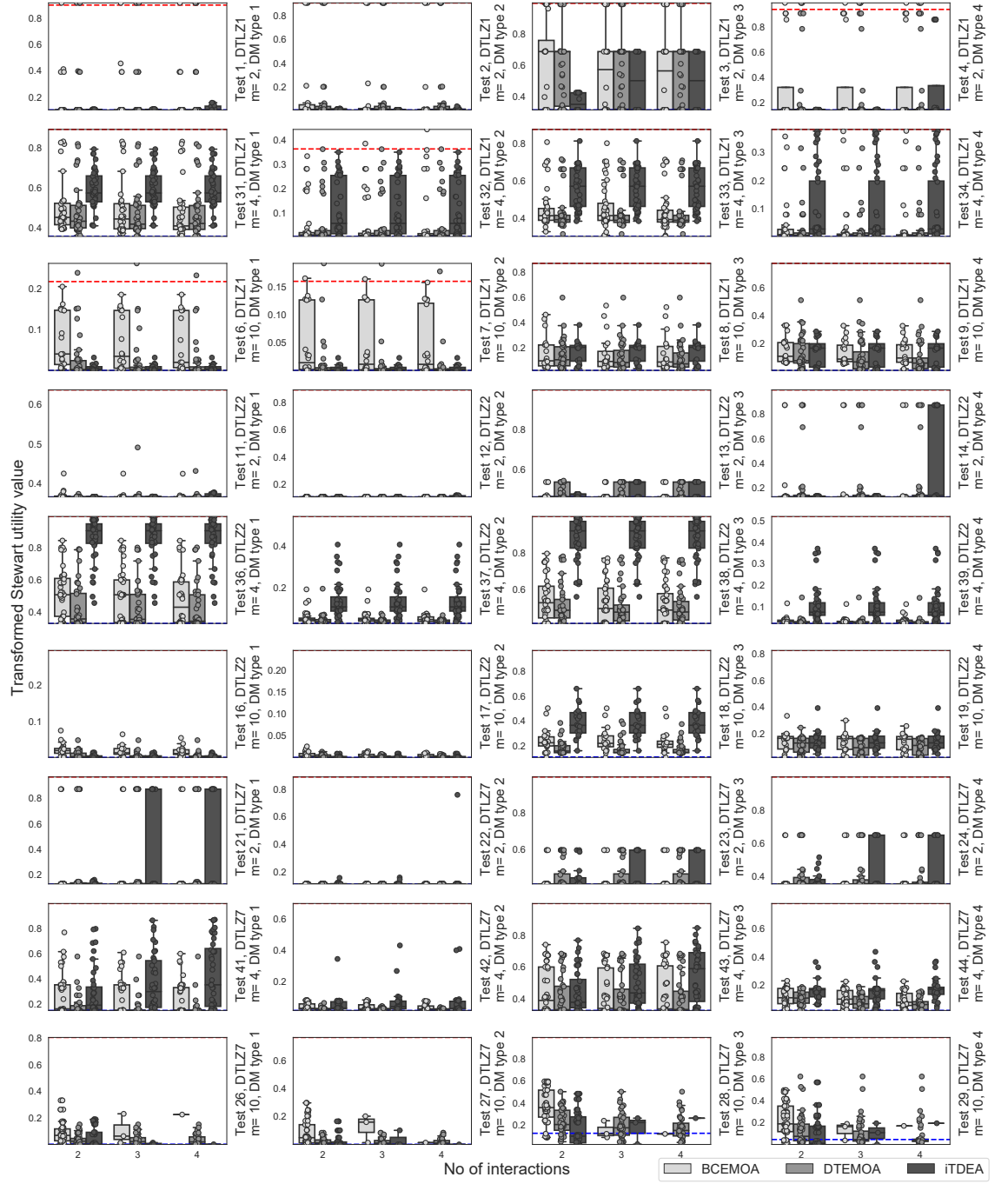


Figure C.1. Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA under ideal conditions with Stewart UF.

C.2 Results

Table C.4. The results of experiments under ideal conditions for BCEMOA, DTEMOA and iTDEA. The mean (and standard deviation in parenthesis) over 20 independent runs is shown for each test. The values are rounded to 2 decimal points. The performance of DTEMOA is compared with iTDEA and BCEMOA and the better values are indicated with letter 'i' or 'b' (indicating iTDEA or BCEMOA) to highlight the winner where $p\text{-value} \leq .05$ for Wilcoxon test.

Test	BCEMOA			DTEMOA			iTDEA		
N_{int}	2	3	4	2	3	4	2	3	4
Sigmoid									
1	.19(.2)	.19(.2)	.19(.2)	.18(.22)b	.18(.22)b	.18(.22)b	.1(.0)i	.1(.0)i	.12(.02)i
2	.16(.32)	.16(.33)	.15(.33)	.12(.27)	.12(.27)	.12(.27)b	.01(.0)	.01(.0)i	.02(.01)i
3	.61(.27)	.57(.27)	.57(.27)	.6(.23)	.59(.24)	.59(.24)	.37(.06)i	.5(.19)	.5(.19)
4	.34(.34)	.34(.34)	.34(.34)	.28(.3)	.28(.3)	.28(.3)i	.14(.0)i	.14(.0)i	.32(.32)
6	.07(.07)	.07(.07)	.06(.07)	.03(.05)b	.02(.05)b	.02(.05)b	.01(.01)	.01(.01)	.01(.01)
7	.05(.06)	.05(.06)	.05(.06)	.01(.04)b	.01(.03)b	.01(.03)b	.0(.01)	.0(.01)	.0(.01)
8	.16(.13)	.15(.14)	.15(.14)	.14(.11)	.13(.12)	.12(.11)i	.17(.09)	.17(.09)	.17(.09)
9	.15(.09)	.14(.08)	.14(.08)	.13(.11)	.11(.12)b	.1(.11)b	.15(.1)	.15(.1)	.15(.1)
11	.37(.01)	.37(.01)	.37(.01)	.37(.0)i	.37(.02)	.37(.01)i	.37(.0)	.37(.0)i	.37(.0)
12	.11(.0)	.11(.0)	.11(.0)	.11(.0)i	.11(.0)i	.11(.0)i	.11(.0)	.11(.0)	.11(.0)
13	.47(.03)b	.47(.03)b	.47(.03)b	.49(.03)	.49(.03)i	.49(.03)i	.47(.01)	.5(.04)	.5(.04)
14	.21(.23)	.2(.23)	.2(.23)b	.22(.24)	.22(.24)	.22(.24)i	.13(.0)	.13(.0)	.46(.38)
16	.02(.02)	.02(.01)	.02(.01)	.01(.01)b	.0(.01)b	.0(.01)b	.0(.0)	.0(.0)	.0(.0)
17	.01(.01)	.0(.0)	.0(.0)	.0(.0)b	.0(.0)b	.0(.0)b	.0(.0)	.0(.0)	.0(.0)
18	.25(.1)	.25(.1)	.23(.09)	.18(.05)bi	.18(.06)bi	.17(.05)bi	.39(.12)	.39(.12)	.39(.12)
19	.14(.07)	.14(.07)	.13(.06)	.12(.06)	.1(.06)bi	.1(.06)bi	.15(.08)	.15(.08)	.15(.08)
21	.2(.23)	.2(.23)	.2(.23)	.24(.27)	.24(.27)	.24(.27)	.13(.01)	.39(.37)	.39(.37)
22	.11(.0)	.11(.0)	.11(.0)	.11(.0)bi	.11(.0)bi	.11(.0)bi	.11(.01)	.11(.01)	.14(.15)
23	.43(.07)b	.43(.07)b	.43(.07)b	.45(.08)	.45(.08)	.45(.08)	.44(.07)	.48(.09)	.48(.09)
24	.38(.09)b	.38(.09)b	.38(.09)b	.41(.11)	.4(.11)	.4(.11)	.38(.05)	.46(.14)	.46(.14)
26	.09(.09)	.1(.11)	.22(.0)	.04(.04)bi	.04(.04)	.04(.04)	.05(.07)	.01(.01)	.0(.0)
27	.08(.09)	.12(.1)	.01(.0)	.02(.02)bi	.02(.02)b	.02(.02)	.02(.05)	.03(.06)	.0(.0)
28	.36(.16)	.15(.08)	.12(.0)	.23(.12)b	.19(.12)	.17(.11)	.17(.15)i	.17(.13)	.26(.0)
29	.23(.15)	.13(.08)	.17(.0)	.14(.15)b	.1(.13)	.09(.13)	.12(.16)i	.1(.1)	.2(.0)
31	.49(.12)	.48(.13)	.47(.13)	.48(.12)i	.47(.12)i	.47(.12)i	.59(.09)	.59(.09)	.59(.09)
32	.05(.09)	.05(.09)	.05(.11)	.05(.1)i	.05(.1)i	.05(.1)i	.12(.12)	.12(.12)	.12(.12)
33	.45(.11)	.46(.12)	.43(.11)	.43(.11)bi	.43(.11)i	.43(.1)i	.56(.11)	.56(.11)	.56(.11)
34	.05(.09)	.04(.09)	.04(.09)	.03(.06)i	.03(.06)i	.03(.06)i	.1(.13)	.1(.13)	.1(.13)
36	.52(.15)	.49(.17)	.47(.16)	.43(.14)bi	.43(.14)i	.42(.14)i	.85(.14)	.85(.14)	.85(.14)
37	.05(.03)	.05(.03)	.05(.03)	.04(.03)bi	.04(.01)bi	.04(.01)bi	.13(.09)	.13(.09)	.13(.09)
38	.55(.12)	.54(.12)	.53(.11)	.52(.11)i	.5(.1)i	.51(.1)i	.88(.12)	.88(.12)	.88(.12)
39	.03(.01)	.03(.01)	.03(.02)	.03(.02)bi	.03(.02)i	.03(.02)bi	.11(.09)	.11(.09)	.11(.09)
41	.28(.16)	.26(.14)	.25(.13)	.18(.08)bi	.17(.08)bi	.17(.07)i	.29(.19)	.37(.22)	.43(.26)
42	.03(.02)	.03(.02)	.03(.02)	.02(.01)bi	.02(.01)i	.02(.0)i	.05(.05)	.06(.07)	.06(.08)
43	.45(.15)	.44(.14)	.44(.15)	.42(.13)	.42(.13)i	.42(.12)i	.44(.13)	.5(.15)	.54(.17)
44	.12(.06)	.11(.05)	.1(.05)	.11(.05)i	.08(.05)bi	.07(.04)bi	.16(.06)	.15(.08)	.17(.07)
Tchebychef									
5	.14(.05)	.14(.05)	.14(.05)	.13(.04)	.13(.04)	.13(.04)i	.12(.01)i	.12(.01)i	.14(.03)
10	.06(.03)	.06(.03)	.06(.03)	.05(.02)i	.05(.02)i	.05(.02)bi	.06(.03)	.06(.03)	.06(.03)
15	.32(.0)	.32(.0)	.32(.0)	.32(.0)i	.32(.0)i	.32(.0)i	.33(.01)	.33(.01)	.37(.05)
20	.08(.01)	.08(.01)	.08(.01)	.07(.01)bi	.07(.01)bi	.07(.01)bi	.11(.03)	.11(.03)	.11(.03)
25	.04(.0)	.05(.05)	.04(.01)	.04(.0)i	.04(.0)i	.04(.0)i	.1(.1)	.04(.0)	.04(.0)
30	.12(.01)	.12(.02)	.12(.0)	.1(.01)b	.1(.01)b	.1(.01)b	.06(.03)i	.06(.02)i	.05(.02)i
35	.06(.02)	.06(.02)	.05(.02)	.05(.02)bi	.05(.02)i	.05(.02)i	.07(.03)	.07(.03)	.07(.03)
40	.08(.04)	.08(.03)	.08(.03)	.07(.02)i	.07(.02)i	.07(.02)i	.17(.06)	.17(.06)	.17(.06)
45	.1(.04)	.1(.04)	.1(.04)	.09(.02)i	.08(.02)i	.08(.01)i	.11(.06)	.13(.08)	.15(.08)

Table C.5. The results of experiments with simulation of inconsistencies in DM's decisions for BCEMOA, DTEMOA and iTDEA. The mean (and standard deviation in parenthesis) over 20 independent runs is shown for each test. The values are rounded to 2 decimal points. The performance of DTEMOA is compared with iTDEA and BCEMOA and the better values are indicated with letter 'i' or 'b' (indicating iTDEA or BCEMOA) to highlight the winner where $p\text{-value} \leq .05$ for Wilcoxon test.

Test	BCEMOA			DTEMOA			iTDEA		
	$\sigma = 0.005$	$\sigma = 0.01$	$\sigma = 0.2$	$\sigma = 0.005$	$\sigma = 0.01$	$\sigma = 0.2$	$\sigma = 0.005$	$\sigma = 0.01$	$\sigma = 0.2$
Sigmoid									
1	.25(.32)	.25(.32)	.28(.32)	.2(.25)b	.19(.23)b	.16(.16)b	.1(.0)i	.1(.0)i	.15(.07)i
2	.13(.28)	.13(.28)	.22(.35)	.1(.24)	.1(.24)	.07(.17)b	.01(.0)i	.01(.0)i	.04(.05)
3	.54(.28)	.57(.27)	.66(.26)	.62(.22)	.62(.22)	.58(.24)	.49(.19)	.49(.19)	.51(.21)
4	.35(.36)	.35(.36)	.37(.36)	.28(.3)b	.28(.3)b	.28(.3)b	.14(.0)i	.15(.0)i	.16(.03)i
6	.09(.08)	.07(.07)	.12(.1)	.03(.06)b	.02(.05)b	.03(.06)b	.01(.01)	.01(.01)	.01(.01)
7	.05(.06)	.04(.06)	.1(.09)	.01(.04)b	.01(.04)b	.01(.03)b	.0(.01)	.0(.01)	.0(.01)
8	.14(.09)	.14(.09)	.27(.18)	.14(.12)	.14(.12)	.13(.09)bi	.17(.09)	.17(.09)	.17(.09)
9	.14(.09)	.15(.09)	.21(.13)	.12(.1)	.11(.1)	.11(.11)b	.15(.1)	.15(.1)	.15(.1)
11	.37(.0)	.37(.02)	.44(.1)	.37(.0)bi	.37(.0)b	.37(.01)bi	.37(.0)	.37(.0)i	.45(.09)
12	.11(.02)	.12(.03)	.14(.03)	.11(.0)bi	.11(.0)bi	.11(.01)bi	.11(.0)	.11(.0)	.14(.03)
13	.49(.05)	.49(.05)	.56(.16)	.48(.03)i	.48(.03)i	.49(.04)bi	.5(.04)	.5(.04)	.63(.21)
14	.25(.27)	.29(.31)	.27(.28)	.21(.23)	.21(.23)b	.16(.15)bi	.13(.0)i	.13(.0)i	.26(.3)
16	.02(.01)	.03(.03)	.05(.05)	.01(.01)b	.01(.01)b	.01(.01)b	.0(.0)	.0(.0)	.0(.0)
17	.01(.0)	.01(.01)	.02(.03)	.0(.0)b	.0(.0)b	.0(.0)b	.0(.0)	.0(.0)	.0(.0)
18	.27(.09)	.26(.07)	.35(.17)	.17(.03)bi	.17(.05)bi	.21(.11)bi	.39(.12)	.39(.12)	.39(.12)
19	.14(.07)	.13(.06)	.18(.11)	.09(.06)bi	.1(.06)bi	.13(.09)b	.15(.08)	.15(.08)	.15(.08)
21	.2(.23)	.2(.23)	.21(.23)	.22(.25)	.2(.23)b	.16(.16)bi	.32(.33)	.32(.33)	.28(.31)
22	.11(.0)	.15(.17)	.19(.23)	.11(.0)bi	.13(.12)bi	.13(.12)bi	.12(.01)	.15(.16)	.23(.28)
23	.44(.07)	.44(.07)	.5(.18)	.44(.07)	.44(.07)	.45(.07)b	.48(.09)	.48(.1)	.54(.14)
24	.39(.09)	.39(.09)	.42(.15)	.4(.11)	.4(.11)	.41(.13)	.46(.14)	.46(.14)	.48(.19)
26	.11(.1)	.09(.09)	.2(.15)	.04(.04)b	.04(.04)b	.05(.05)b	.08(.12)	.08(.12)	.09(.13)
27	.09(.11)	.1(.11)	.24(.15)	.02(.02)b	.02(.02)b	.04(.04)b	.06(.09)	.06(.09)	.06(.09)
28	.36(.16)	.36(.17)	.43(.21)	.18(.12)b	.19(.13)b	.24(.16)b	.2(.17)	.2(.17)	.19(.16)
29	.22(.15)	.23(.15)	.33(.18)	.13(.14)b	.12(.14)b	.17(.18)b	.14(.16)	.14(.16)	.17(.18)
31	.49(.13)	.49(.13)	.58(.15)	.46(.11)i	.46(.12)i	.49(.11)bi	.59(.09)	.59(.09)	.59(.09)
32	.06(.1)	.06(.09)	.13(.14)	.05(.08)i	.05(.08)i	.07(.1)i	.12(.11)	.12(.11)	.12(.11)
33	.46(.13)	.44(.11)	.52(.16)	.42(.1)bi	.42(.09)bi	.45(.09)bi	.56(.11)	.56(.11)	.56(.11)
34	.04(.08)	.04(.08)	.08(.11)	.03(.07)i	.03(.07)i	.04(.07)bi	.1(.13)	.1(.13)	.1(.13)
36	.52(.17)	.5(.17)	.6(.18)	.43(.14)bi	.42(.13)bi	.47(.15)bi	.85(.14)	.85(.14)	.85(.14)
37	.05(.04)	.06(.06)	.15(.15)	.04(.03)bi	.04(.03)i	.07(.06)bi	.13(.09)	.13(.09)	.13(.09)
38	.55(.13)	.56(.12)	.67(.16)	.5(.09)i	.5(.09)bi	.54(.12)bi	.88(.12)	.88(.12)	.88(.12)
39	.04(.02)	.04(.03)	.09(.08)	.03(.01)bi	.03(.01)bi	.04(.03)bi	.11(.09)	.11(.09)	.11(.09)
41	.26(.15)	.3(.18)	.4(.23)	.19(.11)bi	.18(.09)bi	.25(.17)bi	.38(.21)	.38(.21)	.39(.22)
42	.04(.02)	.04(.02)	.06(.07)	.02(.01)bi	.02(.01)bi	.03(.02)bi	.09(.13)	.09(.13)	.11(.14)
43	.43(.13)	.44(.14)	.58(.15)	.42(.13)i	.41(.13)i	.44(.13)bi	.52(.16)	.52(.16)	.54(.17)
44	.11(.05)	.12(.07)	.17(.11)	.1(.05)i	.09(.05)bi	.11(.05)bi	.18(.1)	.18(.1)	.22(.13)
Tchebychef									
5	.14(.05)	.15(.05)	.24(.1)	.13(.04)	.13(.04)b	.15(.07)bi	.12(.01)i	.12(.01)i	.19(.06)
10	.07(.03)	.07(.03)	.08(.05)	.05(.02)bi	.05(.02)i	.07(.03)	.06(.03)	.06(.03)	.06(.03)
15	.32(.01)	.34(.04)	.46(.15)	.32(.0)i	.32(.0)bi	.34(.05)bi	.33(.01)	.33(.01)	.47(.15)
20	.08(.01)	.08(.01)	.11(.04)	.07(.01)bi	.07(.01)bi	.09(.02)i	.11(.03)	.11(.03)	.11(.03)
25	.04(.01)	.04(.02)	.16(.19)	.04(.0)bi	.04(.0)bi	.04(.01)bi	.04(.0)	.04(.0)	.14(.19)
30	.12(.02)	.13(.01)	.15(.03)	.1(.01)b	.1(.01)b	.11(.01)b	.07(.03)i	.07(.03)i	.08(.04)i
35	.06(.02)	.06(.02)	.08(.03)	.05(.02)bi	.05(.02)bi	.06(.02)bi	.07(.03)	.07(.03)	.07(.03)
40	.09(.04)	.09(.04)	.16(.07)	.07(.02)bi	.07(.02)bi	.09(.06)bi	.17(.06)	.17(.06)	.17(.06)
45	.12(.05)	.12(.05)	.21(.1)	.08(.02)bi	.09(.03)bi	.11(.05)bi	.16(.1)	.16(.1)	.19(.11)

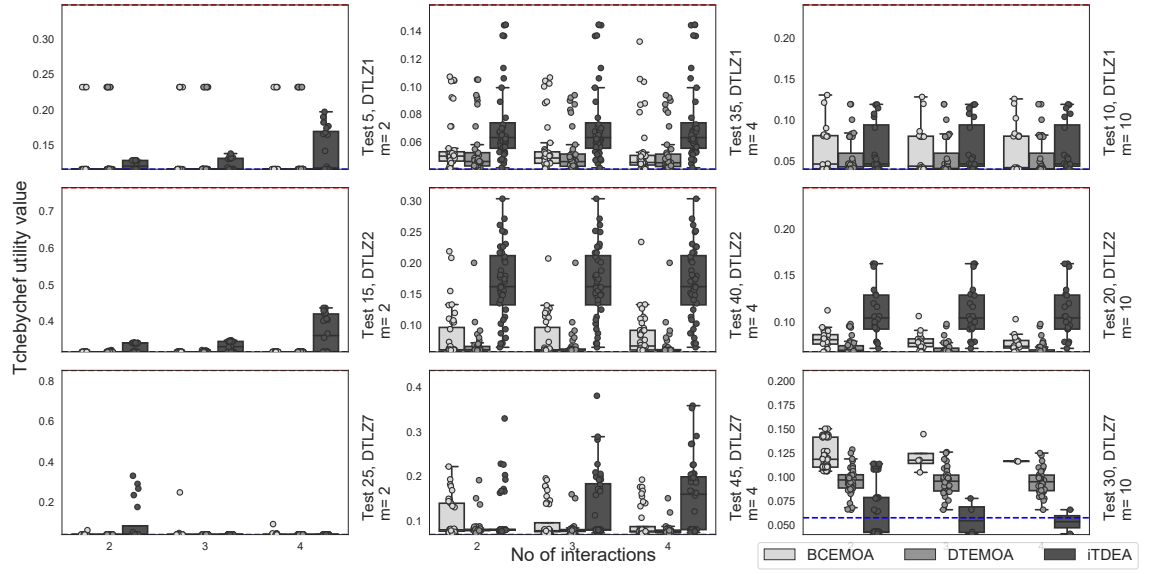


Figure C.2. Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA under ideal conditions with Tchebychef UF.

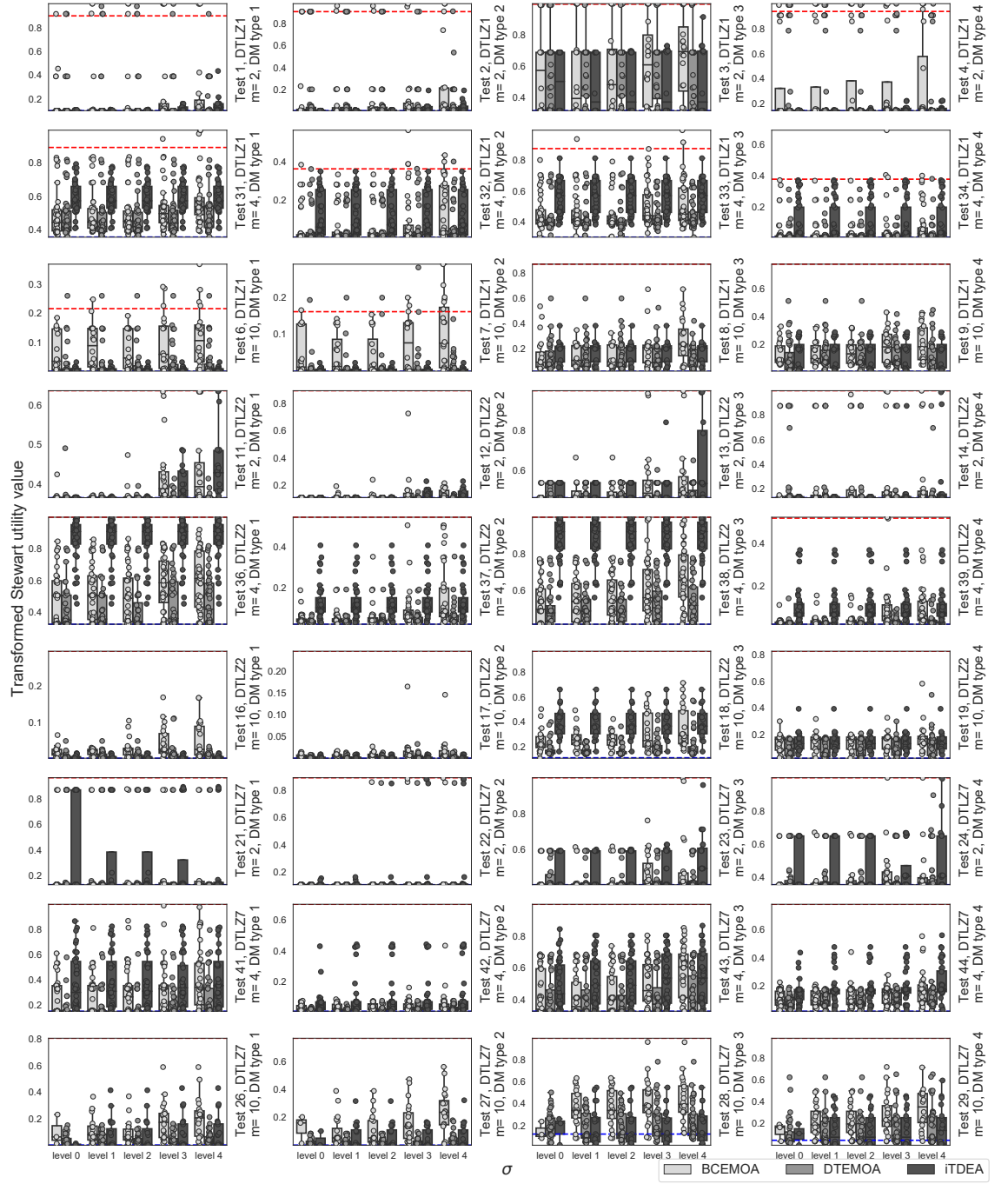


Figure C.3. Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA with simulation of inconsistencies in DM's decisions and Stewart UF.

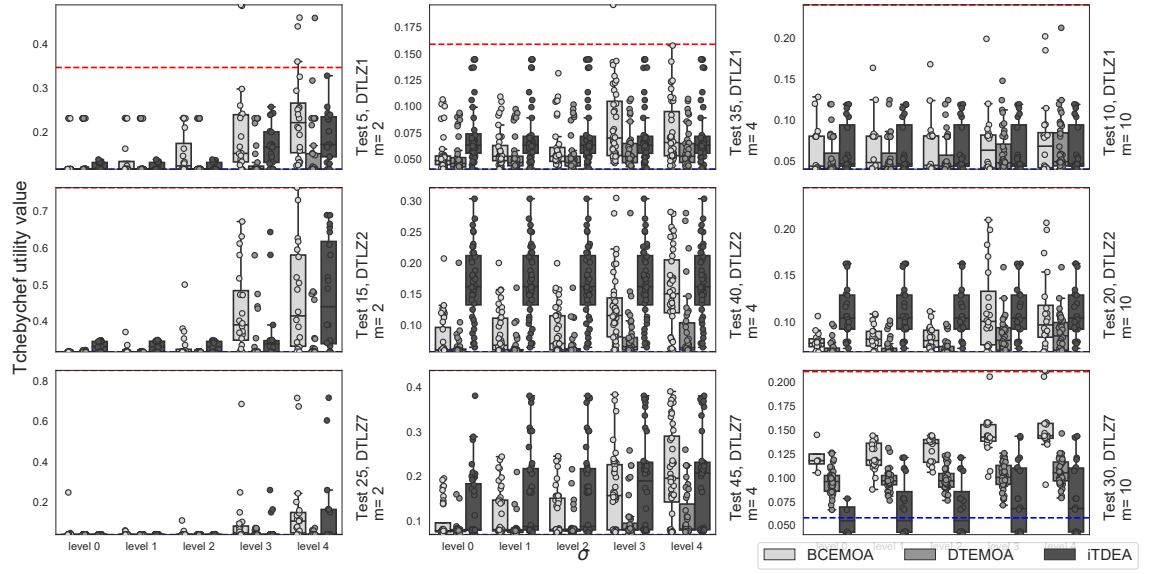


Figure C.4. Evaluation of the DTEMOA's performance in comparison with BCEMOA and iTDEA with simulation of inconsistencies in DM's decisions and Tchebycheff UF.