Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Zacatenco

Departamento de Computación

Herramientas de exploración para el tratamiento de problemas de optimización con muchos objetivos

Tesis que presenta

Oliver Fernando Cuate González

para obtener el Grado de

Doctor en Ciencias en Computación

Director de la Tesis

Dr. Oliver Steffen Schütze

Ciudad de México **Diciembre, 2019**

Center for Research and Advanced Studies
of the National Polytechnical Institute

Zacatenco Campus

Computer Science Department

Exploration Tools for the Treatment of Many-objective Optimization Problems

Submitted by

Oliver Fernando Cuate González

As a fulfillment of the requirement for the degree of

Ph.D. in Computer Science

Advisor

Dr. Oliver Steffen Schütze

Mexico City **December, 2019**

Resumen

El *problema de optimización multiobjetivo (POM)*, surge de manera natural en diversas áreas del conocimiento, como Economía, Finanzas y, en general, en la Industria; en las cuales se requiere optimizar simultáneamente dos o más funciones objetivo. Una de las principales características de un POM es que su conjunto solución, llamado *Conjunto de Pareto*, típicamente forma un objeto de dimensión $(k - 1)$, en donde k es el número de objetivos involucrados en el problema. En la actualidad, es posible aproximar dicho conjunto de interés de forma completa para un número relativamente moderado de funciones objetivo (por ejemplo, para $k = 3$ o 4). En este trabajo, abordamos el tratamiento numérico de POMs con más de 4 objetivos, que también se denominan *problemas de optimización con muchos objetivos (POMOs)*, que recientemente han captado el interés en la industria, ya que los procesos de toma de decisiones resultan ser cada vez más complejos. Para resolver este problema, utilizamos como marco de referencia el *Pareto Explorer (PE)*, que se introdujo por primera vez en la tesis de maestría del autor de este trabajo. La fase principal de este método es la exploración del conjunto de soluciones basado en las preferencias del tomador de decisiones. Aunque esta fase mostró resultados muy prometedores en problemas continuos y dos veces diferenciables, el uso del PE como una herramienta completa para resolver diferentes tipos de POMs estaba inconcluso hasta ahora.

En este trabajo, nos enfocamos en resolver algunas deficiencias del PE como una herramienta completa. Entre otros, nos concentramos en el tratamiento de problemas con diferentes supuestos de suavidad, el cálculo de buenas soluciones iniciales y la aplicación del PE a problemas del mundo real. Además, desarrollamos una heurística para preservar la diversidad en el espacio de decisión y proporcionamos una descripción completa sobre cómo construir POMOs que sean escalables en variables, objetivos y número de restricciones.

Abstract

In many areas such as Economics, Finance, or Industry, problems naturally arise, having several objectives that need to be optimized simultaneously. These are the so-called *Multiobjective Optimization Problems* (MOPs). One important characteristic of a MOP is that its solution set, the *Pareto Set* (PS), typically forms a $(k-1)$-dimensional object where k is the number of objectives involved in the MOP. Today, it is only possible to approximate the entire set of interest for relatively few objectives (say, $k = 3$ or 4). In this work, we address the numerical treatment of MOPs with more than 4 objectives which are also termed as *Many Objective Optimization Problems* (MaOPs) which have recently caught the interest in Industry since decision making processes are getting increasingly complex. To solve this problem, we use as baseline framework the *Pareto Explorer* (PE), a recently proposed continuation method for the treatment of MaOPs, which was first introduced in the Masters thesis of the author. The main phase of this method is the steering along the solution set based on the preferences of the decision maker. Although this phase showed very promising results in continuous and twice differentiable problems, the use of the PE as a complete framework to solve different kinds of MOPs was incomplete until now.

In this work, we focus on solving some shortcomings of the PE as a complete tool. Among others, we concentrate on the treatment of problems with different smoothness assumptions, the computation of good initial solutions, and the application of the PE to real-world problems. Furthermore, we develop a heuristic to preserve diversity in decision space and we provide a full description of how to construct M(a)OPs that are scalable in variables, objectives, and number of constraints.

A Manuel y Javier,
dos ángles en el cielo
y a mi madre, Consuelo,
mi ángel en vida.

Agradecimientos

Primeramente, agradezco a mi asesor, Dr. Oliver Schütze, por su incansable labor como investigador y su presión constante, ambos factores contribuyeron enormemente en los resultados de este trabajo.

Agradezco también a CONACyT por la beca proporcionada para realizar mis estudios de doctorado, así como los apoyos para estancias y congresos. A CINVESTAV, un gran institución en cuyas instalaciones encontré el ambiente propicio para llevar a buen puerto mi investigación; además, me permitió tener un desarrollo integral mediante actividades culturales y deportivas. De igual forma, agradezco los apoyos económicos recibidos por parte del CINVESTAV para estancias y congresos.

Gracias a mis sinodales, Dr. Guillermo Morales y Dr. Saúl Zapotecas, por sus importantes aportes a este trabajo. Al Dr. Carlos Coello, no solo por sus comentarios sobre este documento, sino también por sus excelente cursos, aprendí mucho en cada uno de ellos. De forma muy especial, a la Dra. Adriana Lara, quien ha estado presente en mi trayectoria académica, como mi asesora de tesis en la licenciatura y como sinodal en maestría y doctorado, y a quien siempre he admirado por su labor profesional, pero aún más por su forma de ser como persona.

Quiero agradecer y reconocer el apoyo incondicional de mi madre, Consuelo Cuate. Gracias por tu amor y comprensión en todas las etapas de mi vida, por siempre dar lo mejor de ti, eres una motivación constante para hacer lo mismo. Agradezco también al resto de mi familia, siempre presente, en especial a mis tíos, Francisco y María, por permitirme quedarme en su hogar cuando fue necesario.

Agradezco a mi gran amigo, Carlos González, por estar presente en todos los buenos y malos momentos. A Diana Medrano, por su apoyo y amistad. A mis amigos de la maestría, David Laredo y Jhonatan Perera, por siempre tener los mejores consejos, a pesar de la distancia y de seguir caminos distintos. A mis amigos de la maestría que siguieron también en el doctorado, Jesús Chi, por recordar traerme algo cada que sale de viaje; y especialmente, a Angélica Serrano, por su compañía y por todos los buenos momentos. A Rebeca Plata, por leer este trabajo en detalle y por sus valiosos comentarios. Gracias a todos mis amigos, son parte importante en mi vida.

A mis “hermanos académicos”, Adrián Sosa, Carlos Hernández y Sergio Alvarado, aprendí algo de cada uno de ellos y en verdad fue notoria su ausencia en la etapa final de mi doctorado. Mención aparte merece Lourdes Uribe, quien además es una gran amiga y que afortunadamente compartió conmigo toda mi estancia en el doctorado; hicimos un muy buen equipo.

Al grupo de secretarias, Sofía, Felipa y Erika, por estar pendiente de todos los estudiantes y apoyarnos siempre con la mejor disposición. A todo el personal del Departamento de Computación, entre otros, Arcadio y José Luis, por hacer aún más agradable mi estancia en este lugar. A todos los doctores con los que tuve la oportunidad de tomar algún curso y que ayudaron a ampliar mis conocimientos. También a todas las personas con las que tuve la fortuna de colaborar académicamente, en especial al Dr. Alejandro Alvarado, que se nos adelantó en el camino. A todos los miembros de mi equipo de fútbol, el Bayer, fueron buenos torneos y al final se logró el campeonato. A la toda comunidad Cinves-tenis, disfruté mucho practicando este deporte.

Por último, dedico este trabajo a mi tío, Manuel Baltazar, quien me inculcó el gusto por la lectura y, en gran medida, por las matemáticas. Fuiste parte fundamental de lo que soy el día de hoy, me hubiera gustado que leyeras estas líneas. También a mi primo, Javier Cuate, quien sin palabras logró darme algunas de las lecciones más importantes en mi vida. En paz descansen.

Gracias a todos los que de una forma u otra formaron parte de esta experiencia.

Contents

List of Figures **xviii**

List of Tables **xxi**

List of Algorithms **xxiv**

List of Acronyms **xxvii**

1 Introduction **1**

1.1 Motivation 2

1.2 Hypothesis 3

1.3 Aims of the Thesis 3

1.4 Contributions of this thesis 4

1.5 Publications 5

1.5.1 Prizes 5

1.5.2 JCR Journals 5

1.5.3 Other Journals 6

1.5.4 Proceedings 7

1.5.5 Textbooks 7

1.6 Organization of the Thesis 8

2 Basic Concepts 9

2.1 Single Objective Optimization . . . 9

2.1.1 Line Search Strategies . . . 11

2.1.2 Newton Method . . . 12

2.1.3 BFGS Method . . . 12

2.2 Multiobjective Optimization . . . 13

2.2.1 Definitions . . . 14

2.2.2 Optimality Conditions . . . 16

2.3 Solving a MOP . . . 17

2.3.1 Mathematical Programming Techniques . . . 17

Continuation Methods . . . 17

Method of Hillermeier . . . 18

Directed Search Predictor-Corrector Method . . . 20

Reference Point Methods . . . 22

Reference Point Problem . . . 22

Achievement Functions . . . 22

2.3.2 Interactive Methods . . . 23

Pareto Navigator Method . . . 23

NIMBUS Method . . . 24

Nautilus Method . . . 24

2.3.3 Many-objective Optimization and Evolutionary Algorithms . . . 24

Indicator based . . . 26

Large Populations . . . 26

Dimension Reduction . . . 26

2.3.4 MOEAs for CMOPs . . . 27

2.3.5 Diversity in Decision Space 27
2.4 Multiobjective Benchmark Problems 29
2.5 Performance Indicators 30
2.5.1 Δ_p indicator 31
2.5.2 Feasibility Ratio 32
2.5.3 Reference Inverted Generational Distance 32
2.6 Pareto Tracer and Pareto Explorer 33
2.6.1 Pareto Tracer Method 33
Predictor 33
Corrector 35
Handling Equality Constrains 35
Handling Inequality Constraints 38
2.6.2 Pareto Explorer 40
Steering in Objective Space 41
Steering in Decision Space 42
Steering in Weight Space 46

3 Extensions of the Pareto Explorer for Continuous MaOPs 49
3.1 Unbiased Neighborhood Exploration 49
3.2 Pareto Explorer for Finding the Knee 51
3.2.1 Definition of the Knee 53
3.2.2 Finding the Knee 53
3.3 Real World Application: Plastic Injection Molding 55
3.3.1 The Model 55
Design parameters 55
Objectives 57

3.3.2 Case study: A plastic gear 58
Building the model 59
3.3.3 Numerical Results 59
Multi-objective PIM design 60
Many-objective PIM design 61

4 Exploration in Objective Space 71

4.1 Fine tuning method and application to knapsack 71
4.1.1 Framework for the fine tuning method 72
4.1.2 Framework instantiation 74
The evolutionary solving process 75
Illustrative scenarios 76
4.1.3 Numerical results 77
4.2 Pareto Explorer for Linear MaOPs 79
4.2.1 Change in Objective Space 79
4.2.2 Change in Weight Space 82
4.2.3 Numerical Results 84
3-objective case 84
5-objective case 85

5 MOEA-PT 87

5.1 First stage: Rough Approximation via Micro-NSGA-II 87
5.2 Second stage: Refinement via PT 89
5.2.1 BOPs 89
5.2.2 General MOPs 91
5.3 Proposed Test Problems 92
5.3.1 Eq1-ZDT1 92

5.3.2 Eq2-ZDT1 96

5.3.3 Eq-Quad 97

5.4 Numerical Results 98

6 Exploration in Decision Space 103

6.1 Proposed Framework 104

6.1.1 Using the Averaged Distance in Decision Variable Space . . . 104

6.1.2 Variation Rate 106

6.1.3 Integration into NSGA-II 109

6.1.4 Integration into NSGA-III 109

6.1.5 Integration into MOEA/D 111

6.1.6 Integration into SMS-EMOA 112

6.2 Numerical Results 113

7 A New Benchmark suite for Equality Constrained MOPs 125

7.1 Hyper-spheres as equality constraints 125

7.1.1 Hyper-spheres 125

7.1.2 Embedding into higher dimensions 127

7.2 Equality constrained MOPs 127

7.2.1 Eq-DTLZ 128

Pareto sets for Eq-DTLZ 1-4 129

7.2.2 Eq-IDTLZ 131

7.2.3 Examples 132

7.3 Performance of MOEAs on Eq-(I)DTLZ 135

8 Conclusions and Future Work 143

8.1 Obtained Results 143

8.2 Conclusions 145

8.3 Future Work 147

A Appendix 1 149

A.1 A Plastic Gear 149

References 152

List of Figures

2.1 Functions in conflict 13

2.2 Compromise solutions 15

2.3 Pareto points 16

2.4 Geometrical idea of Hillermeier method 19

2.5 Example of the Directed Search predictor-corrector method 21

2.6 Tangent vectors 34

2.7 Illustration of the steering in objective space 42

2.8 Illustration of the steering in decision space 44

2.9 Illustration of the steering in weight space 48

3.1 Example of the unbiased neighborhood exploration 52

3.2 Illustrative example of the use of PE to find the knee 56

3.3 Illustrative example where PE is not able to find the knee 56

3.4 Result for on the PIM model with two objectives (f_2 and f_6) 61

3.5 Graphical result on the PIM model with two objectives (f_1 and f_5) . 62

3.6 Graphical result on the the PIM model with three objectives (f_1, f_5, and f_6) 65

3.7 Graphical result on the complete PIM model for the first scenario (minimize f_1 and f_5). 66

3.8 Graphical result on the complete PIM model for the second scenario (minimize f_2 and f_6). 67

3.9 Graphical result on the complete PIM model for the third scenario (minimize f_1, f_5, and f_6). 68

3.10 Graphical result on the complete PIM model for the last scenario (minimize f_3, f_5, and f_6). 69

4.1 Illustrative example of fine tuning in objective space. 74

4.2 Illustrative scenarios on the MOKP for the bi-objective case 76

4.3 Illustrative example of the steering in objective space for linear MOPs. 82

4.4 Illustrative example of the change in weight space for linear MOPs. . 84

4.5 Graphical result on the bensolvedron problem with three objectives. . 85

4.6 Result of the 5-objective case. 86

5.1 Pareto set and front of the Eq1-ZDT1 with $n = 30$. 95

5.2 Pareto set and front of the Eq2-ZDT1 with $n = 30$. 98

5.3 Constraints and feasible region for the Torus problem 99

5.4 Pareto front approximations on the Eq1-Quad problem for the selected MOEAs 101

5.5 Pareto front approximations on the Eq2-Quad problem for the selected MOEAs 101

5.6 Pareto front approximations for the C-ZDT1 problem for the selected MOEAs 101

5.7 Pareto front approximations on the C-ZDT2 problem for the selected MOEAs 102

5.8 Pareto front approximations on the C-ZDT3 problem for the selected MOEAs 102

5.9 Pareto front approximations on the D&D problem for the selected MOEAs 102

6.1 Hypothetical bi-objective problem where two connected components of the PS map to the entire PF. 104

6.2 Four different Pareto set/front approximations where all PF approximations are good, but the PS approximation is complete only in one of them. 105

6.3 Illustrative example of the VR. 106

6.4 Graphical results of the run with the median values on the OMNI1 function with the (VR-)NSGA-II algorithms. 120

6.5 Graphical results of the run with the median values on the OMNI2 function with the (VR-)NSGA-II algorithms. 121

6.6 Graphical results of the run with the median values on the SCM1 function with the (VR-)SMS-EMOA algorithms. 121

6.7 Graphical results of the run with the median values on the RPH1 function with the (VR-)NSGA-II algorithms. 122

6.8 Graphical results of the run with the median values on the RPH2 function with the (VR-)NSGA-III algorithm. 122

6.9 Graphical results of the run with the median values on the RPH3 function with the (VR-)MOEA/D algorithms. 123

7.1 Different projections of the analytic Pareto set with $k = 4$, $n = 13$ and one equality constraint. 133

7.2 Different projections of the analytic Pareto set with $k = 4$, $n = 13$ and two equality constraints. 133

7.3 Graphical results of the selected MOEAS on the Eq-DTLZ1 problem 141

7.4 Graphical results of the selected MOEAS on the Eq-DTLZ2 problem 142

A.1 Warpage in the plastic part. 151

List of Tables

3.1 Objective functions of the PIM model 59

3.2 Parameters of the NSGA-III and the PT for the PIM. 60

3.3 Computational cost of the PE for the Scenarios 1-4 on the PIM. . . . 63

3.4 Comparison of the values from our model (F_M) against the simulated values (F_S) on the PIM. 63

4.1 Parameters settings of the fine tuning method adopted for obtaining the numerical results. 78

4.2 Numerical results of the fine tuning method on the MOKP 78

5.1 Values of γ and η 97

5.2 Values of Δ_2 and I_F on the selected test problems. 100

6.1 Parameter configuration for each algorithm 116

6.2 Numerical results for the original and variation rate version of some MOEAs in standard benchmark test problems. 118

6.3 Numerical results for the original and variation rate version of some MOEAs in Type III test problems. 119

7.1 Eq-DTLZ and Eq-IDTLZ benchmark problems 134

7.2 Equality constraints for different values of k and p and the general case.135

7.3 Parameters of the selected MOEAs. 137

7.4 Results for Eq-DTLZ 1-4 and Eq-IDTLZ 1-2 with $k = 3$ and $p = 1$ for some MOEAs. 138

7.5 Results for Eq-DTLZ 1-4 and Eq-IDTLZ 1-2 with $k = 4$ and $p = 1$ for some MOEAs. 139

7.6 Results for Eq-DTLZ 1-4 and Eq-IDTLZ 1-2 with $k = 4$ and $p = 2$ for some MOEAs. 140

A.1 Material properties of PP. 149

A.2 Design variables. 150

A.3 Process parameters. 150

A.4 Flow rate profile. 150

A.5 Packing rate profile. 150

A.6 Objective functions. 151

List of Algorithms

2.1 Framework of the Pareto Explorer 41
2.2 Steering in objective space 43
2.3 Steering in decision space 45
2.4 Modified steering in decision space 46
2.5 Steering in weight space 47
3.1 Framework of UNE 50
3.2 UNE: Evenly distributed solutions in objective space 51
4.1 Fine Tuning Framework 73
4.2 PE: Movement in Objective Space 81
4.3 PE: Change in Weight Space 83
5.1 ϵ-NGSA-II 90
6.1 Framework to include the Average Distance in Decision Variable Space within any MOEA 108
6.2 Pseudocode of VR-NSGA-II 110
6.3 Iteration of the VR-NSGA-III 111
6.4 Pseudocode of VR-MOEA/D 112
6.5 Iteration of the VR-SMS-EMOA 113

Acronyms

BFGS: *Quasi-Newton method of Broyden, Fletcher, Goldfarb, and Shanno*

DM: *Decision Maker*

DTLZ: *Deb-Thiele-Laumanns-Zitzler*

GA: *Genetic Algorithm*

HV: *Hypervolume*

KKT: *Karush, Kuhn and Tucker*

MaOP: *Many Objective Optimization Problem*

MOP: *Multiobjective Optimization Problem*

MOEA/D: *Multiobjective Evolutionary Algorithm Based on Decomposition*

MOKP: *Multi-Objective (multi-dimensional) 0-1 Knapsack Problem*

PE: *Pareto Explorer*

PF: *Pareto Front*

PIM: *Plastic Injection Molding*

PS: *Pareto Set*

SOP: *Scalar Optimization Problem*

QN: *Quasi-Newton*

WASF: *Wierzbicki's Achievement Scalarizing Function*

ZDT: *Zitzler-Deb-Thiele*

Chapter 1

Introduction

Optimization problems with multiple objectives arise naturally in areas such as Economics, Finance and Industry, where it is necessary to obtain the greatest profit with limited resources. Thus, the objectives under consideration are generally in conflict with each other. Nowadays, there exist a lot of methods available for solving such *Multiobjective Optimization Problem*s (MOPs) [Coello et al., 2007, Deb, 2001, Miettinen, 1999], each of then with strengths and weaknesses, looking increasingly to efficiently solve more general problems. The solution to these problems typically is not a single point but should rather a set of compromise vectors of the objectives to be optimized. Ideally, these methods obtain a set of points whose images have a uniform spread along the entire solution set of the given problem. However, for many applications, it is important for a *Decision Maker* (DM) to find a set of points which satisfy certain values for each objective function.

To solve this problem, we use as our baseline framework the *Pareto Explorer* (PE), a recently proposed continuation method for the treatment of *Many Objective Optimization Problem*s (MaOPs), which was first introduced in the MSc thesis of the author. The main phase of this method is the steering –in objective or decision space– based on the preferences of the decision maker, which is presented in [Schütze et al., 2019]. Although this phase shows very promising results in continuous and doubly differentiable problems, the use of the PE as a complete framework to solve different kinds of MOPs was incomplete until now.

In this work, we focus on solving some shortcomings of the PE as a complete tool. Among others, we concentrate on the treatment of problems with different smoothness assumptions, the computation of good initial solutions, and the applications of the PE in real-world problems. Furthermore, we provide two more ways of steering for the continuous case.

1.1 Motivation

An important characteristic of continuous MOPs is that their solution sets, the *Pareto Sets* (PS), typically form a $(k-1)$-dimensional object where k is the number of objectives involved in the MOP. Thus, it is only possible to approximate the entire set of interest for a relatively low number of objectives (say, $k = 3$ or 4). In this work, we also address the numerical treatment of MOPs with more than 4 objectives which are termed as MaOPs. MaOPs have recently caught the interest in industry due to the huge success of existing methods for the treatment of MOPs and because decision making processes are getting more and more complex.

Several strategies for the treatment of continuous MaOPs have been proposed and tested over the past decades. However, we can identify two principal classes of approaches as the most common techniques to solve MaOPs. The first one is the use of evolutionary algorithms in order to compute the *entire* set of optimal solutions. This approach has the issue that for a MaOP a good approximation to the entire solution set implies computing a huge number of candidate solutions. For example, if we have k objective functions and we want M points for each dimension, then the solution set will have M^{k-1} points. That is, we have an exponential growth in k. This is not suitable for a DM since he/she must evaluate all the solutions.

The second one is the use of local programming techniques that produce only one single solution, which is, in general, not enough for a DM. Examples within this class of methods are the reference point methods.

The PE method raises as a solution for continuous MaOPs and it is conceived as a global/local exploration tool for the treatment of MaOPs, which consists of two principal phases: *i)* obtaining a global optimal solution for a given MaOP and *ii)* the local exploration of optimal solutions in a given direction provided by the DM.

In order to effectively set up this idea, we need to precisely define the role of the direction provided by the DM and the meaning of improving a given solution according to this direction. First, let us comment that specifying a direction with respect to a starting solution can easily be thought by the DM in many different ways and for different purposes. For the sake of illustration, let us consider a MOP with three objectives (f_1, f_2, f_3) and the following scenario: the DM has an optimal solution for this problem that he/she is not fully satisfied with, e.g., he/she would like to minimize the value of f_2 as much as possible. However, a lot of options can be considered for the above example. For instance, the vector $d_1 = (1, -1, 0)^T$ can refer to a direction (in objective space) aiming at reducing the second objective, while increasing the first one. Similarly, the direction $d_2 = (0, -1, 1)^T$ would imply a reduction of the second objective together with a growth on the third objective. In both cases, we could obtain the minimum value for f_2, but following the direction d_1 or d_2 typically produces different paths that can be associated with different DM preferences. By defining a direction, the DM can actually decide which objectives to improve and which ones to "sacrifice" in order to refine his/her preferences. Performing such local movements in objective space while following a whole path of Pareto optimal solutions with respect to the preferred direction of the DM is the goal of the proposed framework.

Though the first results were very promising, there are several lines of research arising from this method. For instance, the PE method has been originally developed for continuous MaOPs. However, in this work we extended the implementation of the PE method for different smoothness assumptions, in particular, we consider discrete and linear problems. The use of evolutionary algorithms or other techniques to find global solutions is the first step to develop a memetic algorithm with the PE. This may be considered in order to potentiate its applicability for real word problems.

Finally, the steering phase of the PE is based on the Pareto Tracer method which can handle equality constraints. MOPs with equality constraints are scarce in the literature, thus, the proposal of a benchmark for MaOPs is also needed.

1.2 Hypothesis

It is expected that the PE will be a widely accepted tool in academy and industry for the numerical treatment of MaOPs. The PE will be developed and extended for problems with different smoothness assumptions and will be used on several demonstrators coming from real-world applications.

1.3 Aims of the Thesis

To develop and extend the PE method in order to provide an efficient numerical tool for the adequate numerical treatment of MaOPs with different smoothness assumptions. It is expected that the new algorithm pushes the state-of-the-art in this field.

Our particular aims are presented in the following list

1. Extension of PE for smooth MaOPs. Here, we mainly focus on three aspects: the use of PE for finding the knee of the Pareto front, the neighborhood exploration, and the application of our framework to solve a real-world problem.

    - Development of new steering directions (according to the application). For instance, we can focus on the steering toward the knee of the Pareto front, which usually represents the best trade-off between all the involved objectives. In many applications, the DM is looking for such a point.

   - Detection of better distributions of the neighborhood for the phase of *complete exploration* of the PE. We can see the desirable directions as points evenly distributed in a hypersphere; the computation of such points is still an open problem in maths. However, we can consider the existing approaches to approximate them.
   - Increase of the overall efficiency of the PE method. Although the comparisons against other methods are unfair by the originality of our proposal, we can experimentally show that the computational cost of PE is low in terms of function evaluations.

2. Computation of suitable initial solutions (according to the application). In the context of the PE framework, the DM needs a few representative initial candidate solutions to start with the steering process. The computation of such candidate solutions has to be performed efficiently.

3. Extension to MaOPs with different smoothness assumptions. It is common, for a wide variety of applications, that the second-order derivatives are not available. Thus, we also need a way to deal with such kind of problems.

   - Gradient-free version. Since the approximation of Jacobians and Hessians is computationally expensive.
   - Use of evolutionary algorithms for discrete problems. These algorithms have been very successful in solving such problems.

4. Use of the method to solve real-world problems and different applications. A way to validate our proposal is via an application, as the decision-making process sometimes is not adequately reflected with benchmark problems.

1.4 Contributions of this thesis

This thesis contributes to the area of *many-objective optimization*, specifically it provides an efficient way to steer a local search in a given direction. This contribution is a module of a numerical tool called PE.

Our particular contributions are the following:

- We proposed the PE as a complete framework. In this work, one can find several tools for the exploration of MaOPs, which can be integrated into the PE framework. These tools are related to the computation of initial optimal solutions, the steering in objective space for MaOPs with different smoothness assumptions, and the preservation of diversity in decision variable space.

- A MOEA that can compute a few good initial solutions for the steering in objective space. The obtained solutions are representative and, by the nature of MOEAs, they are at least good approximations of global solutions.
- A heuristic that preserves diversity in decision space. Most of the evolutionary algorithms focus only in objective space, neglecting the critical information provided by the variables of the problem. Moreover, in the PE context, diversity in decision space also means promising initial solutions for the steering in such space.
- A framework for the steering in objective space with different smoothness assumptions. In this way, the PE framework is now more robust, and it can be considered for solving a vast number of problems.
- A benchmark for constrained MOPs, which is scalable in the number of variables, objectives, and constraints. The mathematical expression of the Pareto set is given for all the presented instances.
- A software package for free scientific use.

1.5 Publications

In this section we present the publications that have been derived from this thesis (from 2016 to 2019).

1.5.1 Prizes

- **Cuate, Oliver**; Schütze, Oliver. "Variation Rate: An Alternative to Maintain Diversity in Decision Space for Multi-objective Evolutionary Algorithms". *EMO 2019: Evolutionary Multi-Criterion Optimization.*
Best Paper Award (Second Place)

1.5.2 JCR Journals

- **Cuate, Oliver**; Uribe, Lourdes; Lara, Adriana; Schütze, Oliver. "A Benchmark for Equality Constrained Multi-objective Optimization". *Swarm and Evolutionary Computation* DOI: https://doi.org/10.1016/j.swevo.2019.100619

Q1 in Computer Science, Artificial Intelligence and in Computer Science, Theory and Methods. Impact Factor: 6.33 in 2018 (3.242 5-year)

- Schütze, Oliver; **Cuate, Oliver**; Martín, Adanay; Peitz, Sebastian; Dellnitz, Michael. "Pareto Explorer: A global/local exploration tool for many-objective optimization problems". *Engineering Optimization*
  DOI: https://doi.org/10.1080/0305215X.2019.1617286
  Q2 in Engineering, Multidisciplinary and in Operations Research & Management Science. Impact Factor: 1.809 in 2018 (2.136 5-year)

- Alvarado-Iniesta, Alejandro; **Cuate, Oliver**; Schütze, Oliver. "Multi-objective and many objective design of plastic injection molding process". *International Journal of Advanced Manufacturing Technology.* Volume: 102, Issue: 9-12, Pages: 3165-3180. 2019.
  Q2 in Automation & Control Systems and in Engineering, Manufacturing. Impact Factor: 2.496 in 2018 (2.75 5-year)

- Wang, Honggang; Laredo, David; **Cuate Oliver**; Schütze, Oliver. "Enhanced directed search: a continuation method for mixed-integer multi-objective optimization problems". *Annals of Operations Research.* Volume: 279, Issue: 1-2, Pages: 343-365. 2019.
  Q2 in Operations Research & Management Science. Impact Factor: 2.284 in 2018 (2.267 5-year)

- Hernandez Mejía, Jesus Alejandro; Schütze, Oliver; **Cuate, Oliver**; Lara, Adriana; Deb, Kalyanmoy. "RDS-NSGA-II: A memetic algorithm for reference point based multi-objective optimization". *Engineering Optimization.* Volume: 49, Issue: 5, Pages: 828-845
  Q2 in Engineering, Multidisciplinary and in Operations Research & Management Science. Impact Factor: 1.809 in 2018 (2.136 5-year)

1.5.3 Other Journals

- **Cuate, Oliver**; Schütze, Oliver. "Variation Rate to Maintain Diversity in Decision Space within Multi-objective Evolutionary Algorithms". *Mathematical and Computational Applications.* Volume: 24, Issue: 3, Pages 82
  Emerging Sources Citation Index (ESCI) in Web of Science

- Bogoya, Johan; Vargas, Andrés; **Cuate, Oliver**; Schütze, Oliver. "A (p,q)-Averaged Hausdorff Distance for Arbitrary Measurable Sets". *Mathematical and Computational Applications.* Volume: 23, Issue: 3, Pages 51
  Emerging Sources Citation Index (ESCI) in Web of Science

- Pérez, Nancy; **Cuate Oliver**; Schütze, Oliver; Alvarado, Alejandro. "Including Users Preferences in the Decision Making for Discrete Many Objective Optimization Problems". *Computación y Sistemas*. Volume: 20, Issue: 4, Pages: 589-607
  LatIndex

1.5.4 Proceedings

- **Cuate, Oliver**; Schütze, Oliver. "Pareto Explorer for Solving Real World Applications". *18th Mexican International Conference on Artificial Intelligence (MICAI 2019)* (to appear)

- **Cuate, Oliver**; Uribe, Lourdes; Ponsich, Antonin; Lara, Adriana; Beltrán, Fernanda; Rodríguez Sánchez, Alberto; Schütze, Oliver. "A New Hybrid Metaheuristic for Equality Constrained Bi-objective Optimization Problems". *EMO 2019: Evolutionary Multi-Criterion Optimization*

- **Cuate, Oliver**; Schütze, Oliver; Grasso, Francesco; Tlelo-Cuautle, Esteban. "Sizing CMOS operational transconductance amplifiers applying NSGA-II and MOEAD". *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*

- **Cuate, Oliver**; Derbel, Bilel; Liefooghe, Arnaud; Talbi, El-Ghazali; Schütze, Oliver. "An Approach for the Local Exploration of Discrete Many Objective Optimization Problems". *EMO 2017: Evolutionary Multi-Criterion Optimization*

- **Cuate, Oliver**; Lara, Adriana; Schütze, Oliver. "A Local Exploration Tool for Linear Many Objective Optimization Problems". *2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*

1.5.5 Textbooks

- Schütze, Oliver; **Cuate, Oliver**. "Cálculo Diferencial". *Editorial Punto Fijo.* In press.
  Textbook on Differential Calculus on high school level in Spanish

- Schütze, Oliver; **Cuate, Oliver**. "Cálculo Integral". *Editorial Punto Fijo.* In press.
  Textbook on Integral Calculus on high school level in Spanish

- Schütze, Oliver; **Cuate, Oliver**. "Cálculo Mental". *Editorial Punto Fijo.* In press.
  Textbook on Mental Calculus on middle school level in Spanish

- Schütze, Oliver; **Cuate, Oliver**. "Club de Matemáticas: Resolviendo problemas con Álgebra". *Editorial Punto Fijo.* In press.
  Textbook for introductory algebra on middle school level in Spanish

1.6 Organization of the Thesis

This thesis consists of seven chapters, including this introductory chapter. The remainder of this document is organized as follows.

In Chapter 2, we present the basic concepts of scalar, multi, and many objective optimization problems. Further, we review some of the methods for solving a continuous MOP along with some methods for MaOPs and *Scalar Optimization Problem*s (SOPs). The PE and PT methods, the main tools for this work, are also described in detail in Chapter 2. In Chapter 3, we present some extensions of the PE for continuous problems; in particular, we propose the unbiased neighborhood exploration, and we provide a significant theoretical result that allows the use of PE for finding the knee of the PF. Also, in this chapter, we demonstrate the effectiveness of PE solving a real-world application: the plastic injection molding. On the other hand, the adaption of the steering phase in objective space with different smoothness assumptions is presented in Chapter 4; mainly, focusing on discrete and linear MaOPs. In Chapter 5, we present a hybrid algorithm using the Pareto Tracer, which is the core of PE, as the local searcher. While, in Chapter 6, we develop a heuristic to preserve diversity in the decision space. In Chapter 7, we provide a full description of how to construct M(a)OPs that are scalable in variables, objectives, and number of constraints. Finally, in Chapter 7, we present our conclusions along with some ideas to be considered for future work.

Chapter 2

Basic Concepts

We introduce some principal concepts and the necessary theoretical background throughout this chapter to understand this work. The scope of this work contemplates the treatment of a particular class of optimization problems, *Many Objective Optimization Problem*s (MaOPs). Thus, we start by defining a *Scalar Optimization Problem* (SOP) in Section 2.1 together with three different classes of algorithms to solve such problems. In Section 2.2 we address *Multiobjective Optimization Problem*s (MOPs) and state the definitions and optimality conditions used along this work. We also explain the difference between SOPs and MOPs, as well as the conflict that exists to find one or more suitable solutions for MOPs. Due to the importance of some methods and approaches developed to solve MOPs numerically, we describe the most widely used ones related to this work in Section 2.3. We describe some benchmarks for MOPs in Section 2.4; and in Section 2.5, we define the indicators that we use along this work. Finally, in Section 2.6, we describe in details the Pareto Tracer and the Pareto Explorer methods, that build the core of this work.

2.1 Single Objective Optimization

In a SOP, we have a unique objective function which depends on one or more variables, $f : \mathbb{R}^n \to \mathbb{R}$, and which is subject to certain constraints. We can write a continuous SOP in a standard form as:

$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f(x), \\ \text{s.t} & g_j(x) \leq 0, \quad i = 1, ..., m, \\ & h_i(x) = 0, \quad j = 1, ..., p, \end{array} \tag{2.1}$$

where $g_j : \mathbb{R}^n \to \mathbb{R},\ i = 1, \ldots, m$, are the inequality constraints and $h_i : \mathbb{R}^n \to \mathbb{R},\ j = 1, \ldots, p$, are the equality constraints. A special type of inequality constraints are the so called *box constraints*, which define limits for each component of the vector $x \in \mathbb{R}^n$, so box constraints have the form $a_i \leq x_i \leq b_i$, with $a, b \in \mathbb{R}^n$.

Definition 2.1. *The set of points* $\mathcal{X} = \{x \in \mathbb{R}^n \,|\, g(x) \leq 0 \textit{ and } h(x) = 0\}$ *is called feasible region where* $g : \mathbb{R}^n \to \mathbb{R}^m$ *and* $h : \mathbb{R}^n \to \mathbb{R}^p$ *are defined as the vector functions of the given inequalities and equalities, respectively.*

Unless specified differently, we assume along this work that the objective function and the constraints are continuously differentiable. Under this assumption, we define the gradient $\nabla f(x) \in \mathbb{R}^n$ of a multivariable function f as the vector consisting of the function partial derivatives

$$\nabla f(x) = \begin{pmatrix} \dfrac{\delta f}{\delta x_1}(x) \\ \vdots \\ \dfrac{\delta f}{\delta x_n}(x) \end{pmatrix} \tag{2.2}$$

and the Hessian matrix $\nabla^2 f(x) \in \mathbb{R}^{n \times n}$ as the square matrix of the second order derivatives

$$\nabla^2 f(x) = \begin{pmatrix} \dfrac{\delta f}{\delta^2 x_1}(x) & \dfrac{\delta f}{\delta x_1 \delta x_1}(x) \dots & \dfrac{\delta f}{\delta x_1 \delta x_n}(x) \\ \dfrac{\delta f}{\delta x_2 \delta x_1}(x) & \dfrac{\delta f}{\delta^2 x_2}(x) \quad \dots & \dfrac{\delta f}{\delta x_2 \delta x_n}(x) \\ \vdots & & \\ \dfrac{\delta f}{\delta x_n \delta x_1}(x) & \dfrac{\delta f}{\delta x_n \delta x_2}(x) \dots & \dfrac{\delta f}{\delta^2 x_n}(x) \end{pmatrix}. \tag{2.3}$$

To solve (2.1), the task is to find a vector $x^* \in \mathcal{X}$, such that the function evaluation at x^* gets a minimum value $f(x^*) \in \mathcal{X}$. That is, there is no other $x \in \mathcal{X}$, such that $f(x) < f(x^*)$. We can express this mathematically as follows:

Definition 2.2. *a) A point* x^* *a local minimizer of problem* (2.1) *if* $f(x^*) \leq f(x)$ *is satisfied within a feasible neighborhood of* x^**.*

b) A point x^* *is a global minimizer of problem* (2.1) *if* $f(x^*) \leq f(x)$ *for all* $x \in \mathcal{X}$

Notice that, for some problems, the value $f(x^*)$ could be obtained for more than one vector. This means that, if we have a feasible solution and depending on characteristics thereof, then it is possible to find a set of solution vectors, such that the function takes the minimum value for each one. However, the optimal value of the objective function is unique.

There exist a lot of methods to solve a SOP, each one of which tries to exploit the characteristics of the given problem. Due to the scope of this work, we will not be thorough about these methods. However, we briefly describe three of the most common methods for the continuous case since those concepts will be useful in this work.

2.1.1 Line Search Strategies

The idea of this kind of methods is to find a *descent direction* and then to compute a step size that produces a sufficient decrement along the given direction [Nocedal and Wright, 2006]. In general, an iteration of this method is given by

$$\begin{aligned} &x_0 \in \mathbb{R}^n, \\ &x_{i+1} = x_i + t_i \nu_i, \qquad i = 0, 1, \dots, \end{aligned} \tag{2.4}$$

where $x_i \in \mathbb{R}^n$, x_0 is the starting point, $t_i \in \mathbb{R}^+$ is a step size and $\nu_i \in \mathbb{R}^n$ is the descent direction. Mathematically, we have the following

Definition 2.3. *$\nu_i \in \mathbb{R}^n$ is a descent direction for a function $f : \mathbb{R}^n \to \mathbb{R}$ at a point $x_i \in \mathbb{R}^n$ if*

$$\nabla f(x_i)^T \nu_i < 0. \tag{2.5}$$

The above condition implies that

$$f(x_i + t_i \nu_i) < f(x_i), \quad \text{for } t > 0 \text{ sufficienty small.}$$

We can solve the following problem to find the best possible step size

$$\min f_\nu(t) = f(x + t\nu). \tag{2.6}$$

However, computing the exact solution of (2.6) may be costly. As a remedy, we can seek for an *acceptable* step size.

The Armijo condition allows to get a *sufficient decrease* in the objective function f. This condition is given by

$$f(x + t\nu) \leq f(x) + c_1 t \nabla f(x)^T \nu, \tag{2.7}$$

where $c_1 \in (0, 1)$. On the other hand, a rule that prevents *too small* steps lengths is given by

$$\nabla f(x + t\nu)^T \nu \leq c_2 \nabla f(x)^T \nu, \tag{2.8}$$

where $c_2 \in (c_1, 1)$. Equations (2.7) and (2.8) together are called the Wolfe conditions.

2.1.2 Newton Method

A Newton method step [Nocedal and Wright, 2006] is computed by

$$\begin{aligned} &x_0 \in \mathbb{R}^n, \\ &x_{i+1} = x_i - \nabla^2 f(x_i)^{-1} \nabla f(x_i), \qquad i = 0, 1, \ldots \end{aligned} \tag{2.9}$$

Here again, $x_i \in \mathbb{R}^n$ and $x_0 \in \mathbb{R}^n$ is the starting point. Notice that the structure of (2.9) is very similar to the line search step (2.4), we can consider that the direction $\nu_i = -\nabla^2 f(x_i)^{-1} \nabla f(x_i) \in \mathbb{R}^n$, and the step size $t_i = 1$.

The Newton method possesses some important properties, for instance, it presents typically local quadratic convergence. However, we need the Hessian at each iteration making the method computationally expensive.

2.1.3 BFGS Method

We can use numerical methods to approximate the Hessians, e.g. finite differences or automatic differentiation [Griewank and Corliss, 1992]. Yet, there exist more suitable methods to update the Hessians when we do not have this information at hand: the *Quasi-Newton* (QN) methods [Dennis and Moré, 1977]. QN methods build a quadratic model of the problem to approximate the Hessians at each iteration. These approximations produce, at most, superlinear convergence.

Some of the QN methods use a line search strategy, where the search direction is given by

$$\nu = -B^{-1} \nabla f(x), \tag{2.10}$$

where $B \approx \nabla^2 f(x)$. Notice that the the only difference with the Newton method is the use of B instead of the Hessian.

The most common method update is the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS), which is given by

$$B_{i+1} = B_i + \frac{B_i s s^T B_i}{s^T B_i s}, \tag{2.11}$$

where $s = x_{i+1} - x_i$ and and $y = \nabla f(x_{i+1}) - \nabla f(x_i)$.

A necessary and sufficient condition to guarantee the positive definiteness of B is the curvature condition which is satisfied by imposing the Wolfe condition (2.8) on the step size control.

2.2 Multiobjective Optimization

The continuous MOP is defined as

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & F(x), \\ \text{s.t} \quad & g_j(x) \leq 0, \quad i = 1, ..., m, \\ & h_i(x) = 0, \quad j = 1, ..., p, \end{aligned} \tag{2.12}$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^k,\ F(x) = (f_1(x), \ldots, f_k(x))^T,\ f_i(x) : \mathbb{R}^n \rightarrow \mathbb{R}\ i = 1, \ldots, k$, with a feasible region as in Definition 2.1.

The main difference between a SOP and a MOP is the nature of the solution set. As we explained before, the solution for a SOP, if it exists, is typically a unique optimum value that can be achieved by more than one point in the function domain. On the other hand, the solution of a MOP implies finding a trade-off among all the objective functions, and consequently, it leads to find an entire set of points instead of a single solution. The above occurs when functions are in conflict with each other (see e.g. Figure 2.1); thus, while we decrease one of the objective functions, some of the others increase their value and *vice versa*.

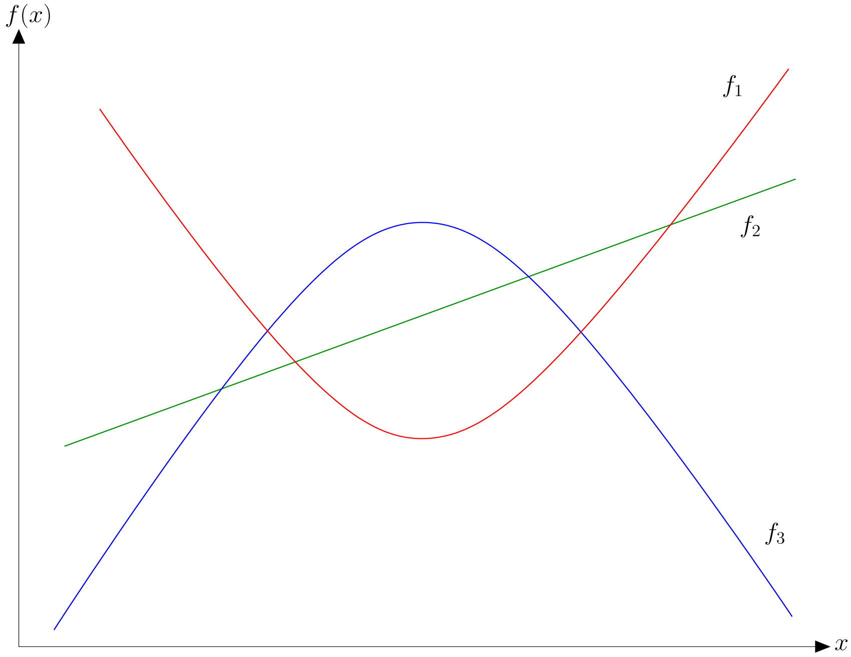


Figure 2.1: Functions in conflict, values of functions f_1, f_2 and f_3 have different behavior with respect to x.

One important characteristic of MOPs is that their solution sets, the *Pareto Set*s (PSs), typically form a $(k-1)$-dimensional object where k is the number of objectives involved in the MOP. Thus, it is only possible to approximate the entire set of compromise solutions for a relatively low number of objectives (say, $k = 3$ or 4). However, the vast majority of the time it is not easy to get such approximation. Therefore, it is important to provide numerical methods to obtain subsets of representative solutions.

Further, in a real world problem, the objective functions commonly have distinct units making them incomparable to each other. For example, we can not compare a cost function versus a quality function. Even if all functions have the same measurements, there is not a *total order* for vectors but only a *partial order*.

A *total order* in $\mathbb{R}$ refers to that, for all $a, b \in \mathbb{R}$, it is always possible to know if $a \leq b$. We can define a *partial order* for vectors as follows, given $c, d \in \mathbb{R}^k$ we say that $c \leq d \iff c_i \leq d_i \; \forall i \in \{1, \ldots, k\}$. Therefore, we need a different way to compare two vectors based on the values of the objectives.

For multiobjective optimization, the most commonly adopted method to compare solutions is the one called *Pareto dominance relation*. This notion of optimality takes into account the aspects that we have considered intuitively in this section. It was originally proposed by Francis Ysidro Edgeworth in 1881 [Edgeworth, 1881] and was later generalized by Vilfredo Pareto in 1896 [Pareto, 1896].

In order to formalize the above, we will introduce some definitions.

2.2.1 Definitions

We have two principal spaces when considering MOPs. The first one is called *decision space*. This is the space formed by the variables $x \in \mathbb{R}^n$ of the problem; according to our notation, this space is within the $\mathbb{R}^n$. The second one is called *objective space* and it is formed by the values $F(x)$. The image of a decision vector is in $\mathbb{R}^k$. When we solve a MOP, we must do comparisons between images of decision vectors, namely, we must do its comparison in objective space.

Definition 2.4 (Pareto dominance). *A point $y \in \mathcal{X}$ is dominated by a point $x \in \mathcal{X}$ if $f_i(x) \leq f_i(y) \; \forall \; i \in \{1, \ldots, k\}$ and $f_j(x) < f_j(y)$ for some $j \in \{1, \ldots, k\}$. In this case, we use the notation $x \prec y$; otherwise, we say that y is non dominated by x.*

Definition 2.5. *Let $x^* \in \mathcal{X}$ be a feasible point of* (2.12)*, x^* is called* ***weakly Pareto optimal*** *if $\nexists \; x \in \mathcal{X}$ s.t. $f_i(x) < f_i(x^*) \; \forall \; i = 1, \ldots, k$.*

Definition 2.6. *A decision vector $x^* \in \mathbb{X}$ is* ***Pareto optimal*** *with respect to* (2.12) *if there does not exist another decision vector $x \in \mathcal{X}$ such that $x \prec x^*$.*

Definition 2.7. *A point $x^* \in \mathcal{X}$ is locally (weak) optimal if it is (weak) optimal in a feasible neighborhood of x^*.*

In Figure 2.2, we show a representation of a set of *compromise solutions* for two functions, $f_i(x)$ and $f_2(x)$, which depend of the variable $x \in \mathbb{R}$; each function f_i takes its minimum value at x_i^*. As we can see, the line segment connecting x_1^* and x_2^* forms the Pareto set.

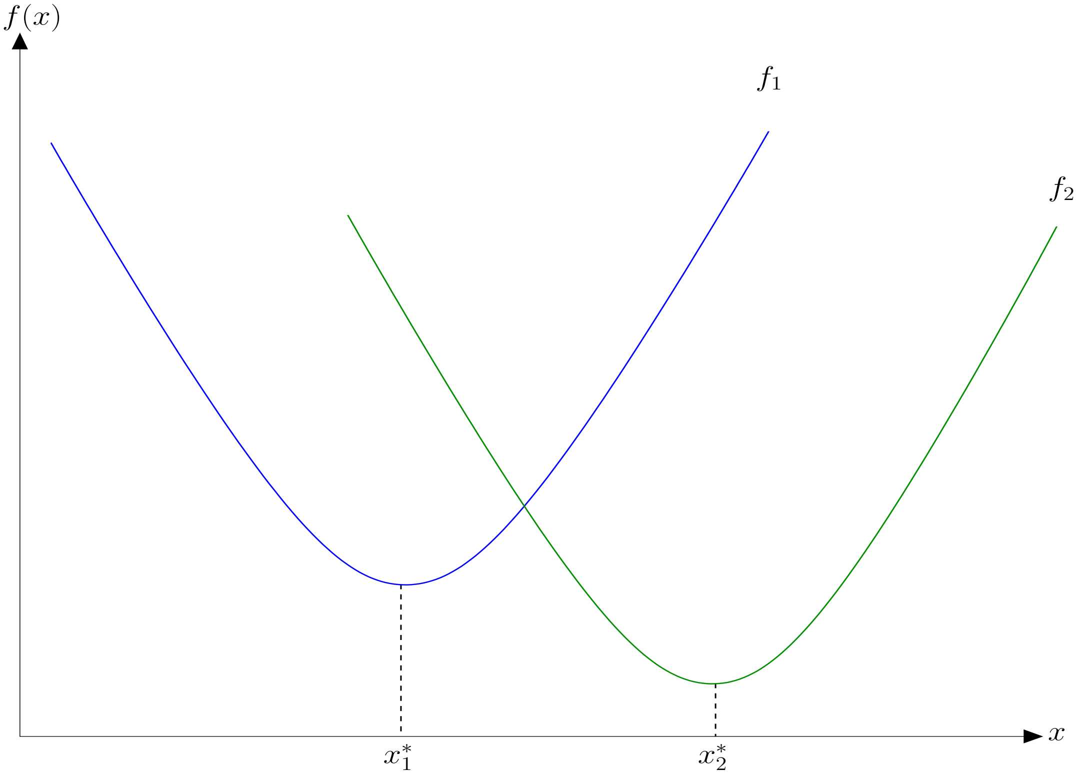


Figure 2.2: We illustrate the idea of compromise solutions. In this graph, we have optimal values in the interval $[x_1^*, x_2^*]$, as we can see, values of functions in this interval present a trade-off for both.

We can see in Figure 2.3 examples of Pareto dominance, weakly Pareto optimality, and Pareto optimality. This plot represents a surface in objective space for two objective functions, $f_1(x)$ and $f_2(x)$. The points p_2 and p_3 are Pareto optimal. As we can see, there is no other point with lesser value in both components; the point p_1 is weakly Pareto optimal because the point p_2 dominates it, even so, p_1 is not dominated by the point p_3; for its part, the point p_4 is dominated by the other three points.

Definition 2.8. *The set of optimal points $\mathcal{P}$ for* (2.12)*,*

$$\mathcal{P} = \{x \in \mathcal{X} \mid \nexists\, y \in \mathcal{X} : y \prec x\},$$

is called the PS.

Definition 2.9. *The set of images $\mathcal{F}$ of $\mathcal{P}$,*

$$\mathcal{F} = F(\mathcal{P}) = \{F(x) \in \mathbb{R}^k \mid x \in \mathcal{P}\},$$

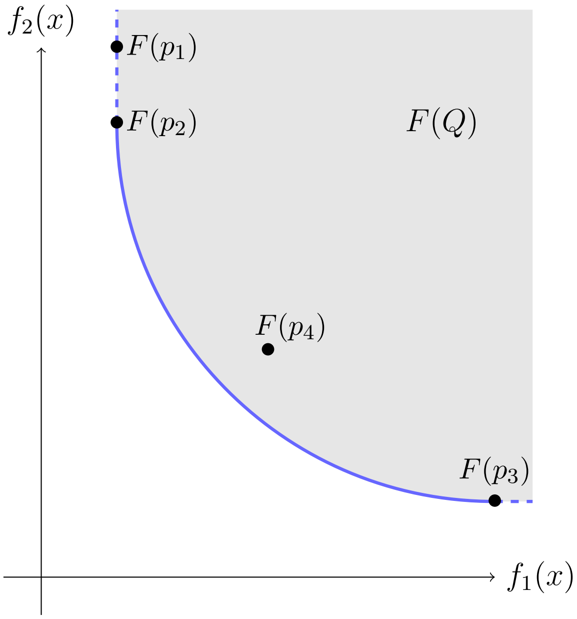


Figure 2.3: We show in this figure a $2D$ plot in objective space with four points, p_1 is a weakly Pareto optimal point, p_2 and p_3 are Pareto optimal points, and finally, p_4 is a dominated point in the Pareto sense.

is called the Pareto Front (PF).

Examples of PS and PF are shown in Figure 2.2 and Figure 2.3, respectively. In Figure 2.2 the PS is the line segment connecting x_1^* and x_2^*. Whereas that, in Figure 2.3, the PF is represented by the curve from $F(p_2)$ to $F(p_3)$, the points in the dashed line are weakly optimal points, (notice that $f_1(p_1) = f_1(p_2)$); and the point p_4 is dominated by p_2 and p_3 (actually, it is dominated by any point in the continuous line).

2.2.2 Optimality Conditions

A first order condition of optimality for differentiable MOPs is given by the *Karush-Kuhn-Tucker* (KKT) equations, named after the work of Karush [Karush, 1939] and Kuhn and Tucker [Kuhn and Tucker, 1951].

Theorem 2.1. *Suppose that x^* is a local solution of* (2.12)*. Then, there exist Lagrange multipliers $\alpha \in \mathbb{R}^k$, $\lambda \in \mathbb{R}^p$ and $\gamma \in \mathbb{R}^m$ such that the following conditions are* *satisfied*

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x^*) + \sum_{i=1}^{p} \lambda_i \nabla h_i(x^*) + \sum_{i=1}^{m} \gamma_i g_i(x^*) = 0 \tag{2.13a}$$

$$h_i(x^*) = 0, \quad i = 1 \dots p, \tag{2.13b}$$

$$g_i(x^*) \leq 0, \quad i = 1 \dots m, \tag{2.13c}$$

$$\alpha_i \geq 0, \quad i = 1 \dots k, \tag{2.13d}$$

$$\sum_{i=1}^{k} \alpha_i = 1, \tag{2.13e}$$

$$\gamma_i \geq 0, \quad i = 1 \dots m, \tag{2.13f}$$

$$\gamma_i g_i(x^*) = 0, \quad i = 1 \dots m. \tag{2.13g}$$

Finally, an important aspect in the context of our work is that Pareto points typically form a $(k-1)$-dimensional differentiable manifold [Hillermeier, 2001]. This and subsequent applications will be subject to an in-depth discussion throughout the thesis.

2.3 Solving a MOP

A large variety of methods has been developed to solve MOPs. These methods try to get a set of optimal solutions and they focus on two principal aspects. On the one hand, it is important to generate points in all the PF, that is, a good extension; on the other hand, an uniform distribution along the PF is desirable. Nevertheless, as we will show in Section 2.3, according to the problem, it is sometimes necessary to adopt a different approach to solve a MOP.

In this section, we describe the most important methods related to the solution of MOPs. First, we focus on some Mathematical Programming techniques for MOPs. After that, we describe some on Multi-objective Evolutionary Algorithms (MOEAs) that are usually used to deal with MaOPs and with constrained MOPs (CMOPs), as well as some MOPs in the literature related with the diversity in variable space.

2.3.1 Mathematical Programming Techniques

Continuation Methods

Continuation methods have been used to solve MOPs. These methods have the advantage that they perform a movement along a set of interest. To achieve this, we need an initial optimal solution. Starting from this point, we compute a predictor, which is a movement according to certain criteria; and then, we correct this point to the solution set leading to a new candidate solution. The consideration of both the predictor and in the corrector, gives rise to different methods.

Method of Hillermeier Many predictor-corrector methods are based on the *Implicit Function Theorem* [Krantz and Parks, 2002]. We briefly state the main steps of classical predictor-corrector methods for tracing one-dimensional solution sets. According to this theorem if x is a solution of

$$H(x) = 0, \tag{2.14}$$

where $H : \mathbb{R}^{N+1} \to \mathbb{R}^N$ and $rank(H'(x)) = N$, then there exists a value $\epsilon > 0$ and a curve $c : (-\epsilon, \epsilon) \to \mathbb{R}^{N+1}$ such that $c(0) = x$ and

$$H(c(s)) = 0 \quad \forall\, s \in (-\epsilon, \epsilon). \tag{2.15}$$

Differentiating we obtain

$$H'(c(s))c'(s) = 0. \tag{2.16}$$

This means that we can get the tangent vectors $c'(s)$ computing the kernel vectors of $H'(x)$, This can be done via a QR factorization of the matrix $H'(x)^T$, i.e.

$$H'(x)^T = QR \tag{2.17}$$

for an orthogonal matrix $Q \in \mathbb{R}^{(N+1)\times(N+1)} = (q_1, \ldots .q_{N+1})$ and a right upper triangular matrix $R \in \mathbb{R}^{(N+1)\times N}$. Doing so, the last column vector q_{N+1} is a kernel vector.

Finally, the orientation of the curve can be controlled be monitoring the sign of

$$\det \begin{pmatrix} H'(x) \\ q_{N+1}^T \end{pmatrix}. \tag{2.18}$$

Thus, we can compute a *predictor* point p along the linearized solution curve following the same orientation. Now we can get back to a curve $c(x)$ using (2.14) and p as starting point e.g. via a Gauss-Newton or a Levenberg-Marquardt method [Björck, 1996].

A method was developed by Hillermeier in 2001 [Hillermeier, 2001], for the multiobjective optimization context by considering the auxiliary function $\hat{F} : \mathbb{R}^{n+k} \to$

$\mathbb{R}^{n+1}$,

$$\hat{F}(x,\alpha,\lambda) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) + \sum_{i=1}^{p} \lambda_i \nabla h_i(x) \\ h(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix} = 0. \tag{2.19}$$

The set of KKT points of a non linear equality constrained is contained in the zero set of $\hat{F}$, which motivates the continuation along $\hat{F}^{-1}(0)$. We show a geometrical idea of this method in Figure 2.4.

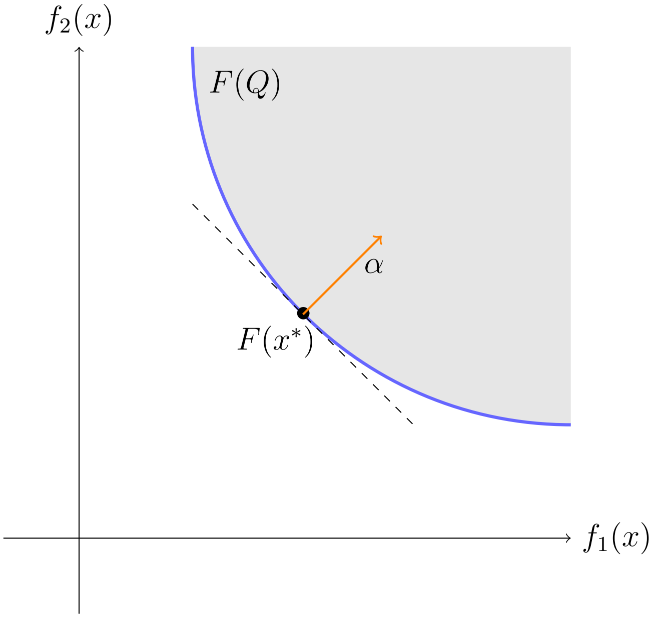


Figure 2.4: We show in this figure the α vector, which is orthogonal to the linearization PF at the point $F(x^*)$ (dotted line).

The Hillermeier method proceeds as the general technique described above, but instead of computing the determinant given in (2.18), the author checks the following condition for two consecutive solutions

$$[x_i - x_{i+1}]q \geq 0, \tag{2.20}$$

where $x_i, x_{i+1} \in \mathbb{R}^{n+k+p}$ and q is the tangent vector.

Also, the author suggests a step size which guarantees a uniform spread of the solutions on the PF. That is, for two consecutive solutions we want

$$\|F(x_i) - F(x_{i+1})\| \approx \tau, \tag{2.21}$$

where $\tau > 0$ is the desirable spread. The suggested step size is given by

$$t = \frac{\tau}{\|J\nu\|}, \tag{2.22}$$

where $\nu = x_{i+1} - x_i$.

Directed Search Predictor-Corrector Method This method defines a way to steer the search into any given direction in objective space [Schütze et al., 2011]. The main idea of this method is as follows.

Let be $x_0 \in \mathbb{R}^n$ a point in parameter space with $rank(J(x_0)) = k$ and $d \in \mathbb{R}^k$ a given vector which represents a desired search direction in image space. Then, a search direction $\nu \in \mathbb{R}^n$ in decision space is sought such that for $y_0 := x_0 + t\nu$, where $t \in \mathbb{R}_+$ is the step size (i.e., y_0 represents a movement from x_0 in direction v), it holds:

$$\lim_{t \to 0} \frac{f_i(y_0) - f_i(x_0)}{t} = \langle \nabla f_i(x_0), \nu \rangle = d_i, \qquad i = 1, ..., k. \tag{2.23}$$

Using the Jacobian of F, Eq. (2.23) can be stated in matrix vector notation as

$$J(x_0)\nu = d. \tag{2.24}$$

Hence, such a search direction ν can be computed by solving a system of linear equations. Since typically the number of decision variables is (much) higher than the number of objectives for a given MOP, i.e., $n \gg k$, system (2.24) is (probably highly) underdetermined, which implies that its solution is not unique. One possible choice is to take

$$\nu_+ = J(x_0)^+ d, \tag{2.25}$$

where $J(x_0)^+ \in \mathbb{R}^{n \times k}$ denotes the pseudo inverse[1] of $J(x_0)$. A new iterate x_1 can be computed as the following discussion shows: given a candidate solution x_0, a new solution is obtained via $x_1 = x_0 + t\nu$, where $t > 0$ is a step size and $\nu \in \mathbb{R}^n$ is a vector that satisfies (2.24). Among the solutions of system (2.24), ν_+ is the one with the smallest Euclidean norm. Hence, given t, one expects for a step in direction ν_+ (decision space) the largest progress in d-direction (objective space).

Predictor Given a KKT point $x_0 \in \mathbb{R}^n$, it is known that its associated weight vector α is orthogonal to the linearized Pareto front at $F(x_0)$ [Hillermeier, 2001] and hence any direction orthogonal to α could be a promising predictor direction. To compute such a direction a QR factorization on α can be performed:

$$\alpha = QR = (q_1, \dots, q_k)(r_{11}, 0, \dots, 0)^T, \tag{2.26}$$

where $Q \in \mathbb{R}^{k \times k}$ is an orthogonal matrix and $R \in R^{k \times 1}$ with $r_{11} \in \mathbb{R} \backslash \{0\}$ is an upper triangular matrix. Since by Eq. (2.26) $\alpha = r_{11} q_1$, it follows that a well spread set of directions can be taken from any of the normalized search directions ν_i such that:

$$J(x_0)\nu_i = q_{i+1}, \quad i = 1, \dots, k-1. \tag{2.27}$$

[1] If the rank of $J := J(x_0)$ is k (i.e., maximal) the pseudo inverse is given by $J^+ = J^T(JJ^T)^{-1}$.

To orientate the curve (i.e., to determine the signum of p) the change in one of the objective values can be used. For this, the signum of the according entry of the the direction vector q_2 can be taken. If, for instance, an improvement according to f_2 is sought, then

$$p = x_0 - sgn(q_{22}) \frac{t\nu_2}{||v_2||}, \tag{2.28}$$

where t is the chosen step size.

Corrector Given a predictor p, the subsequent solution along the curve can be computed by solving

$$\begin{aligned} &x(0) = x_0 \in \mathbb{R}^n \\ &\dot{\vec{x}}(t) = J(x(t))^+ d,\ t > 0. \end{aligned} \tag{2.29}$$

Using p as initial value and choosing $d = -\alpha_0$, i.e. the negative of the weight from the previous solution x_0, leading to a new solution x_1. The new associated weight vector α_1 can be updated as follows:

$$\alpha_1 = \min_{\lambda \in \mathbb{R}^k} \left\{ \left\| \sum_{i=1}^{k} \lambda_i \nabla f_i(x) \right\|^2 \text{ s. t. } \lambda_i \geq 0, i = 1, \ldots k, \sum_{i=1}^{k} \lambda_i = 1 \right\}. \tag{2.30}$$

Figure 2.5 displays a single iteration of the DS method, p stands for the predictor direction, while c stands for the corrector direction. It can be seen from the images that even though a movement along a linearization of the Pareto front at $f(x_0)$ is desired, it is not always possible to move in such direction and, in consequence, predictors usually end up above the Pareto front, making the use of corrector steps necessary in most cases.

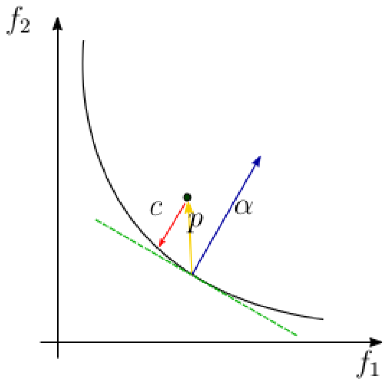


Figure 2.5: Example of the Directed Search predictor-corrector method.

Reference Point Methods

Usually, the solution of a MOP involves finding a set of optimal points with certain properties, e.g. that the image of this set has a uniform spread along the whole PF of the given problem. However, there are real-world problems in which a *Decision Maker* (DM) has knowledge about the problem or he/she wants to obtain optimal solutions with certain characteristics instead of solutions in the whole PS. The *reference point methods* are useful for these scenarios. The idea is to get the closest solution to a given vector, usually infeasible, which is a guess of the DM. These methods, where the DM has an active participation in the solution process, are called interactive. The difference between an interactive method and the others relies on the kind of information asked to the DM [Miettinen, 1999].

Reference Point Problem We can find different alternatives, which consider one reference point at the same time, in order to get a solution. An example is the classic reference point method, which was presented by Wierzbicki [Wierzbicki, 1981] in 1981. This method uses a given reference point, that represents the preferences of the DM, to solve a SOP. The solution of the SOP is presented to the DM and, if the solution is not good enough for the DM, then a new reference point is proposed. The process continues until the DM is in agreement with the solution.

The SOP in the reference point method employs an *achievement function*, which we will define in the next section. Other methods consider different ways to use a reference point, e.g, light beam search [Jaszkiewicz and Słowiński, 1999] and GUESS [Buchanan, 1997].

Achievement Functions Most of the *achievement scalarization functions* are based on the Tchebycheff metric. For this work, we the the so-called *Wierzbicki's achievement scalarizing function* (WASF) an appropriate achievement scalarizing function. Given a reference point Z, the WASF is defined as follows:

Definition 2.10.

$$g(x|Z,\lambda) := \max_{i=1,\ldots,k}\{\lambda_i|f_i(x) - Z_i|\} + \rho\sum_{i=1}^{k}\lambda_i(f_i(x) - Z_i), \tag{2.31}$$

where the parameter ρ is the so-called augmentation coefficient, *that must be set to a small positive value, and $\lambda = (\lambda_1, \ldots, \lambda_k)$ is a vector of weights, such that $\forall i\ \lambda_i \geq 0$ and, for at least one i, $\lambda_i > 0$.*

The exploration of the objective space, in most of the reference points methods, is made by moving the reference point at each iteration. That is, weights do not define preferences, but they are mainly used for normalizing each objective function. Usually, the weights are set as:

$$\lambda_i = \frac{1}{z_i^{\text{nad}} - z_i^{\text{utp}}},$$

where z^{nad} and z^{utp} are the *nadir* and the *utopian* point, respectively.

The utopian point is a vector formed by the minimum of each objective function, $z_i^{\text{nad}} = \min f_i(x) \mid x \in \mathcal{X}\ \forall i \in \{1, \ldots, k\}$; this point is generally infeasible. The nadir point is a vector formed by the maximum of each objective function on the PF, $z_i^{\text{utp}} = \max f_i(x) \mid x \in \mathcal{P}\ \forall i \in \{1, \ldots, k\}$. Typically, the estimation of the nadir point is more complicated than the estimation of the utopian point.

The weighted Tchebycheff scalarizing function poses some convenient properties over other scalarizing functions. As it is proved in [Miettinen, 1999] by using the augmented version of this function we can find any Pareto optimal solution. It is important to mention that the DM can provide both feasible and infeasible reference points.

2.3.2 Interactive Methods

For its part, a different class of interactive methods, the learning-oriented methods, exploit the preferences of the DM to direct the search and reduce the number of solutions to consider. Such methods are useful when the set of optimal solutions is very large, which is given for MaOPs. A wide variety of these interactive methods have been developed in recent years [Branke et al., 2008].

Pareto Navigator Method

The Pareto Navigator [Eskelinen et al., 2010] is an interactive learning-oriented method for nonlinear multiobjective optimization, which uses a set of optimal solutions to create a polyhedral approximation of the PF. The DM can direct the search along this polyhedral approximation according to her/his preferences. Once an interesting region has been identified, the DM can continue with another method to get an optimal solution.

It is important to stress that the navigation is not along optimal solutions and that we need a set of initial optimal solutions to use this method, which is composed by two principal phases: the *initialization* and the *navigation*.

1. **Initialization**. Given a set of optimal solutions, we construct a polyhedral in objective space.

2. **Navigation**. According to the preferences of the DM, we steer along the polyhedral until reaching a good zone for the DM. Then, we can use an achievement function to get the closest optimal solution to this zone.

If the DM is not satisfied, we repeat the process, adding the new solution of the initialization phase.

NIMBUS Method

NIMBUS [Miettinen and Mäkelä, 2000] is a system for non-linear optimization problems. It has, as its main characteristic, an on-line GUI to solve MOPs. The GUI allows the user to easily define preferences for the search. The method used by this system to solve the MOP is an evolutionary algorithm, so it is a robust system.

Also, this system provides to the user with different kinds of plots to visualize results for problems with many objective functions.

Nautilus Method

This is an interactive method, which is based on the assumption that the DM prefers getting better solutions at each iteration [Miettinen et al., 2010] instead of sacrificing the value of some function. For this reason, the interactive process of this method starts at the nadir point and, based on the preferences of the user, all the objectives can be improved at each iteration until reaching an optimal solution.

2.3.3 Many-objective Optimization and Evolutionary Algorithms

The notion of many-criteria optimization was used for the first time in [Farina and Amato, 2002]. Mathematically, a MaOP is defined in the same way as in Equation (2.12) but with $k > 3$ objectives. Thus, the methods described in Section 2.3 can be satisfactorily applied to MaOPs, since these mathematical techniques compute only one solution at each execution.

However, another approach to work with high dimensional problems consist in using *Evolutionary Algorithms*. This approach considers Genetic Algorithms and other population-based algorithms. A thorough survey on multiobjective evolutionary algorithms for MaOPs can be found in [von Lücken et al., 2014]. We can classify the methods for the treatment of MaOP in two groups. We can briefly describe these groups as follows:

i) Methods that adopt alternative preference relations:

- Crisp. An example of this kind of method is provided in [Di Pierro et al., 2007]. In this work, the authors propose the use of *Preference Ordering*, a generalization of Pareto optimality which employs two more stringent definitions of optimality: *Efficiency of Order* and the *Efficiency of order k with degree z*, as a ranking criterion in the framework of NSGA-II [Deb et al., 2002]. This approach was validate with problems having up to 8 objective functions.

- Fuzzy. A fuzzy relation is introduced in [Farina and Amato, 2002]; it is based on the number of components: bigger, smaller and equal between two vectors. We can find in this paper an expression for the portion e in a k-dimensional criteria domain, such that the dominance concept classifies as equivalent solutions, $e = \dfrac{2^k - 2}{2^k}$. Thus, the definition of Pareto optimality is not effective for MOPs.

ii) Methods that transform the original MaOP into a SOP:

- Based on scalarization functions,
  - Decomposition based. The most famous method is MOEA/D [Zhang and Li, 2007]; this method decomposes the original problem into a set of scalar optimization problems, and a scalarization function with different weights for each individual is assigned. In the original paper, the method is tested on problems with up to 4 objective functions.

- Indicator based. A method to approximate the value of the *Hypervolume* (HV) indicator was developed in [Bader et al., 2010]. The idea is to use a Monte Carlo algorithm to estimate values of the HV for a large number of objectives.

- Based on dimensional reduction techniques. Two kinds of objective reduction, *linear objective reduction* and *nonlinear objective reduction*, are presented in [Saxena et al., 2013]. Both are applied in an *a posteriori manner* to get a set of nondominated solutions with some MOEA. The core idea is to consider the correlation of the solution via an eigenvalue analysis to identify the set of important objectives. Tests of this approach include problems with up to 50 objective functions.

- Based on space partitioning. ϵ Ranking-Evolutionary Multiobjective Optimizer (ϵR-EMO) [Aguirre and Tanaka, 2009]; this method takes the basis of the NSGA-II [Deb et al., 2002]. It uses a partition strategy to define a schedule of subspace sampling and an adaptive ϵ-ranking procedure to re-rank solutions in each subspace. The number of solutions to be considered at each partition and the number of generations before creating a new partition are both set by the user.

Next, we provide descriptions of some specific approaches corresponding to the above taxonomy.

Indicator based

- S-metric Selection-EMOA (SMS-EMOA) [Beume et al., 2007]. The aim of this method is to maximize the HV. The former selection criterion is the non-dominated sorting procedure and the latter one is the HV. If the change of certain individual by a new one improves the HV, then this change is preserved.

- Hypervolume Estimation Algorithm for Multi-objective Optimization (Hype) [Bader and Zitzler, 2011]. As in [Bader et al., 2010], the idea is to approximate the HV indicator. Hype uses the concept of environmental selection to create a new population from the best solutions in the union set of the parent and offspring populations; this allows us to estimate the HV value by sampling solutions in different fronts. In this work, test problems with up to 50 objectives are considered.

Large Populations

- A survey about MOEA/D and NSGA-II with large population, e.g. 10,000 individuals, is presented in [Ishibuchi et al., 2009].

- Dynamical Multiobjective Evolutionary Algorithm (DMOEA) [Zou et al., 2008]. This method is based on the principle of the minimal free energy in thermodynamics. The method defines a fitness function, which considers three aspects: the Pareto rank value of the individual, a function analog to the temperature and the crowding distance [Deb et al., 2002].

- Grid-Based Evolutionary Algorithm (GrEA) [Yang et al., 2013]. This method tries to strengthen the selection pressure toward the optimal direction while maintaining an extensive and uniform distribution among solutions with the help of a grid. Three grid-based criteria, based on grid dominance and grid difference, are included to compute the fitness of individuals.

Dimension Reduction

- Pareto Corner Search Evolutionary Algorithm (PCSEA) [Singh and Ray, 2011]. This algorithm does a dimensionality reduction searching corners of the Pareto front. The authors identify two classes of corners, and the minimization is made using the L_2 norm. Solutions that minimize either one of the objectives or the rest of the objectives simultaneously are preferred. The dimensionality analysis for the reduction is performed using a heuristic technique, which considers a rate between the number of non-dominated solutions in a reference set and the number of non-dominated solutions of such set after omitting specific individuals.

2.3.4 MOEAs for CMOPs

An important part of this work are the constrained MOPs, for this reason, we consider four state-of-the-art MOEAs in order to compare our proposed approaches. These MOEAs incorporate different constraint-handling strategies in their environmental selection procedures.

- NSGA-II. The popular non-dominated sorting genetic algorithm II [Deb et al., 2002] was adopted in our comparative study. NSGA-II employs a binary tournament based on feasibility in the mating selection procedure. In order to determine the next generation, the crowding comparison operator considers the feasibility of solutions.
- GDE3. The third evolution step of generalized differential evolution [Kukkonen and Lampinen, 2005] was also adopted in our experimental analysis. GDE3 introduces the concept of constraint-domination explained before to discriminate solutions.
- cMOEA/D-DE. We also adopted the first version of the multiobjective evolutionary algorithm based on decomposition for constrained multi-objective optimization [Jan and Zhang, 2010]. cMOEA/D-DE utilizes a penalty function in order to satisfy the constraint of the problem. The penalty function is added to the scalarizing function employed by MOEA/D-DE [Li and Zhang, 2009] in a straight forward manner to approximate the PF of a constrained MOP.
- eMOEA/D-DE. A version of MOEA/D-DE based on the ε-constraint method for constrained optimization [Martinez and Coello, 2014] is also adopted in our experimental study. eMOEA/D-DE employs the ε-constraint method to satisfy the constraints of the problem by obtaining information about feasible solutions in the neighborhood of MOEA/D-DE. Thus, the neighboring solutions are used to defined the ε-constraint value which is dynamically adapted during the search process of eMOEA/D-DE.

2.3.5 Diversity in Decision Space

Unlike evolutionary algorithms for single objective optimization problems (SOPs), maintaining diversity in decision space is not a priority for most MOEAs; even the performance indicators are developed in order to measure the accuracy based only on the objective function (e.g., the hypervolume [Zitzler and Thiele, 1999] and the DOA [Dilettoso et al., 2017]). As an exception, there is the Δ_p indicator [Schütze et al., 2012, Bogoya et al., 2018], which can be viewed as an averaged Hausdorff distance and which actually measures the distance between two general sets. For this reason, we can use it as indicator both in objective space as well as in decision space.

Although works that explicitly consider at the same time variables and objectives are scarce, one can find some related work on this topic. For instance, the NSGA [Srinivas and Deb, 1994] (the algorithm that precedes the well-known NSGA-II) uses fitness sharing in decision space. In [Jeffrey et al., 1993], some possible techniques are proposed to spread out solutions, both in objective and decision decision space: *pointwise expansion*, *threshold sharing*, *sequential sharing*, *simultaneous sharing multiplicative*, and *simultaneous sharing additive*. It is important to point out that the above approaches are only part of the discussion of the paper and they were not implemented; the implemented algorithm was the Niched Pareto GA, a method with phenotypic sharing. Besides, all of the described techniques depend on the normal fitness sharing method, that is, two additional parameters must be provided or approximated (the niche radius σ_{share} in each space).

The omni-optimizer algorithm [Deb and Tiwari, 2008] is proposed as a procedure that aims at solving a wide variety of optimization problems (single or multi-objective and uni- or multi-modal problems). The authors argue that, to solve different kinds of problems, it is necessary to know different specialized algorithms. Thus, it is desirable to have an algorithm which adapts itself for handling any number of conflicting objectives, constraints, and variables. The omni-optimizer is important in the context of this work as it uses a two-tier fitness assignment scheme based on the crowding distance of the NSGA-II. The primary fitness is computed using the phenotypes (objectives and constraint values) and the secondary fitness is computed using both phenotypes and genotypes (decision variables). The modified crowding distance computes the average crowding distance of the population, both in objective and decision variable space. If the crowding value for some individual is above average (in any space), it is assigned the larger of the two distances; else the smaller of the two distances is assigned. However, omni-optimizer has a more general purpose.

An algorithm that explicitly promotes the diversity of the decision space is the *MOEA/D with Enhanced Variable-Space Diversity (MOEA/D-EVSD)*, proposed in [Castillo et al., 2017]. This method is an extension of the MOEA/D [Zhang and Li, 2007], but with an enhanced variable-space diversity control. In the first generations, the MOEA/D-EVSD tries to induce a larger diversity via promoting the mating of dissimilar individuals. Similarly to MOEA/D, a new individual is created for each subproblem. Then, instead of randomly selecting two individuals of the neighborhood, a pool of α candidate parents is randomly filled from the neighborhood with probability δ, whereas it is randomly selected from the whole population with probability $1-\delta$.

Thus, the two selected parents are the ones that have the largest distance. As the δ parameter is dynamically set, a gradual change between exploration and exploitation can be induced. Additionally, a final phase to further promote intensification is included, which is essentially a traditional MOEA/D coupled with Differential Evolution (DE) operators. For the last generations of MOEA/D-EVSD the traditional mating selection of MOEA/D is conserved together with the Rand/1/bin scheme for the DE operators. The authors of this paper show that, by inducing a gradual loss of diversity in the decision space, the performance of state-of-the-art of MOEAs can be improved.

2.4 Multiobjective Benchmark Problems

In recent years, researchers have developed challenging optimization test problems. Test problems are useful to evaluate characteristics of optimization algorithms and eventually help in the design of more efficient solvers. In [Deb, 2001], a procedure to design bi-objective unconstrained test problems via three functionals is proposed. These functionals control some features of the problem. In particular, six specific test problems are included (known as ZDT functions). The user is able to construct a new test problem if a certain function f_1 is chosen or a different variable mapping strategy is adopted: $y = M \cdot x$, where M is a constant matrix. This mapping strategy is applied in [Deb et al., 2002]. In [Deb et al., 2005], three different approaches for systematically designing test problems are presented and the DTLZ benchmark is introduced. Some main features of these test problems are the scalability in both the number of decision variables and the number of objective functions. Due to their construction, the user is able to control some difficulties such as convergence or distribution of the solution set. Later, in [Huband et al., 2005], the WFG test problems were proposed. This suite is scalable both in objective and decision variable space. Due to its design, the user is able to integrate different characteristics such as multi-modality and non-separability to the desired test problem. The authors included a wide variety of Pareto optimal geometries. Later, Deb, Sinha, and Kukkonen [Deb et al., 2006] introduced a test problem suite that uses linkages among variables. They considered three types of linkages. The first one, linkage among variables that affects either convergence or diversity. The second linkage affects both convergence and diversity. The third one, non-linear linkage, causes linear operators to have difficulties when preserving optimality of solutions. Finally, in [Ishibuchi et al., 2017] the authors modified the DTLZ benchmark to evidence design issues of some references based MOEAs to deal with different geometries of traditional problems.

Although several test problem suites have been designed, not all of them consider constraints. In [Deb et al., 2001], some test problems for constrained multi-objective optimization are proposed (CTP). In this proposal, the complexity of the constrained search space can be controlled. Its design causes two kinds of difficulties: (i) Difficulty near the Pareto front and (ii) difficulty in the entire search space. The test problem generator has six controllable parameters. By setting different parameter values, the user can create a new constrained test problem. Later, in [Zhang et al., 2008], the authors proposed more test instances that resemble complicated real-life problems to stimulate the MOEA research. The Adaptive Approach for Handling is proposed in [Jain and Deb, 2014], also in this work several constrained MOPs are stated. Such problems are classified in three different types: constrained problems of Type-1, where the original Pareto-optimal front is still optimal, but there is an infeasible barrier in approaching the Pareto-optimal front; constrained problems of Type-2, which introduce infeasibility to a part of the Pareto-optimal front, that is, they produce discontinuities in the Pareto-optimal fronts; and constrained problems of Type-3, they involve multiple constraints in such way that the constrained Pareto-optimal front is different to the Pareto-optimal front of the unconstrained problem. Recently, in [Fan et al., 2019a], the authors proposed a general toolkit to construct difficulty-adjustable and scalable constrained MOPs. The problems that can be constructed with this toolkit are obtained via the modification of a triplet of parameters, (η, ζ, γ), that are related with three primary difficulty types: convergence-hardness, diversity-hardness, and feasibility-hardness. The authors explain how to adjust the difficulty level for each primary difficulty type according with the values of the triplet.

It is important to note that the above proposals focus on inequality constraints. In particular, benchmark MOPs that contain equality constraints are scarce so far. For instance, in [Saha and Ray, 2012], equality constraints are imposed to ZDT problems in order to test a new MOEA that seeks for the preservation of feasible solutions. This suite is scalable neither in the number of objectives nor in the number of constraints. Further, for all constrained problems, the Pareto set is identical to the solution set of the corresponding unconstrained problem. Consequently, a search that neglects all constraints of the problem may lead to satisfying results.

The latter is, of course, an unwanted effect in the investigation of the ability of the algorithm to handle constraints. That is, since solving an equality constrained MOP implies satisfying all the equations, then in principle, it would be sufficient to explore the feasible region. So it is important that a good benchmark has a feasible region that differs from the optimal set. In this way, it is possible to evaluate the performance of the algorithms from two different points of view, how good they are to reach the feasible region and once in it, how effective they are to obtain the optimality without losing the feasibility.

2.5 Performance Indicators

When dealing with SOPs, it is easy to determine when one solution is better than another one. For doing that, we only look at the value of the objective function and if $f(a) < f(b)$, then, clearly, a is better than b. However, when dealing with MOPs, comparisons are not so easy, since we have to compare two sets that represent two different approximations to the PF. Performance indicators help us with this task, as they map an approximation set to a single value, which can then be compared.

In this section we present the performance indicators used along this work.

2.5.1 Δ_p indicator

The Δ_p indicator [Schütze et al., 2012, Bogoya et al., 2018, Bogoya et al., 2019] can be viewed as an averaged Hausdorff distance between an approximation set and the real Pareto front of a MOP. This indicator is defined by slight modifications of the indicators Generational Distance (GD) [Van Veldhuizen, 1999] and Inverted Generational Distance (IGD) [Coello and Cortés, 2005]. Formally, the Δ_p indicator can be written as follows.

Let $P = \{\vec{x}^1, \ldots, \vec{x}^{|P|}\}$ be an approximation and $R = \{\vec{r}^1, \ldots, \vec{r}^{|R|}\}$ be a discretization of the PF of a MOP, then

$$\Delta_p(P, R) = \max\{GD_p(P, R), IGD_p(P, R)\}, \tag{2.32}$$

where $GD_p(P, R) = \left(\frac{1}{|P|}\sum_{i=1}^{|P|} d_i^p\right)^{\frac{1}{p}}$ and $IGD_p(P, R) = \left(\frac{1}{|R|}\sum_{j=1}^{|R|} \hat{d}_j^p\right)^{\frac{1}{p}}$, and where d_i and $\hat{d}_j$ are the Euclidean distance from $\vec{x}^i$ to its closest member $\vec{r} \in R$, and the Euclidean distance from $\vec{r}^j$ to its closest member $\vec{x} \in P$, respectively. Here we have chosen $p = 2$. The ideal indicator value is 0, and a low Δ_p value indicates a good approximation of P. Recently, Rudolph et al. [Rudolph et al., 2016] have shown that for bi-objective problems the Δ_p indicator prefers evenly spread solutions around the Pareto front, and that this indicator thus complies with the terms "spread" and "convergence" as used in the EMO community.

Hypervolume Indicator

The Hypervolume (or S-metric) [Zitzler and Thiele, 1999] is the most commonly accepted indicator, as it is Pareto compliant. That is, let $A_1 = \{a, a_2, a_3, \ldots, a_j\}$ and $A_2 = \{b, a_2, a_3, \ldots, a_j\}$; if $b < a$, it follows that $\mathcal{H}(A_2) \geq \mathcal{H}(A_1)$. Where $\mathcal{H}$ is defined as follows [Coello et al., 2007].

Definition 2.11. *Let $y^{(1)}, \ldots, y^{(\mu)} \in \mathbb{R}^k$ be a non-dominated set and $r \in \mathbb{R}^k$ such that $y^{(i)} \prec r$ for all $i = 1, \ldots, \mu$. The value*

$$\mathcal{H}\left(y^{(1)}, \ldots, y^{(\mu)}; r\right) = \mathcal{L}\left(\cup_{i=1}^{\mu}\left[y^{(i)}, r\right]\right),$$

is termed the dominated hypervolume with respect to the reference point r, where $\mathcal{L}(\cdot)$ denotes de Lebesgue measure in $\mathbb{R}^k$.

This indicator measures the size of the space covered or dominated. The hypervolume is described as the Lebesgue measure $\mathcal{L}$ of the union of hypercubes defined by a non-dominated point $y^{(i)}$ and a reference point r. This union is expressed as $\cup_{i=1}^{\mu} \left[y^{(i)}, r\right]$. Note that $\mathcal{H}$ dependens on the selection of r. This performance indicator is computational expensive, as its estimated complexity is $\mathcal{O}(n^{k+1})$, where n is the number of decision variables and k is the number of objective functions.

2.5.2 Feasibility Ratio

The feasibility ratio (I_F) indicator refers to the ratio of the number of feasible solutions found in the final approximation P given by a MOEA. Mathematically, this indicator can bee stated as follows

$$I_F(P) = \frac{P_f}{|P|}, \tag{2.33}$$

where P_f denotes the number of feasible solutions in P and $|P|$ represents the cardinality of the population P.

2.5.3 Reference Inverted Generational Distance

In this work, we consider a modification of the Inverted Generational Distance (IGD_Z) [Mejía et al., 2017], which allows us to work with a set of reference points. More specifically, given the set Z of reference points and a reference set archive A, the distance of Z toward $F(A)$ is measured as follows:

$$IGD_Z(F(A), Z) := \frac{1}{|Z|}\textstyle\sum_{i=1}^{|Z|} \displaystyle\min_{j=1,\ldots,|A|} dist(Z_i^*, F(a_j)), \tag{2.34}$$

where Z_i^* denotes the point from the PF which is closest to Z_i, i.e., $\|Z_i - Z_i^*\| = dist(Z, F(P_Q))$. Hereby, $dist$ measures the distance between point and set and between two sets as $dist(u, A) = \inf_{v\in A} \|u - v\|$ and $dist(B, A) = \sup_{u\in B} dist(u, A)$, where u and v denote points from sets A and B. Like this, the optimal IGD_Z value is always zero. Notice, however, that the evaluation of the IGD_Z value requires the knowledge of the exact PF.

Other important performance indicators that are not considered in this work are, R2 [Brockhoff et al., 2012, Hansen and Jaszkiewicz, 1994, Zitzler et al., 2008], and the DOA [Dilettoso et al., 2017].

2.6 Pareto Tracer and Pareto Explorer

The next chapters of this work are highly related with the Pareto Tracer and Pareto Explorer, which are described in detail in the following.

2.6.1 Pareto Tracer Method

The idea of the Pareto Tracer method [Martín and Schütze, 2018] is to separate the decision and weight space as opposed to the Hillermeier method. In the compound (x, α, λ) space (see Equation 2.19), the non linearity may increase compared to the x-space. For instance, if the PS is linear, then the related solution set does not have to be linear in the compound space. Thus, when we separate x, α, and λ spaces the non linearity comes to decrease and it implies that a corrector step is not needed for linear PSs. The above also implies a reduction of the total computational cost. However, the most important aspect of this work about this separation is that this makes it possible to find a relationship between a direction in objective and decision space, which is a fundamental part of this work.

We first consider the following unconstrained MOP

$$\min_{x \in \mathbb{R}^n} F(x), \tag{2.35}$$

and describe the sequel of the predictor and the corrector steps.

Predictor

The typical task for the computation of predictor points in continuation methods is to determine the tangent space to the given set. In order to do this, we use the function $\hat{F}$ that is obtained from the KKT conditions

$$\hat{F}(x, \alpha) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix} = 0. \tag{2.36}$$

Differentiating $\hat{F}$ we obtain

$$\hat{F}'(x, \alpha) \begin{pmatrix} \nu \\ \mu \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x) & \nabla f_1(x) & \dots & \nabla f_k(x) \\ 0 & 1 & \dots & 1 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \end{pmatrix} = 0 \tag{2.37}$$

The second equation of (2.37) yields,

$$\sum_{i=1}^{k} \mu_i = 0. \tag{2.38}$$

Having a $\mu \in \mathbb{R}^k$ that satisfies (2.38) we obtain

$$\sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x)\nu = -\sum_{i=1}^{k} \mu_i \nabla f_i(x) = -J^T\mu, \tag{2.39}$$

and it is possible to find a relationship between ν and μ, i.e., a relationship between the objective space and the variable space:

$$\nu_\mu = -W_\alpha^{-1} J^T \mu, \tag{2.40}$$

where $W_\alpha \in \mathbb{R}^{n\times n}$ is given by

$$W_\alpha := \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x). \tag{2.41}$$

If $rank(J) = k-1$, we can compute the set of tangent vectors via a QR factorization of α

$$\alpha = QR = (q_1, q_2, \ldots, q_k)R, \tag{2.42}$$

where $q_i \in \mathbb{R}^k$ and $R \in \mathbb{R}^{k\times 1}$. Let Q_2 denote the matrix formed by the last $k-1$ columns vectors of Q

$$Q_2 = (q_2, \ldots q_k). \tag{2.43}$$

The column vectors of this matrix form an orthonormal basis of the linearized Pareto front at $F(x)$ (see Figure 2.6).

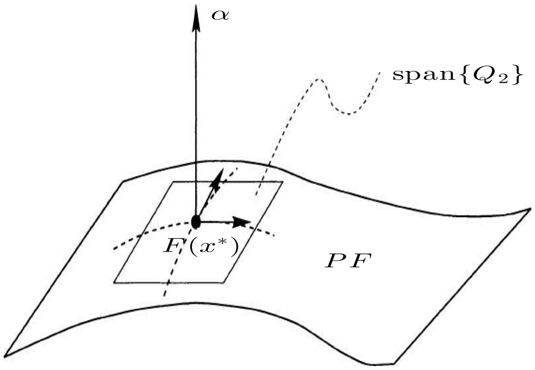


Figure 2.6: We show a plane formed by the tangent vectors of the PF at the point $F(x^*)$ via QR factorization.

Given a direction $\nu \in \mathbb{R}^n$ in decision space, the corresponding movement in objective space for infinitesimal step sizes is given by

$$d = J\nu. \tag{2.44}$$

The orientation of the movements along the tangent space is related to (2.44). Thus, the task is to find the proper orientation vector $\mu_d \in \mathbb{R}^k$ that satisfies

$$J\nu_{\mu_d} = d. \tag{2.45}$$

The vector ν_μ can be obtained with the vector μ_d that solves:

$$\begin{pmatrix} -JW_\alpha^{-1}J^T \\ 1\dots 1 \end{pmatrix} \mu_d = \begin{pmatrix} d \\ 0 \end{pmatrix}. \tag{2.46}$$

Hence, the predictor is given by the following expression,

$$p = x^* + t\nu_\mu, \tag{2.47}$$

where t is the step size which can be chosen, as for the Hillermeier method, by Equation (2.22).

Corrector

The goal of the corrector phase is to ensure that the resulting solution is on the efficient set.

The PT applies the Newton method for MOPs [Fliege et al., 2009] to realize the corrector step. The Newton direction is defined as the solution to:

$$\begin{array}{cc} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta \\ \text{s.a} & \nabla f_i(x)^T\nu + \frac{1}{2}\nu^T\nabla^2 f_i(x)\nu \leq \delta, \quad i = 1, ..., k, \end{array} \tag{2.48}$$

where δ serves as a measure of the expected decrease in objective space produced by a line search in direction ν in parameter space. An acceptable step size may be decided by a backtracking procedure with a modification of the (component-wise) Armijo condition.

Handling Equality Constrains

Now, we consider the following MOP:

$$\begin{array}{cc} \min\limits_{x\in\mathbb{R}^n} & F(x), \\ \text{s.t} & h_i(x) = 0, \quad i = 1, ..., p. \end{array} \tag{2.49}$$

In the presence of equality constraints, the following function has to be considered, that again evolves from the KKT systems; let $\tilde{F} : \mathbb{R}^{n+k+p} \to \mathbb{R}^{n+p+1}$,

$$\tilde{F}(x, \alpha, \lambda) = \begin{pmatrix} \sum_{i=1}^{k} \alpha_i \nabla f_i(x) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x) \\ h(x) \\ \sum_{i=1}^{k} \alpha_i - 1 \end{pmatrix} = 0. \tag{2.50}$$

Now, we can proceed as in the unconstrained case. We define:

$$W_{\alpha,\lambda} := \sum_{i=1}^{k} \alpha_i \nabla^2 f_i(x) + \sum_{j=1}^{p} \lambda_j \nabla^2 h_j(x) \in \mathbb{R}^{n\times n}, \tag{2.51}$$

and

$$H := \begin{pmatrix} \nabla h_1(x)^T \\ \vdots \\ \nabla h_p(x)^T \end{pmatrix} \in \mathbb{R}^{p\times n}. \tag{2.52}$$

Doing so, we can write $\tilde{F}'$ as:

$$\tilde{F}'(x, \alpha, \lambda) = \begin{pmatrix} W_{\alpha,\lambda} & J^T & H^T \\ H & 0 & 0 \\ 0 & 1, \dots, 1 & 0 \end{pmatrix}. \tag{2.53}$$

Predictor. In order to compute a kernel vector of (2.53), we consider $\nu \in \mathbb{R}^n$, $\mu \in \mathbb{R}^k$ and $\xi \in \mathbb{R}^p$, such that:

$$\begin{pmatrix} W_{\alpha,\lambda} & J^T & H^T \\ H & 0 & 0 \\ 0 & 1, \dots, 1 & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \\ \xi \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.54}$$

The choice of a vector μ which satisfies (2.46) allows to reduce (2.54) to:

$$\begin{pmatrix} W_{\alpha,\lambda} & H^T \\ H & 0 \end{pmatrix} \begin{pmatrix} \nu_\mu \\ \xi \end{pmatrix} = \begin{pmatrix} -J^T\mu \\ 0 \end{pmatrix}. \tag{2.55}$$

If $\text{rank}(W_{\alpha,\lambda}) = n$ and $\text{rank}(H) = p$, then the matrix on the left hand side is regular and so the solution of (2.55) is unique.

Corrector. For the corrector step, we need a modification of the Newton method for the given MOP. A suggestion is to modify the Newton direction via:

$$\begin{array}{lll} \min\limits_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta & \\ \text{s.a} & \nabla f_i(x)^T\nu + \frac{1}{2}\nu^T\nabla^2 f_i(x)\nu \leq \delta, & i = 1, ..., k. \\ & h_i(x) + \nabla h_i(x)^T\nu = 0, & i = 1, ..., p. \end{array} \tag{2.56}$$

The additional restriction arises from applying the Newton method to h. The following result shows how we can view this modification as a particular penalization method. The new penalized MOP is given by:

$$\min_{x\in\mathbb{R}^n} F_h : \mathbb{R}^n \to \mathbb{R}^k, \tag{2.57}$$

where $F_h = \left(f_1^h, \ldots, f_k^h\right)^T$, and

$$f_i^h(x) = f_i(x) + CP(x), \tag{2.58a}$$

$$P(x) = \frac{1}{2}\sum_{i=1}^{p} h_i(x)^2 = \frac{1}{2}\|h(x)\|^2. \tag{2.58b}$$

Proposition 2.1 ([Martín and Schütze, 2018]). *Let $x \in \mathbb{R}^n$ be given and the $f_i^{'}s$ be strictly convex, and (ν^*, δ^*) be a solution of* (2.56)*. Then*

- *(a) If $\nu^* = 0$, then $\delta^* = 0$ and x is a KKT point of* (2.49)*.*
- *(b) If $\nu^* \neq 0$ and $\delta^* < 0$, then ν^* is a descent direction of* (2.57) *for $C = 0$ (i.e. a descent direction of the unconstrained MOP* (2.12)*).*
- *(c) If $\nu^* \neq 0$ and $\delta^* \geq 0$, then $\|h(x)\|^2 \neq 0$ and ν^* is a descent direction of* (2.57) *for*

$$C > \frac{\max_{i=1,\ldots,k} \nabla f_i(x)^T \nu^*}{\|h(c)\|^2} \geq 0. \tag{2.59}$$

As we compute descent directions of (2.57), there are 3 possibilities: (i) we can improve F but not P, (ii) we can improve both, and (iii) we can improve P but not F. This is reflected in the step size control. A component-wise Armijo condition is used together with the following function:

$$\tilde{F}_h(x) = \begin{cases} F(x) & \delta < 0 \ \text{ y } \ \|h(x)\|^2 = 0 \\ (F(x), P(x))^T & \delta < 0 \ \text{ y } \ \|h(x)\|^2 \neq 0 \\ P(x) & \qquad \delta \geq 0 \end{cases} \tag{2.60}$$

Notice that if $\delta < 0$, then (i) or (ii) is satisfied. We know, by Proposition 2.1, that ν^* is a descent direction of F. If $\|h(x)\|^2 = 0$, then we can not further improve P, so this is not considered. Now, if $\|h(x)\|^2 \neq 0$, then F and P can be simultaneously decreased through a linear search in direction ν^*. If $\delta \geq 0$, then at least one of the objectives increases its value with the direction ν^*.

By Proposition 2.1, we have $\|h(x)\|^2 \neq 0$ and ν^* is a descent direction of (2.57) for some $C > 0$. The choice of a step size, which produces a *sufficient decrement* in

P, which depends on the value of $C \gg 0$. So, we take as an acceptable step size a $t \in \mathbb{R}_+$, which satisfies:

$$\tilde{F}_h(x - t\nu) \leq \tilde{F}_h(x) + ct\Delta\tilde{F}_h(x)\nu. \tag{2.61}$$

Here, δ is a measurement of the expected decrement of F in objective space and the derivative of P in the direction ν^* is given by:

$$h(x)^T H\nu^* = -\|h(x)\|^2 \leq 0.$$

Thus, we can use $-\|h(x)\|^2$ to measure the possible reduction as a penalty. The term $\Delta\tilde{F}_h$ of (2.61) represents the expected decrement of $\tilde{F}_h$ in objective space, and it is given by:

$$\Delta\tilde{F}_h(x) = \begin{cases} \delta e & \delta < 0 \text{ y } \|h(x)\|^2 = 0 \\ (\delta e, -\|h(x)\|^2)^T & \delta < 0 \text{ y } \|h(x)\|^2 \neq 0 \\ -\|h(x)\|^2 & \delta \geq 0, \end{cases} \tag{2.62}$$

where $e = (1, \ldots, 1)^T \in \mathbb{R}^k$.

Handling Inequality Constraints

We consider the box constrained case

$$\min_{l \leq x \leq u} \quad F(x), \tag{2.63}$$

where $l,\ u \in \mathbb{R}^n$ are the lower and upper bounds, respectively.

Predictor. We can include the set of active constraints as equality constraints to solve (2.63), i.e.

$$\begin{aligned} -x_i + l_i = 0, &\ \ i \in \mathcal{I}_l \\ x_i - u_i = 0, &\ \ i \in \mathcal{I}_u, \end{aligned}$$

where

$$\begin{aligned} \mathcal{I}_l &= \{i | -x_i + l_i > -\epsilon, \ \ i = 1, \ldots, n\} \\ \mathcal{I}_u &= \{i | x_i - u_i > -\epsilon, \ \ i = 1, \ldots, n\} \end{aligned}$$

for some $\epsilon \in \mathbb{R}_+$, and $\mathcal{I}_{l,u} = \{i | i \in \mathcal{I}_l \ \text{ o } \ i \in \mathcal{I}_u\}$. Hence $\tilde{F} : \mathbb{R}^{n+k+r} \to \mathbb{R}^{n+r+1}$ is given by

$$\tilde{F}(x, \alpha, \rho, \varrho) = \begin{pmatrix} \sum_{i=1}^k \alpha_i \nabla f_i(x) - \sum_{i \in \mathcal{I}_l} \rho_i e_i + \sum_{i \in \mathcal{I}_u} \varrho_i e_i \\ (-x_i + l_i)_{i \in \mathcal{I}_l} \\ (x_i - u_i)_{i \in \mathcal{I}_u} \\ \sum_{i=1}^k \alpha_i - 1 \end{pmatrix} = 0, \tag{2.64}$$

where e_i is the $i-th$ canonical vector and $r = |\mathcal{I}_{l,u}|$. We define $\mathcal{I}_{l,u} \in \mathbb{R}^{r\times n}$ as

$$\left[\mathcal{I}_{l,u}\right]_{j_i} = \left\{ \begin{array}{cc} -e_i^T & i \in \mathcal{I}_l \\ e_i^T & i \in \mathcal{I}_u \end{array} \right. , \quad j = 1, \ldots, r, \tag{2.65}$$

where $\left[\mathcal{I}_{l,u}\right]_{j_i}$ denotes the $j-th$ row of $\mathcal{I}_{l,u}$. Notice that

$$\tilde{F}^{'}(x, \alpha, \rho, \varrho) = \begin{pmatrix} W_\alpha & J^T & \mathcal{I}_{l,u}^T \\ \mathcal{I}_{l,u} & 0 & 0 \\ 0 & 1, \ldots, 1 & 0 \end{pmatrix}. \tag{2.66}$$

Now we must compute a kernel vector of (2.66). Let $\nu \in \mathbb{R}^n$, $\mu \in \mathbb{R}^k$ and $\eta \in \mathbb{R}^r$ be vectors, such that

$$\begin{pmatrix} W_\alpha & J^T & \mathcal{I}_{l,u}^T \\ \mathcal{I}_{l,u} & 0 & 0 \\ 0 & 1, \ldots, 1 & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \mu \\ \eta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}. \tag{2.67}$$

A vector μ, which satisfying (2.46), reduces (2.67) to

$$\begin{pmatrix} W_\alpha & \mathcal{I}_{l,u}^T \\ \mathcal{I}_{l,u} & 0 \end{pmatrix} \begin{pmatrix} \nu \\ \eta \end{pmatrix} = \begin{pmatrix} -J^T\mu \\ 0 \end{pmatrix}. \tag{2.68}$$

If $\text{rank}(W_\alpha) = n$ and $\text{rank}(I_{l,u}) = r$, then we have a regular matrix and the solution of (2.68) is unique. We obtain by (2.68)

$$W_\alpha \nu + I_{l,u}^T \eta = -J^T\mu \tag{2.69a}$$

$$I_{l,u}\nu = 0, \tag{2.69b}$$

and by (2.69b), we notice that $\nu_i = 0$ for $i \in I_{l,u}$. So, it is enough to compute the previous "j-th" components of ν, such that $j \notin I_{l,u}$. In addition, we know that $(I_{l,u}^T\nu)_j = 0$ for $j \notin I_{l,u}$, then

$$W_\alpha^{I_c}\nu^{I_c} = -J_{I_c}^T\mu, \quad I_c := \{1, \ldots, n\}\backslash I_{l,u}, \tag{2.70}$$

where $W_\alpha^{I_c}$ comes to remove $i-th$ row and column of W_α $\forall\, i \in I_{l,u}$. Analogously, ν^{I_c} is obtained from removing the i-th element of ν and J_{I_c} comes from removing the i-th column of J.

Corrector. The Newton direction is computed via solving

$$\begin{array}{ccc} \displaystyle\min_{(\nu,\delta)\in\mathbb{R}^n\times\mathbb{R}} & \delta & \\ \text{s.t} & \nabla f_i(x)^T\nu + \frac{1}{2}\nu^T\nabla^2 f_i(x)\nu \leq \delta, & i = 1, ..., k. \\ & -\nu_i - x_i + l_i \leq 0, & i \in I_l. \\ & \nu_i + x_i - u_i \leq 0, & i \in I_u. \end{array} \tag{2.71}$$

We assume that x_i is not active respect to both its upper and lower bound at the same time. For the step size control, we use again a component-wise Armijo condition, but in this case, we impose the following upper limit:

$$t_{max} = \min_{i=1,\ldots,n} t_i, \tag{2.72}$$

where

$$t_i = \begin{cases} \frac{l_i - x_i}{\nu_i} & \nu_i < 0 \\ \frac{u_i - x_i}{\nu_i} & \nu_i > 0, \quad i = 1, \ldots, n. \\ +\infty & \nu_i = 0. \end{cases} \tag{2.73}$$

2.6.2 Pareto Explorer

The main limitation of the PT for its application to MaOPs is that it is focused on the computation of the entire Pareto set/front of a given problem. This, however, gets intractable for an increasing number of objectives as discussed above. Moreover, even if we can compute such a huge amount of data, the problem arises of how it would be presented to the DM in a useful way. This is where the Pareto Explorer (PE) finds its niche, as this approach is mainly focused on the computation of a single trajectory along the landscape defined by the Pareto set/front. PE can hence be seen as a compilation of methods to follow the path given by the DM's needs, ultimately leading him/her to discover what he/she is looking for in a timely manner. The PE is thought to evolve as an interactive tool, where it can receive feedback from the user at any stage of the exploration. The PT, on the other hand, is designed to receive a fixed initial set of settings and run until completion, i.e., ideally until an approximation of the entire solution set is generated. Both techniques PT and PE complement each other and shall optimize their implementations based on their respective scopes.

The PE method consists of two main stages: first an initial solution x_0, which is ideally Pareto optimal or at least a KKT point, is computed or selected out of a set of possible solutions for the given problem; and secondly, the Pareto landscape is explored around x_0, where a steering is performed according to the DMs' preferences. Algorithm 1 shows the general framework of the PE. In general, the first part of the PE can be accomplished by any existing solver for M(a)OPs, including MP techniques such as scalarization methods that already incorporate preference information, evolutionary reference point problems ([Deb and Sundar, 2006]), knee based solvers ([Rachmawati and Srinivasan, 2006]), or the execution of a MOEA (e.g.,[Deb and Jain, 2014, Zhang and Li, 2007, Beume et al., 2007]) followed by the selection of a starting point out of the final population. Here, it is advisable to choose a global method to avoid searching around a candidate solution that is only locally optimal. For the second stage, however, the PE will play its main role by restricting the search to a movement directed by the user's preferences. These preferences, as we will see

Algorithm 2.1 Framework of the Pareto Explorer

```
Require: multi-objective optimization problem of the form (2.12)
Ensure: sequence $\{x_0, x_1, \ldots\}$ of candidate solutions that perform a movement along the Pareto set/front into user specified directions.
compute/select an initial solution $x_0 \in \mathcal{M}$ of (2.12)
for $i = 0, 1, \ldots$ do
    compute a candidate solution $x_{i+1}$ in the vicinity of $x_i$ according to the decision makers' preferences
end for
```

later on, can be expressed in terms of directions in either decision or objective space, as well as in the space of the weight vectors.

In the following, we present several continuation methods that perform a local search from a given initial solution into user specified directions. All methods will make use of the steering feature of the PT as well as the explicit formulations of the tangent spaces.

We will assume that the initial solution x_0 is regular. All methods generate further regular solutions of the given MaOP (i.e., the movement is performed along the Pareto set/front). All steering procedures except the Unbiased Neighborhood Exploration are presented such that they perform the largest possible trajectory in this direction and are thus not in interaction with the DM. These adaptations, however, are straightforward.

Steering in Objective Space

It might be desired to steer the search into a user specified direction from a given solution x_0. The first proposed approach is to perform the steering in objective space, i.e., PE considers the case where the DM desires to change the objective values w.r.t. $F(x_0)$. That is, a direction $d_y \in \mathbb{R}^k$ is given with the aim to guide the search toward certain preferences that are only known in objective space. However, as the Pareto front is not known, it is of course unclear whether a movement in d_y can be actually performed. Then, since the linearized front at $F(x_0)$ can be computed, and since the underlying idea is to steer the search along the set of optimal solutions, it makes sense to perform a 'best fit' movement as follows (compare to Fig. 2.7a): (i) first, d_y is projected onto $T_yF(\mathcal{M})$ to obtain a best fit direction $d_y^{(i)}$ in the i^{th} step of the algorithm. Next, (ii) a movement is performed in direction $d_y^{(i)}$ using the steering properties of PT. Step (i) can be realized via a QR-factorization of α_0, and step (ii) via selecting the predictor direction ν_i that satisfies (2.40) for $d := d_y^{(i)}$, together with a corrector as used in PT. The process has to be stopped at iteration i when

$$\underbrace{\langle d_y^{(i)}, d_y \rangle \leq \epsilon_1}_{(a)} \text{ or } \underbrace{sign((d_y^{(i)})_j) = -sign((d_y^{(i-1)})_j) \; \forall j = 1, \ldots, k}_{(b)}, \text{ or } \underbrace{\|\alpha_i\|_\infty \geq 1 - \epsilon_2}_{(c)} \tag{2.74}$$

for some (small) thresholds $\epsilon_1, \epsilon_2 > 0$. In case (a) the corrected direction $d_y^{(i)}$ is nearly orthogonal to d_y, and thus, no further improvement in this direction can be expected (see Fig 2.7b). This point, however, can only be reached for certain step sizes. If the steps are too large, an oscillating behavior can be observed (case (b)). Finally, if a corner of the Pareto set is reached (case (c)), no further improvement in the d_y-direction can be performed as the search is restricted to Pareto-optimal solutions. The pseudo code of the steering in objective space is shown in Algorithm 2.2.

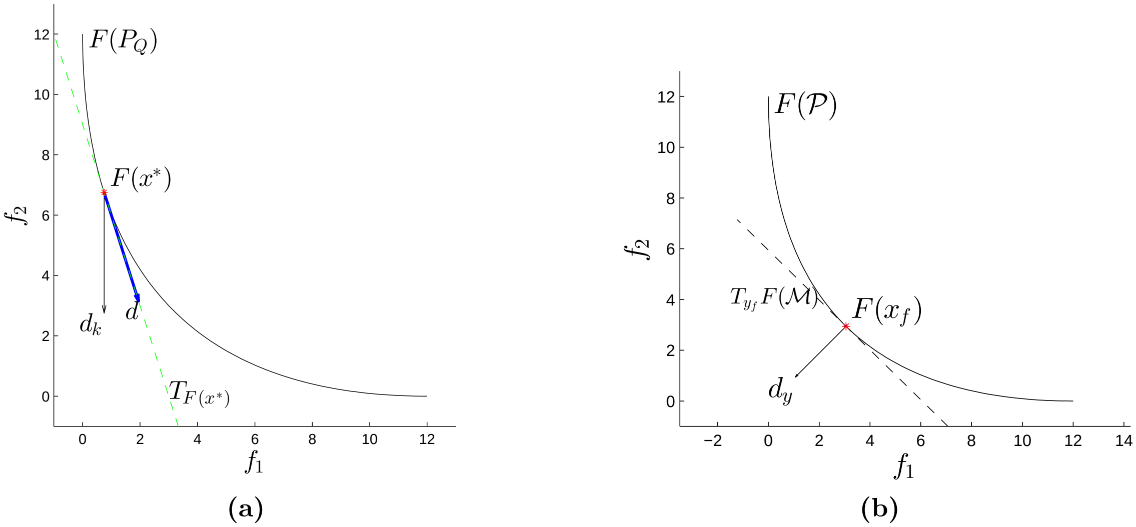


Figure 2.7: (a) $d_y^{(i)}$ is the orthogonal projection of d_y onto the linearized Pareto front at $F(x_i)$ and hence the best fit direction for a movement along the Pareto front. (b) the process has to be stopped at x_f if $T_{y_f}F(\mathcal{M})$ and d_y are orthogonal to each other, that is, $d_y^{(f)} = 0$.

Steering in Decision Space

Similar to the steering in objective space, PE can perform a best fit movement along the Pareto set for a given preference direction in decision space for which we will present two variants.

The first variant is analog to the best fit movement in objective space as presented above. That is, for a given point $x_i \in \mathcal{M}$ and a given a direction $d_x \in \mathbb{R}^n$ in *decision space*, one can project d_x onto the linearized Pareto set at x_i. This vector ν_i can now be used as predictor in a PC step of the PT (compare to Fig. 2.8 (a)). Algorithm 2.3

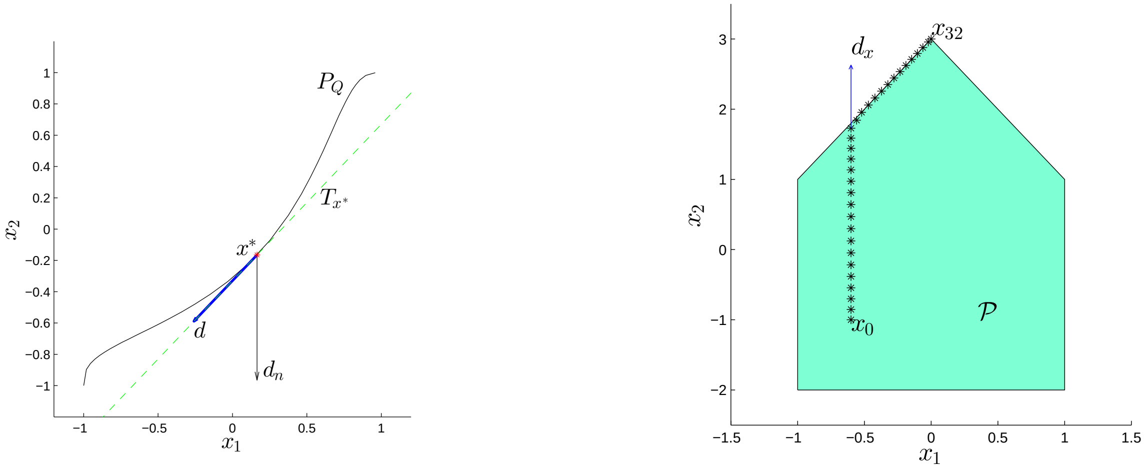
Algorithm 2.2 Steering in objective space

```
Require: $x_0 \in \mathcal{M}$ with associated weight $\alpha_0$, direction $d_y \in \mathbb{R}^k$ in objective space,
    distance $\tau > 0$, tolerances $\epsilon_1, \epsilon_2 > 0$, maximal number $MaxIter$ of iterations
Ensure: set $\{x_1, \dots, x_i\}$ of candidate solutions around $x_0$ whose images $F(x_i)$ are a
    best fit movement in $d_y$-direction along the Pareto front
for $i = 0, 1, \dots, MaxIter$ do
    if $\|\alpha_i\|_\infty \geq 1 - \epsilon_2$ then
        return $\{x_1, \dots, x_i\}$                    ▷ corner of Pareto set reached
    end if
    compute $\alpha_i = Q_i R_i = (q_1^{(i)}, q_2^{(i)}, \dots, q_k^{(i)}) R_i$
    set $B_i := (q_2^{(i)}, \dots, q_k^{(i)})$
    set $d_y^{(i)} := B_i B_i^T d_y$
    if $|\langle d_y^{(i)}, d_y \rangle| \leq \epsilon_1$ then
        return $\{x_1, \dots, x_i\}$  ▷ no movement in the $d_y$-direction can be performed
    end if
    if $sign((d_y^{(i)})j) = -sign((d_y^{(i-1)})_j),\ \forall j = 1, \dots, k$ then
        return $\{x_1, \dots, x_i\}$                    ▷ oscillation of candidate solutions
    end if
    set $t_i := \tau / \|J\nu_i\|$
    set $\nu_i = -W_\alpha^{-1} \mu_{d_i}$ where $\mu_{d_i}$ is computed as in (2.46)
    set $\tilde{x}_{i+1} = x_i + t_i \nu_i$
    compute solution $x_{i+1}$ of (2.12) near $\tilde{x}_{i+1}$ using a corrector step
end for
return $\{x_1, \dots, x_{MaxIter}\}$
```

shows the pseudo code of this steering in decision space. Hereby, the columns of X_i form an ONB of $T_{x_i}\mathcal{M}$. The search has to be stopped when

$$\underbrace{\langle d_x, \nu_i \rangle \leq \epsilon_1}_{(a)} \text{ or } \underbrace{\langle x_i - x_0, d_x \rangle \leq \langle x_{i-1} - x_0, dx \rangle}_{(b)}, \text{ or } \underbrace{\|\alpha_i\|_\infty \geq 1 - \epsilon_2}_{(c)}. \tag{2.75}$$

In case (a), the search direction ν_i is nearly orthogonal to d_x and thus, no further improvement in this direction can be expected. As for the steering in objective space, this can only be achieved for sufficiently small step sizes. Instead, oscillations around a Pareto point whose tangent space is orthogonal to d_x can occur (case (b)). Finally, the search has to be stopped if a corner of the Pareto set is reached (case (c)).



(a) **(b)**

Figure 2.8: (a): best fit direction in decision space. (b): example of the steering in decision space for a MOP with $n = 2$ and $k = 5$. For the final iteration, x_{32} makes $\|\alpha\|_\infty = 1$.

Next, we present a modification of the above movement. For this, assume that the DM has the aim to change the solution in decision space while it is desired to keep the change in objective space as low as possible. This could be the case when there is a change in the supply (e.g., one component has become more expensive) while the product is already established in a certain desired market niche. For this scenario, one possibility is to change the original direction d_x via a best fit direction ν^* for which the change in objective space is zero (for infinitesimal step sizes) leading to

$$\begin{aligned} \min_{\nu} \quad & -d_x^T \nu \\ \text{s.t.} \quad & J\nu = 0 \\ & \|\nu\|_2^2 = 1, \end{aligned} \tag{2.76}$$

Algorithm 2.3 Steering in decision space

```
Require: $x_0 \in \mathcal{M}$ with associated weight $\alpha_0$, direction $d_x \in \mathbb{R}^n$ in decision space,
    distance $\tau > 0$, tolerances $\epsilon_1, \epsilon_2 > 0$, maximal number $MaxIter$ of iterations
Ensure: set $\{x_1, \dots, x_i\}$ of candidate solutions around $x_0$ that perform a best fit
    movement in $d_x$-direction along the Pareto front
for $i = 0, 1, \dots, MaxIter$ do
    if $\|\alpha_i\|_\infty \geq 1 - \epsilon_2$ then
        return $\{x_1, \dots, x_i\}$                    ▷ corner of Pareto set reached
    end if
    set $\nu_i := X_i X_i^T d_x / \|X_i X_i^T d_x\|$
    if $|\langle \nu_i, d_x \rangle| \leq \epsilon_1$ or $\langle x_i - x_0, d_x \rangle \leq \langle x_{i-1} - x_0, dx \rangle$ then
        return $\{x_1, \dots, x_i\}$         ▷ no movement in $d_x$-direction can be performed
    end if
    set $\tilde{x}_i = x_i + t_i \nu_i$
    compute solution $x_{i+1}$ of (2.12) near to $\tilde{x}_{i+1}$ using a corrector step
end for
return $\{x_1, \dots, x_{MaxIter}\}$
```

and proceed as above using the solution ν^* of (2.76) instead of d_x. If d_x is not orthogonal to the kernel of J at the current iterate x_i, ν^* can be computed via an orthogonal projection of d_x onto the kernel: if

$$J^T = QR = (Q_1|Q_2)R \tag{2.77}$$

is a QR-factorization of J^T, then the column vectors of $Q_2 \in \mathbb{R}^{n \times (n-k+1)}$ form an ONB of $ker(J)$. Thus, the projected vector is given by

$$\tilde{\nu}^* = Q_2 Q_2^T d_x, \tag{2.78}$$

and ν^* can be computed via a normalization of $\tilde{\nu}^*$. If $\tilde{\nu}^* = 0$, this means that d_x is orthogonal to $ker(J)$, and the search can be stopped. To compute the predictor for the resulting PC method, ν^* is then projected onto $T_{x_i}\mathcal{M}$, i.e.,

$$\nu_i := \frac{X_i X_i^T \nu^*}{\|X_i X_i^T \nu^*\|} = \frac{X_i X_i^T Q_2 Q_2^T d_x}{\|X_i X_i^T Q_2 Q_2^T d_x\|}. \tag{2.79}$$

The rest of the resulting PC method is identical to its base variant. One possible problem is that the projected vector ν^* could already be orthogonal to $T_{x_i}\mathcal{M}$ leading to a stop of the algorithm. The following short discussion, however, shows that ν^* is always W_α-orthogonal to $T_{x_i}\mathcal{M}$ which means that orthogonality is only given in a few cases. To see the last statement, let $t_i = -W_\alpha^{-1} J^T \mu_i$ be a tangent vector. Then

$$t_i^T W_\alpha \nu^* = -(W_\alpha^{-1} J^T \mu_i)^T W_\alpha \nu^* = -\mu_i^T J W_\alpha^{-1} W_\alpha \nu^* = -\mu_i^T J \nu^* = 0. \tag{2.80}$$

Algorithm 2.4 shows the pseudo code of the modified steering in decision space, where the secondary objective is to minimize the change in objective space.

Algorithm 2.4 Modified steering in decision space

```
Require: $x_0 \in \mathcal{M}$ with associated weight $\alpha_0$, direction $d_x \in \mathbb{R}^n$ in decision space,
    distance $\tau > 0$, tolerances $\epsilon_1, \epsilon_2 > 0$, maximal number $MaxIter$ of iterations
Ensure: set $\{x_1, \dots, x_i\}$ of candidate solutions around $x_0$ that perform a best fit
    movement in the $d_x$-direction along the Pareto front while minimizing the change
    in objective space
for $i = 0, 1, \dots, MaxIter$ do
    if $\|\alpha_i\|_\infty \geq 1 - \epsilon_2$ then
        return $\{x_1, \dots, x_i\}$                    ▷ corner of Pareto set reached
    end if
    compute $\nu_i$ as in (2.79)
    if $|\langle \nu_i, d_x \rangle| \leq \epsilon_1$ or $\langle x_i - x_0, d_x \rangle \leq \langle x_{i-1} - x_0, dx \rangle$ then
        return $\{x_1, \dots, x_i\}$        ▷ no movement in $d_x$-direction can be performed
    end if
    set $\tilde{x}_i = x_i + t_i \nu_i$
    compute solution $x_{i+1}$ of (2.12) near to $\tilde{x}_{i+1}$ using a corrector step
end for
return $\{x_1, \dots, x_{MaxIter}\}$
```

Steering in Weight Space

The last movement we present here is the steering in weight space where the DM might be interested to gradually change the importance of the objectives. The key for this steering is the observation that a vector $\mu \in \mathbb{R}^k$ with $\sum_{i=0}^k \mu_i = 0$ can be seen as a change in weight space. Another way to see this is that for every two convex weights $\alpha^{(1)}, \alpha^{(2)} \in \mathbb{R}^k$ the difference vector $\Delta\alpha := \alpha^{(1)} - \alpha^{(2)}$ is of the above form. For $k = 2$ objectives, there are only two choices for μ after normalization: $\mu = (1, -1)^T$ and $\mu = (-1, 1)^T$. The first one means that the importance of f_1 should be increased and thus its values be decreased for the sacrifice of f_2. Hence, $\mu = (1, -1)^T$ should result in a movement left up the Pareto front from $F(x_0)$ while $\mu = (-1, 1)^T$ should result in a movement right down. Regrettably, the resulting movement depends on the shape of the Pareto front as it makes a difference if the front is convex or concave. In order to steer the movement into the desired direction, we need to apply a small trick: we have to indicate for a selected objective f_i if its value has to increase or decrease during the search. More precisely, we choose a value $c_i \in \{-1, 1\}$ and compute the potential predictor direction $\nu_\mu = -W_\alpha^{-1} J^T \mu$ as described above. Then, we choose the predictor direction ν_i in the ith step as

$$\nu_i := \begin{cases} \nu_\mu & \text{if } sign((J\tilde{\nu})_i) = sign(c_i) \\ -\nu_\mu & \text{else} \end{cases} . \tag{2.81}$$

The search has to be stopped when

$$\underbrace{\alpha_i = 0,\ i \in \{1, \ldots, k\}}_{(a)} \text{ or } \underbrace{sign(J(x_i)\nu_i)_j) = -sign(J(x_{i-1})\nu_{i-1})_j),\ j \in \{1, \ldots, k\}}_{(b)}. \tag{2.82}$$

Case (a) is given near the boundary of the Pareto set, and no further change in α-space according to μ can be performed. Case (b) indicates that there has been a change in the curvature of the Pareto front (e.g., from concave to convex) which means that the value of μ has a different meaning. The pseudo code of this steering can be found in Algorithm 2.5.

Algorithm 2.5 Steering in weight space

```
Require: $x_0 \in \mathcal{M}$ with associated weight $\alpha_0$, direction $\mu \in \mathbb{R}^k$ in weight space,
    preference $c_i \in \{-1, 1\}$ of change in the $i$th objective function distance $\tau > 0$,
    tolerance $\epsilon 0$, maximal number $MaxIter$ of iterations
Ensure: set $\{x_1, \ldots, x_i\}$ of candidate solutions around $x_0$ that perform a best fit
    movement along the Pareto set w.r.t. $\mu$.
for $i = 0, 1, \ldots, MaxIter$ do
    if $\alpha_i \leq \epsilon$ for a $i \in \{1, \ldots, k\}$ then
        return $\{x_1, \ldots, x_i\}$                ▷ boundary of Pareto set reached
    end if
    compute $\nu_\mu$ as in (2.40)
    if $sign((J\tilde{\nu})_i) = sign(c_i)$ then
        $\nu_i := \nu_\mu$
    else
        $\nu_i := -\nu_\mu$
    end if
    if $sign(J(x_i)\nu_i)_j) = -sign(J(x_{i-1})\nu_{i-1})_j)$ for a $j \in \{1, \ldots, k\}$ then
        return $\{x_1, \ldots, x_i\}$                ▷ change in curvature of the Pareto front
    end if
    set $t_i := \tau / \|J\nu_i\|$
    set $\tilde{x}_{i+1} = x_i + t_i\nu_i$
    compute solution $x_{i+1}$ of (2.12) near to $\tilde{x}_{i+1}$ using a corrector step
end for
return $\{x_1, \ldots, x_{MaxIter}\}$
```

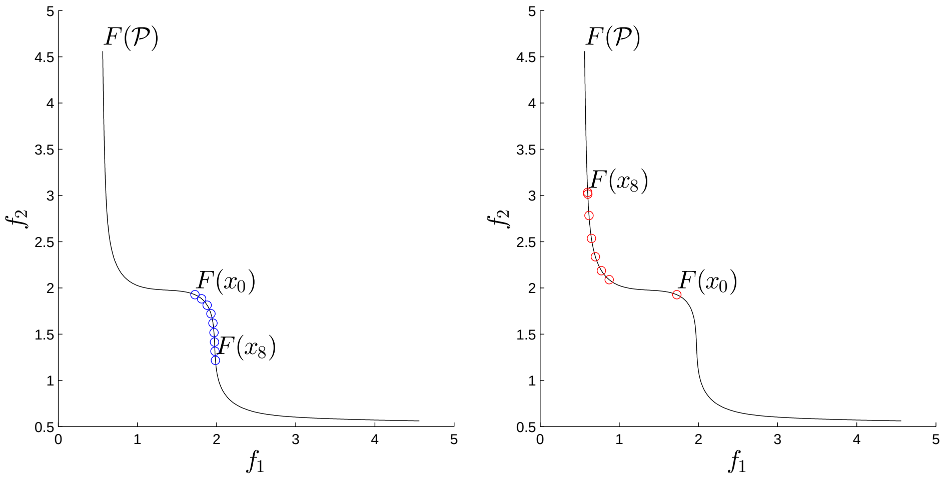
This method is at first highly related to the weighted sum method (WS, [Gass and Saaty, 1955]), which is probably the most widely used scalarization method, where the SOP

$$\min_{x \in Q} \sum_{i=1}^{k} \alpha_i f_i(x) \tag{2.83}$$

is obtained by introducing a convex weight $\alpha \in \mathbb{R}^k$. The above mentioned steering could e.g. be realized via solving a sequence of SOPs of the form (2.83) using the weights

$$\alpha_i = \alpha_0 + t_i \mu, \tag{2.84}$$

where α_0 is the weight vector of the initial solution x_0, t_i a chosen step size, and μ the desired change in weight space. It is, however, well known ([Das and Dennis, 1997]) that the choice of the weight vector α in (2.83) is a delicate problem. For instance, small changes in α do not have to lead to small changes in the solution of (2.83), while such changes can be controlled in Algorithm 2.5 via the choice of τ. Next, as discussed above, the resulting movement actually depends on the curvature of the Pareto front, and cannot be adjusted as in (2.81).



(a) Steering in weight space

(b) Weighted sum method

Figure 2.9: Example where a movement towards the lower right of the Pareto front desired.

Chapter 3

Extensions of the Pareto Explorer for Continuous MaOPs

As we explained before, the *Pareto Explorer* (PE) consists of two principal phases. The first one is about how to obtain a global optimal solution for a given MaOP, which will be addressed in the next chapter. The second phase –explained in detail in Section 2.5– is the local exploration of the optimal solutions, both in objective and decision spaces. In this chapter, we describe two more ways to steer the search using the PE for continuous MaOPs, and we present a real-world application which was solved using the continuous PE.

For this, in Section 3.1, we consider the scenario where the DM is almost satisfied with a particular solution x_0, then he/she wants to explore the nearest optimal solutions around x^0, i.e., an unbiased neighborhood exploration. In Section 3.2, we present the second way of steering, which employs the PE to find points in the *"knee"* of the PF. The knee of the PF is a region of interest for many applications, that will be mathematically defined in this section. Finally, in Section 3.3, we explain how to use the PE framework to solve the problem of the Plastic Injection Molding (PIM) with seven objectives.

3.1 Unbiased Neighborhood Exploration

The first exploration tool is the Unbiased Neighborhood Exploration (UNE, see Algorithm 3.1). The aim of this tool is to provide the DM with a set of well-distributed solutions in the vicinity of the current solution. This way, an unbiased overview of the Pareto landscape is generated around x_0 (respectively around $F(x_0)$) with a budget of N further solutions that have to be computed. As no bias is given or even wanted, it makes sense to evenly distribute the search directions in which the new solutions x_i

are computed. Therefore, PE has been designed to make use of an existing specialized algorithm for the computation of evenly distributed points on the unit hypersphere. The key for such an even distribution is (2.40) that allows to explicitly compute the respective direction in tangent space. In particular, one can obtain such even distributions as follows:

A compute a set $p_1, \ldots, p_N \in \mathbb{R}^m$ of evenly distributed points on the unit sphere $\Omega_m := \{x \in \mathbb{R}^m \ : \ x^T x = 1\}$ via the *minimizing potential energy problem* ([Nie and Ellingwood, 2004])

$$\min \Pi_2(p_1, \ldots, p_N) = \sum_{1 \leq j \leq i \leq N} \frac{1}{\|p_i - p_j\|_2^2}. \tag{3.1}$$

B Let $B \in \mathbb{R}^{m \times (k-1)}$ be a matrix whose column vectors b_l build an orthonormal basis (ONB) of the solution set (i.e., either of $T_x\mathcal{M}$—in that case we have $m = n$ or $T_yF(\mathcal{M})$ $(m = k)$). Then, the evenly distributed predictor directions are given by

$$\nu_i := Bp_i = \sum_{l=1}^{k-1} p_{i,l} b_l, \tag{3.2}$$

where $p_{i,l}$ denotes the l-th entry of p_i.

Algorithm 3.2 shows one example of how to obtain evenly spread solutions in *objective* space such that the difference vectors $F(x_i) - F(x_0)$ approximately have the same lengths.

Algorithm 3.1 Framework of UNE

```
Require: $x_0 \in \mathcal{M}$, number $N$ of neighborhood solutions
Ensure: set $\{x_1, \ldots, x_N\}$ of candidate solutions around $x_0$.
  compute a set $\nu_1, \ldots, \nu_N \in \mathbb{R}^n$ of well-distributed directions
  compute step sizes $t_i$, $i = 1, \ldots, N$
  compute predictors $\tilde{x}_i := x_0 + t_i \nu_i$, $i = 1, \ldots, N$
  compute solutions $x_i$, $i = 1, \ldots, N$, of (2.12) near to $\tilde{x}_i$ using a corrector step
  return $X := \{x_1, \ldots, x_N\}$
```

In order to obtain such evenly spread solutions in *decision space*, one has to modify Algorithm 3.2 as follows: (i) the columns of B form an ONB of $T_x\mathcal{M}$, (ii) the predictor directions are set as $\nu_i := Bp_i$, and (iii) the step size is $t_i := \tau$ for all directions.

Figure 3.1 shows some numerical results of the UNE on the benchmark function DTLZ2. As it can be seen, evenly spread solutions can be obtained either in decision or objective space for different values of N.

Algorithm 3.2 UNE: Evenly distributed solutions in objective space

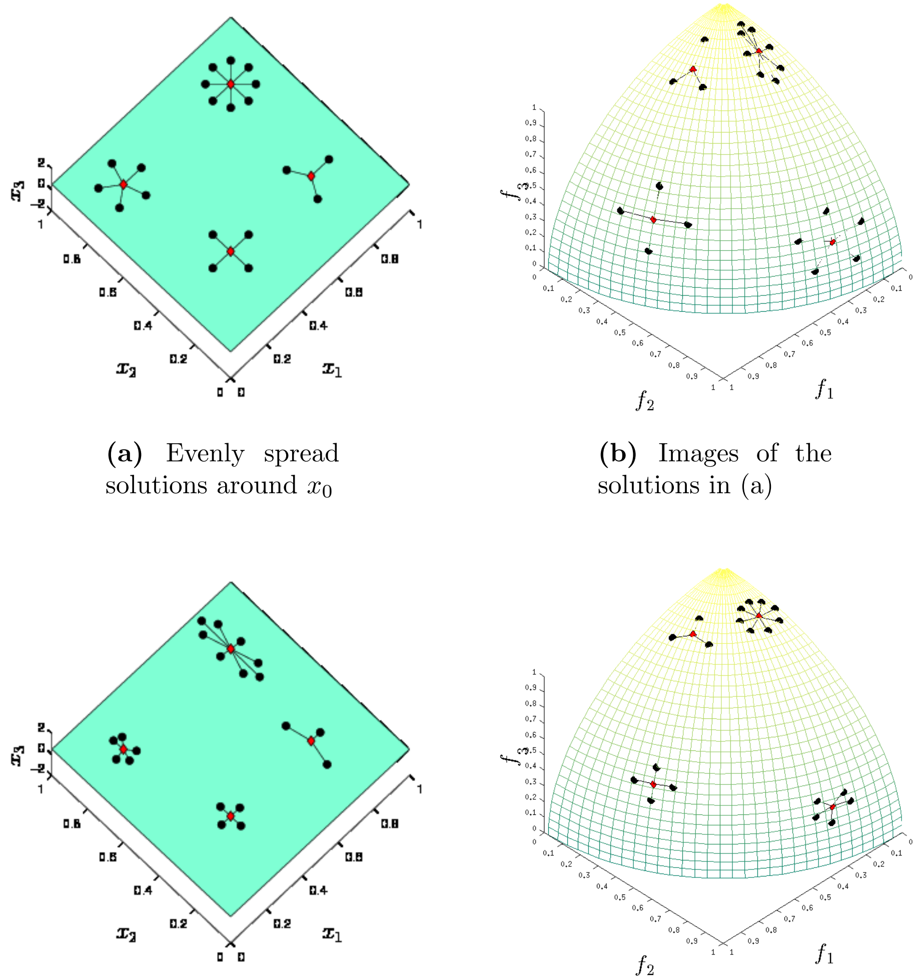
```
Require: $x_0 \in \mathcal{M}$ with associated weight $\alpha_0$, number $N$ of neighborhood solutions,
    distance $\tau > 0$
Ensure: set $\{x_1, \dots, x_N\}$ of candidate solutions around $x_0$ whose images $F(x_i)$ are
    evenly spread around $F(x_0)$ with $\|F(x_i) - F(x_0)\| \approx \tau$
  compute $p_i$, $i = 1, \dots, N$, as described in $A$
  compute $\alpha_0 = QR = (q_1, q_2, \dots, q_k)R$
  set $B := (q_2, \dots, q_N)$
  set $d_i := Bp_i$, $i = 1, \dots, N$
  set $\nu_i := -W_\alpha^{-1}\mu_{d_i}$, $i = 1, \dots, N$, where the $\mu_{d_i}$'s are computed as in (2.46)
  set $t_i := \tau/\|J\nu_i\|$, $i = 1, \dots, N$
  set $\tilde{x}_i := x_0 + t_i\nu_i$, $i = 1, \dots, N$
  compute solutions $x_i$, $i = 1, \dots, N$, of (2.12) near to $\tilde{x}_i$ via a corrector step
  return $X := \{x_1, \dots, x_N\}$
```

We also tested the DTLZ2 problem with rNSGA-II [Deb and Sundar, 2006] trying to reproduce the UNE results in objective space. We consider similar conditions for a fair comparison, that is, we use the initial points of the UNE example as the reference points for the rNSGA-II (they were also placed within the initial population) and $\epsilon = \tau$ in order to obtain a similar spread. Different runs of rNSGA-II show some important issues of this method in contrast to our approach. We obtain a different distribution of the final population, that is, we do not have control about the final number of solutions near to each reference point or about the spread between them.

3.2 Pareto Explorer for Finding the Knee

A lot of applications require finding the solution with an adequate trade-off between all the objectives. The knee of the PF usually provides such a point. In [Das, 1999], we can find a mathematical definition of the knee and a procedure, based on the NBI algorithm, for reaching it.

In this section, we use the Pareto Explorer framework [Schütze et al., 2019] for finding the knee of MaOPs. We also prove the equivalence of the knee definition of [Das, 1999] with the proposed approach. Finally, we demonstrate the advantages of our proposal on several examples.



(a) Evenly spread solutions around x_0

(b) Images of the solutions in (a)

(c) Pre-images of the solutions in (d)

(d) Evenly spread solutions around $F(x_0)$

Figure 3.1: Examples of the unbiased neighborhood exploration on DTLZ2 with $n = 3$, $k = 3$, and different values of N, $N = 3, 4, 5, 8$. On the top we, can see the exploration in decision space; and, on the bottom, we can observe the exploration in objective space. Starting points are the same for both cases in order to show the variations.

3.2.1 Definition of the Knee

The definition of the knee of Das [Das, 1999] is stated as:

$$
\begin{aligned}
& \max_{(\alpha,t,\beta)} \quad t \\
\text{s.t.} & \\
& \begin{array}{rcl}
\Phi\beta + t\hat{n} & = & F(x) - F^* \\
e^T\beta & = & 1 \\
h(x) & = & 0 \\
g(x) & \leq & 0 \\
\beta_i & \geq & 0, \quad i = 1, 2, \cdots, k,
\end{array}
\end{aligned}
\tag{3.3}
$$

where Φ is the matrix of the Convex Hull of Individual Minima (CHIM) and $\hat{n}$ is the normalized normal of the CHIM.

Notice that we can write (3.3) as:

$$
\min_{(\alpha,\beta,t)} \; -t \tag{3.4a}
$$

$$
\begin{aligned}
\text{s.t.} & \qquad \qquad \qquad \qquad \qquad \qquad \qquad & \text{(3.4b)} \\
e^T\beta - 1 &= 0 & \text{(3.4c)} \\
F(x) - \Phi\beta - t\hat{n} - F^* &= 0 & \text{(3.4d)} \\
h(x) &= 0 & \text{(3.4e)} \\
g(x) &\leq 0 & \text{(3.4f)} \\
-\beta_i &\leq 0, \quad i = 1, 2, \cdots, k. & \text{(3.4g)}
\end{aligned}
$$

3.2.2 Finding the Knee

In order to see the equivalence of the solution provided by the PE and the Das method, we present one important theoretical result.

Theorem 3.1. *Let x^* be a KKT point of* (2.12) *such that $\hat{\alpha}^T\hat{n} = -1$ (that is, $\alpha = t\hat{n},\ t \in \mathbb{R}\setminus\{0\}$). If $F(x^*)$ can be obtained via the NBI method, then there exists t^* and β^* such that the vector (x^*, β^*, t^*) is also a KKT point of the problem* (3.4)*.*

Proof. First we write the KKT conditions for the problem (3.4) associated to (2.12) as follows.

There exist Lagrange multipliers $\bar{\lambda} = (\bar{\nu}^T, \bar{\alpha}^T, \tilde{\lambda}^T)^T \in \mathbb{R}^{k+k+p}$ and $\bar{\gamma} = (\tilde{\gamma}^T, \bar{\zeta}^T)^T \in \mathbb{R}^{m+k}$, where $\bar{\nu} \in \mathbb{R}^k$, $\bar{\alpha} \in \mathbb{R}^k$, $\tilde{\lambda} \in \mathbb{R}^p$, $\tilde{\gamma} \in \mathbb{R}^m$, and $\bar{\zeta} \in \mathbb{R}^k$, such that the following conditions are satisfied:

$$\begin{aligned}
&\underbrace{\begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}}_{\nabla(3.4\text{a})} + \underbrace{\begin{pmatrix} 0 \\ e \\ 0 \end{pmatrix}}_{\nabla(3.4\text{c})} \bar{\nu} + \underbrace{\begin{pmatrix} J(x^*) \\ -\Phi \\ -\hat{n} \end{pmatrix}}_{\nabla(3.4\text{d})} \bar{\alpha} + \sum_{i=1}^{p} \tilde{\lambda}_i \underbrace{\begin{pmatrix} \nabla h_i(x^*) \\ 0 \\ 0 \end{pmatrix}}_{\nabla(3.4\text{e})} + \\
&\sum_{i=1}^{m} \tilde{\gamma}_i \underbrace{\begin{pmatrix} \nabla g_i(x^*) \\ 0 \\ 0 \end{pmatrix}}_{\nabla(3.4\text{f})} + \underbrace{\begin{pmatrix} 0 \\ -e \\ 0 \end{pmatrix}}_{\nabla(3.4\text{g})} \bar{\zeta} = 0
\end{aligned} \tag{3.5a}$$

$$e^T\beta - 1 = 0 \tag{3.5b}$$

$$F(x) - \Phi\beta - t\hat{n} - F^* = 0 \tag{3.5c}$$

$$h_i(x^*, \beta^*, t^*) = h_i(x^*) = 0, \quad i = 1 \dots p, \tag{3.5d}$$

$$g_i(x^*, \beta^*, t^*) = g_i(x^*) \leq 0, \quad i = 1 \dots m, \tag{3.5e}$$

$$-\beta_i \leq 0, \quad i = 1, 2, \cdots, k, \tag{3.5f}$$

$$\hat{\gamma}_i \geq 0, \quad i = 1 \dots m, \tag{3.5g}$$

$$\bar{\zeta}_i \geq 0, \quad i = 1 \dots k, \tag{3.5h}$$

$$\hat{\gamma}_i g_i(x^*, \beta^*, t^*) = \hat{\gamma}_i g_i(x^*) = 0, \quad i = 1 \dots m, \tag{3.5i}$$

$$-\bar{\zeta}_i \beta_i = 0, \quad i = 1 \dots k. \tag{3.5j}$$

By our assumptions, we have that $\exists\, t^*$ and β^* such that (3.5b), (3.5c), and (3.5f) are satisfied. Also notice that conditions (2.13b) and (2.13c) are equivalent to (3.5d) and (3.5e), respectively.

Notice that (3.5a) does not depend of β and t, then we only have to show that any point x^* (a KKT point of (2.12)) also satisfies it. As x^* satisfies (2.13a), then we have that:

$$\sum_{i=1}^{k} \alpha_i \nabla f_i(x^*) + \sum_{i=1}^{p} \lambda_i \nabla h_i(x^*) + \sum_{i=1}^{m} \gamma_i g_i(x^*) = 0,$$

and then, for $c \in \mathbb{R}$ it follows that:

$$\begin{aligned}
&c\left(\sum_{i=1}^{k} \alpha_i \nabla f_i(x^*) + \sum_{i=1}^{p} \lambda_i \nabla h_i(x^*) + \sum_{i=1}^{m} \gamma_i g_i(x^*)\right) = c \cdot 0 = 0 \\
&\Rightarrow \sum_{i=1}^{k} c \cdot \alpha_i \nabla f_i(x^*) + \sum_{i=1}^{p} c \cdot \lambda_i \nabla h_i(x^*) + \sum_{i=1}^{m} c \cdot \gamma_i g_i(x^*) = 0
\end{aligned} \tag{3.6}$$

Now, we set $c = \dfrac{1}{\|\alpha\|}$, $\bar{\alpha} = c \cdot \alpha$ (that is, $\bar{\alpha}$ is a unit vector $\bar{\alpha} = \hat{\alpha}$), $\tilde{\gamma} = c \cdot \gamma$, and $\tilde{\lambda} = c \cdot \lambda$.

Finally, we have to verify the three equations for the condition (3.5a). Due to (3.6), the first equation of (3.5a) is satisfied. The last equation is satisfied by our assumption $\hat{\alpha}^T \hat{n} = -1$; while, we can set $\bar{\nu} = \Phi \bar{\alpha}$ for the second equation. On the other hand, if we set $\hat{\zeta} = 0$ (the zero vector), $\hat{\gamma} = \gamma$, and as $c > 0$, then we can easily verify the equivalence of the conditions (2.13f) and (2.13g), with (3.5g) and (3.5i), respectively. Notice this also makes (3.5h) and (3.5j) true.

In this way, all the conditions of (3.5) (i.e., (3.5a)-(3.5j)) are satisfied by any x^* with an associated vector α such that $\hat{\alpha}^T \hat{n} = -1$ □

Notice that Theorem 3.1 depends on the existence of t^* and β^* for a particular x^*, i.e., that the NBI problem can obtain such point x^*. As it is explained in [Das and Dennis, 1998], this is always possible for the bi-objective case. However, for $k \geq 3$, the NBI method is not able to find all the points in the boundary, but such overlooked points are likely lying near the periphery of the Pareto surface, and they are not relevant in the context of finding the knee of the PF. Even so, it is important to have a complete analysis of this case, which is part of our future work.

We can use PE to find x^*. Notice that, by the first stopping criterion of the steering in objective space, the PE will return the desired solution if we steer in objective space with the direction $\hat{n}$ (see Figure 3.2). That is, Theorem 3.1 provides us a way to find the KKT solution of (3.3) using the PE. Notice that the opposite is not always possible, i.e., a solution to a KKT point of (3.3) could not correspond to a KKT point with $\hat{\alpha}^T \hat{n} = -1$ (see Figure 3.3).

3.3 Real World Application: Plastic Injection Molding

3.3.1 The Model

In the following, we describe the many-objective PIM model consisting of seven objectives that we have used for this study. Further on, we present a plastic gear that we use as demonstrator for the application of the PE.

Design parameters

The process parameters we are considering here are the melt temperature (T_{melt}), the packing time (t_{pack}), the packing pressure (P_{pack}) and the cooling time (t_{cool}), which are briefly defined as follows:

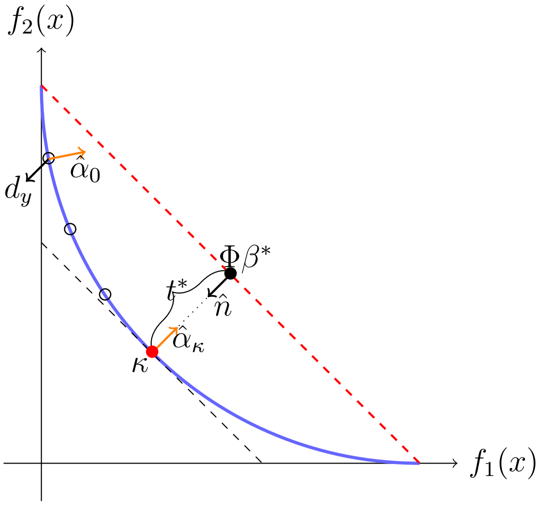


Figure 3.2: Illustrative example of the use of PE to find the knee. Here we start the steering in objective space at the point x_0 (with a corresponding weighted vector $\hat{\alpha}_0$) into direction $d_y := \hat{n}$. We compute a sequence of points until the stopping criterion (a) is satisfied, i.e., we have found the knee $\kappa = (f_1(x^*), f_2(x^*))^T$. Notice that $\hat{\alpha}_\kappa$ (associated to x^*) is in the opposite direction of $\hat{n}$, and hence $\hat{\alpha}^T \hat{n} = -1$.

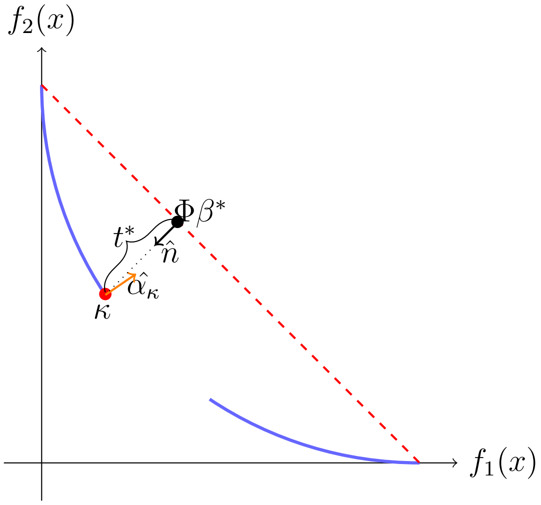


Figure 3.3: In this figure we can see an example where there does not exist a KKT point x^* of problem (2.12) such that $\hat{\alpha}^T \hat{n} = -1$, then for the solution of (3.4) we have that $\hat{\alpha}_\kappa^T \hat{n} \neq -1$. Notice that if we use the PE in this problem, it stops because the stopping criterion (b) at the same point κ, however, this is not the case in general.

- x_1: *Melt temperature* (T_{melt}): the temperature of the plastic melt as it enters the mold.

- x_2: *Packing time* (t_{pack}): the period of time where additional plastic is injected into the cavity to compensate for inherent shrinkage during the injection phase.

- x_3: *Packing pressure* (P_{pack}): the pressure exerted on the melt entrance during the packing phase. In our case, we consider P_{pack} as the packing pressure applied over the effective packing time, t_1=0.5t_{pack}, in a packing pressure profile.

- x_4: *Cooling time* (t_{cool}): the period of time after the packing phase and before the mold opening and part ejection. It can represent up to 50% of the cycle time.

These process parameters are the most frequently used parameters considered in the literature (e.g., [Alvarado-Iniesta et al., 2019, Kitayama et al., 2017, Kitayama and Natsume, 2014] and references therein).

Objectives

The outcomes of interest (or objectives) we consider here are related to the quality and productivity of the PIM process. The quality is measured by means of cosmetic and functional characteristics, while productivity is measured by indicators such as processing time and energy usage. Cosmetic characteristics are measured by means of *warpage* in the product, *shrinkage* and *sink marks*. Commonly, these objectives are considered in other works [Alvarado-Iniesta et al., 2019, Kitayama et al., 2017, Kitayama and Natsume, 2014]. Functional properties are represented by residual stresses such as *Von Mises* and *shear* stresses [Alvarado-Iniesta et al., 2018, Bakhtiari et al., 2016]. Productivity is measured by the *cycle time* and *clamping force* usage [Kitayama et al., 2017]. Likewise, these outcomes are defined next:

- f_1: *Warpage* (mm): produced by non-uniform shrinkage in the plastic part. Besides, by temperature differences from one side of the mold to the other. This objective is mainly affected by packing time and cooling time.

- f_2: *Volumetric Shrinkage* (%): all plastic parts tend to shrink, however, it is desired to have a minimum and uniform shrinkage. Non-uniform volumetric shrinkage leads to warpage and distortion of molded parts. High values may lead to sink marks or voids. It shows the percentage of part volume as the part is cooled from high temperature and high pressure to room conditions. Positive values represent volume shrinkage, while negative values mean volume expansion. This objective is affected by melt temperature, packing pressure and cooling time.

- f_3: *Sink marks* (mm): Plastic parts could present sink marks in the finished look. Higher values of this mean high degree on sink. It is an index to evaluate the packing effect. If it is positive, it means that packing is not enough, leading to sink marks. If it is negative, it means overpacking. A proper packing keeps the indicator close to zero. This objective is affected by packing pressure and packing time.

- f_4: *Von Mises stress* (MPa): the Von-Mises thermal residual stress of the ejected part. Thermal induced residual stress is the stress status after the part is ejected and cooled down to room temperature. Non-uniform volumetric shrinkage will cause residual stress if it did not transform into warpage. Higher values of residual stress cause void defects. Von Misses stress is the scalar that represents the equivalent stress used for breakage test of the product, which is defined with the stress components for each axis. This objective is affected by melt temperature and cooling time.

- f_5: *Shear stress* (%): the source of the residual stress in molded parts. If the shear stress is not distributed evenly, it can cause some dimensional problems. Too high shear stress might tend to drastically deform molecular chains, even to break and then weaken the strength of the plastic part. This objective is mainly affected by melt temperature.

- f_6: *Cycle time* (seconds): the total time of a process run, this includes the filling time, mold opening time, packing time, and cooling time. This objective is affected by packing and cooling times which can represent up to 70% of the total cycle time.

- f_7: *Clamping force* (Ton): the maximum force of machine (clamping unit) to keep the mold closed against the cavity pressure during injection/packing phases. It can be considered as a great influence for energy saving. This objective is affected by packing pressure and packing time.

3.3.2 Case study: A plastic gear

As a case study, we will use in this work the design of a particular plastic gear. A finite element (FE) model containing 32,025 elements was developed for simulating the injection molding process in MOLDEX3D R15 2018 (www.moldex3d.com). The material used is a type of polypropylene (PP) supplied by A. Schulman whose trade name is POLYFLAM RPP 374ND CS1. The properties of the material can be seen in Appendix A.

Building the model

The main goal of a surrogate model is to be as accurate as possible via using as few samples as possible. One of the major steps on the process of constructing a surrogate model is the sample collection. In this work, we collected a total of 150 samples at selected values of $x \in D$, where

$$D := \left\{ x \in \mathbb{R}^4 \; : \; \begin{array}{c} 190 \leq x_1 \leq 230 \\ 3 \leq x_2 \leq 5 \\ 60 \leq x_3 \leq 100 \\ 8 \leq x_4 \leq 14 \end{array} \right\}. \tag{3.7}$$

to evaluate $y \in \mathbb{R}^7$ via D-optimal [Box and Draper, 1971] and Latin hypercube [Mckay et al., 2000] experimental designs. Hence, these samples are used to generate surrogate models of f_i, $i = 1, \ldots, 7$, of the outcomes of interest, which make them suitable for an optimization algorithm.

Generation of a surrogate model can be seen as a multi-dimensional non-linear optimization problem, which can be solved via least squares. Therefore, the problem can be formally defined as

$$\min_{\beta} \| f(x, \beta) - y \|_2^2 = \min_{\beta} \sum_i (f(x_i, \beta) - y_i)^2, \tag{3.8}$$

where x is the input sample vector, y is the output sample vector, β is the parameter vector, and f is the surrogate model. The problem presented in (3.8) can be solved using different methods [Lawson and Hanson, 1995]. Polynomial and artificial neural networks models [Abhishek et al., 2017, D'Addona et al., 2017] represent some of the most popular surrogate models in engineering. Table 3.1 shows the results of the surrogate models generated for each one of the outcomes of interest.

Table 3.1: Objective functions of the PIM model

Function	**Surrogate model**	R^2 **training**	R^2 **testing**
f_1	Quadratic	0.99	0.99
f_2	Quadratic	0.99	0.99
f_3	Shallow neural network	0.98	0.93
f_4	Quadratic	0.98	0.98
f_5	Shallow neural network	0.82	0.89
f_6	Linear	1.00	1.00
f_7	Quadratic	0.93	0.93

3.3.3 Numerical Results

In this section, we present some numerical results for hypothetical scenarios of the PIM design. For this, we will first consider selected sub-problems with two and three objectives. The consideration of these MOPs might be interesting in case the decision maker has a strong preference on just a few objectives. Further, the results show that all the objectives are indeed in conflict and that the entire problem consisting of seven objectives cannot be treated any more with traditional methods. In the second sub-section, we will show some numerical results of the Pareto Explorer on the seven-objective MaOP on four selected scenarios.

Multi-objective PIM design

First we look at the MOP that is defined by the two objectives f_2 (quadratic) and f_6 (linear). Figure 3.4 shows the numerical results of the PT and the evolutionary algorithm NSGA-III [Deb and Jain, 2014]. See Table 3.2 for the chosen parameter setting of both algorithms. NSGA-III and PT yield almost identical results and are capable of approximating the Pareto front perfectly. A huge difference, however, is in the computational effort needed to compute the results. NSGA-III is given a budget of 50,000 function evaluations. In contrast, PT required 190 function and another 190 Jacobian calls. If counting one Jacobian call by 4 function evaluations (as this can be done if the Jacobians were evaluated via automatic differentiation [Griewank and Corliss, 1992]), then the PT result would have been obtained via total of less than 1,000 function evaluations which is significantly less than for the evolutionary algorithm.

Table 3.2: Parameters of the NSGA-III and the PT for the PIM.

Algorithm	Parameter	Value
NSGA-III	Population size	92
	Reference points	91
	Crossover probability	1
	Mutation probability	$1/n$
	Distribution index for crossover	20
	Distribution index for mutation	20
PT	τ	0.01
	x_0	$(210.00, 4.00, 80.00, 11.00)^T$

Second, we consider the MOP that is defined by f_1 (quadratic) and f_5 (which is multi-modal and is defined by a neural network model). Figure 3.5 shows the results obtained by PT and NSGA-III. Again, PT spends much less function evaluations (around 4,000 counting one Jacobian call as 4 function calls as above) than NSGA-III (50,000). However, as PT was run with one single starting point, it only detects one part of the Pareto front that consists of 2 connected components. As it does not detect the 2nd component, it computes also some solutions that are only locally optimal. NSGA-III, on the other hand, is capable of detecting both components and delivers a suitable approximation of the solution set.

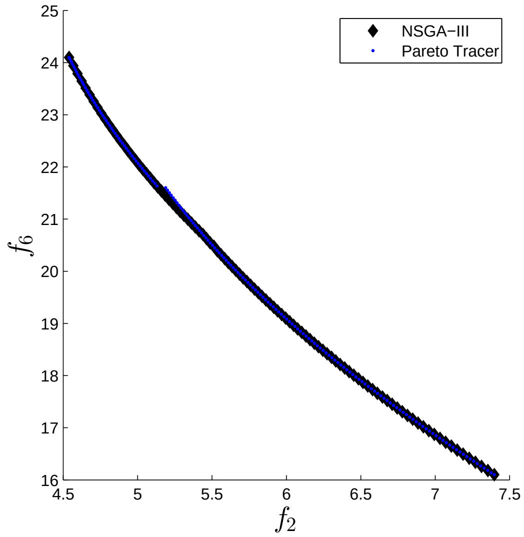


Figure 3.4: Obtained Pareto fronts by PT and NSGA-III for the PIM model defined by f_2 and f_6.

Finally, we consider a three-objective problem defined by the objectives f_1, f_5, and f_6. Since f_5 is multi-modal we first run NSGA-III to obtain a rough approximation of the two-dimensional Pareto front, see Figure 3.6. In order to refine the obtained solutions, we apply PT where we feed this algorithm with each of the individuals from the final population of NSGA-III. As it can be seen, a much better approximation of the entire Pareto front can be obtained. To achieve this result, we have given NSGA-III a budget of 150,000 function calls, which has led to 100 non-dominated solutions. In the second step, PT has used 64,369 function evaluations and 5,877 Jacobian evaluations (leading to 8,7877 function evaluations when using automatic differentiation) leading to a total of 1,948 non-dominated solutions. We think that this combination is most effective for three-objective problems which is confirmed by other computations.

From all results it can be seen that all objectives are in conflict with each other. Thus, one cannot expect to be able to compute suitable finite size approximations of the entire Pareto set/front for problems with more objectives. That is why we will consider applications of the Pareto Explorer for this case in the next section.

Many-objective PIM design

Here, we consider the entire design problem that consists of seven objectives. As one cannot expect any more to compute suitable approximations of the entire solution set (which can then be presented to the decision maker), we have to restrict ourselves to some selected hypothetical scenarios that can occur. We stress, however, that these

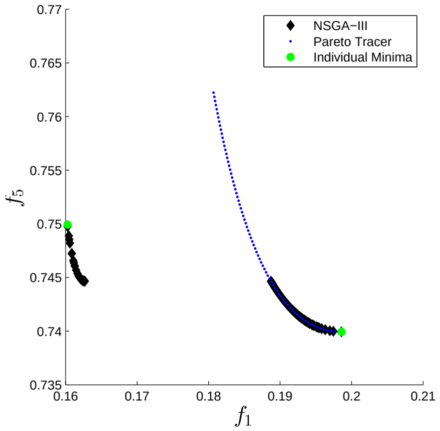


Figure 3.5: Obtained Pareto fronts by PT and NSGA-III for the PIM model defined by f_1 and f_5.

are just illustrators for a possible decision making. The decision making process for a given problem will heavily depend on the given setting and on the preferences of the decision maker.

For all cases we have chosen to take $x_0 = (210.00, 4.00, 80.00, 11.00)^T$ as our initial solution, which is the middle point for each variable in the considered range of the sampling process. Hence, x_0 is chosen as our initial solution for Step 1 of the PE. This is done for simplicity and to have the same starting point for all scenarios and to show the effect of the different steerings. We stress, however, that in principle any other starting point could be taken or computed. For the demonstration of Step 2 of the PE, we consider the four following scenarios:

Scenario 1 (S1): For this scenario, we try to minimize the values of f_1 and f_5 at the same time. Thus, the direction that we consider is $d_y = (-1, 0, 0, 0, -1, 0, 0)^T$ with a step size $\tau = 0.03$. That is, we are interested in solutions that reduce both the warpage and the shear stress (in the same amount) during the search along the Pareto front.

Scenario 2 (S2): Here, our goal is to minimize f_2 and f_6 at the same time. Thus, the direction that we consider is $d_y = (0, -1, 0, 0, 0, -1, 0)^T$ with a step size $\tau = 0.01$.

Scenario 3 (S3): Here, we want to minimize the functions f_1, f_5, and f_6 at the same time. The direction that we consider is then $d_y = (-1, 0, 0, 0, -1, -1, 0)^T$ with a step size $\tau = 0.01$.

Scenario 4 (S4): Finally, we want to minimize the functions f_3, f_5, and f_6 at the same time. The direction that we consider is thus $d_y = (0, 0, -1, 0, -1, -1, 0)^T$ with a step size $\tau = 0.01$.

The computational cost of these scenarios are presented in Table 3.3 and a comparison of the values with our model and the values with the simulator are presented in Table 3.4.

Table 3.3: Computational cost of the PE for the Scenarios 1-4 on the PIM.

	S1	S2	S3	S4
Solutions	323	218	212	205
Function Evaluations	324	226	227	217
Jacobian Evaluations	324	226	227	217

Table 3.4: Comparison of the values from our model (F_M) against the simulated values (F_S) on the PIM.

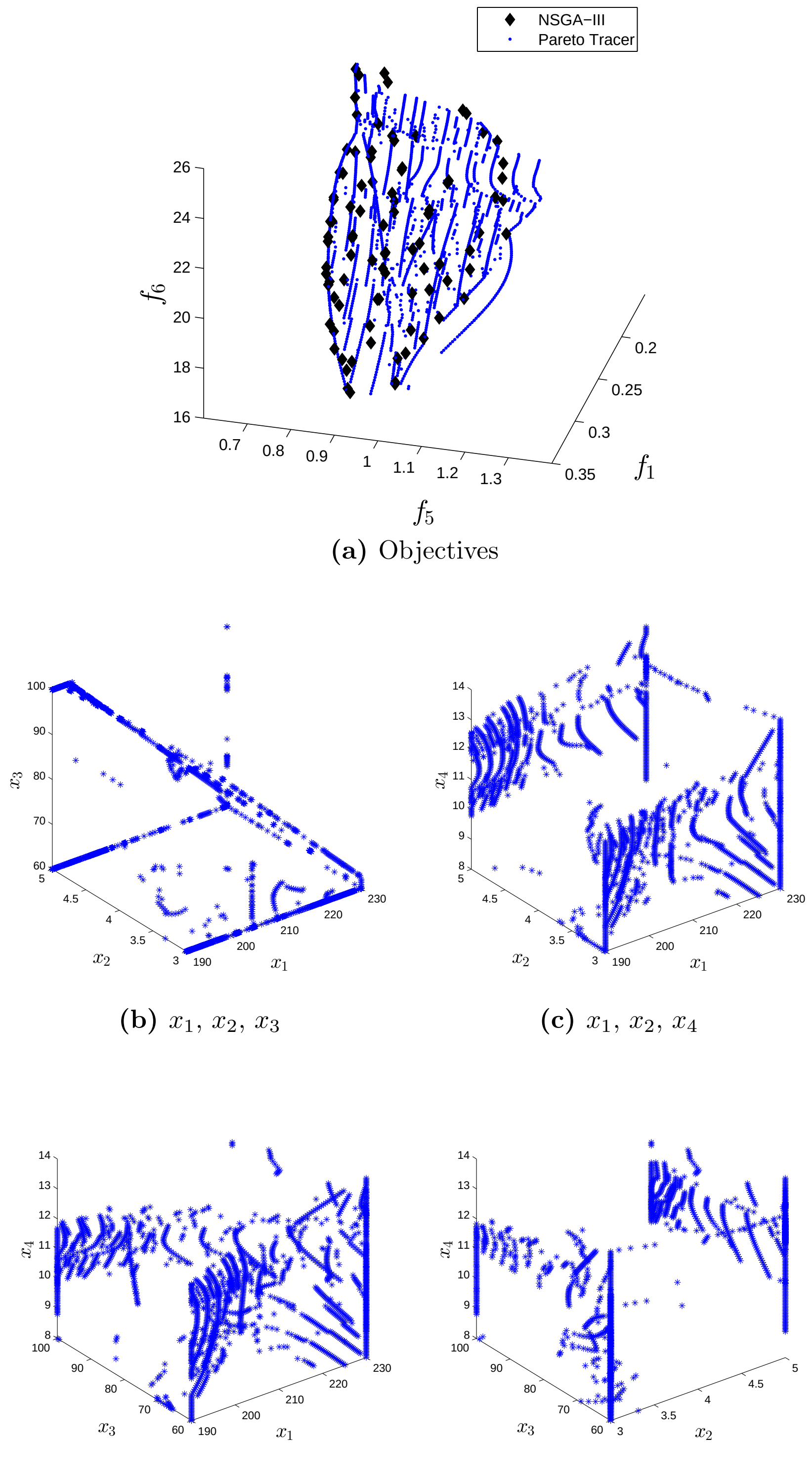
Initial Configuration							
x_0	210.0000	4.0000	80.0000	11.0000			
$F_S(x_0)$	**0.2016**	5.6565	9.7470	0.0717	**0.8690**	20.1000	11.9460
$F_M(x_0)$	**0.2040**	5.7271	9.7329	0.0713	**0.8774**	20.1000	11.8221
Case 1							
x_{323}	230.0000	3.0000	60.0000	13.0163			
$F_S(x_{323})$	**0.1887**	5.2977	8.3664	0.0854	**0.7680**	21.1163	12.4440
$F_M(x_{323})$	**0.1896**	5.3191	8.1392	0.0854	**0.7437**	21.1163	12.9238
Case 2							
x_{218}	212.7412	3.3488	60.0000	9.6531			
$F_S(x_{218})$	0.2442	**6.5442**	9.2361	0.0769	1.0300	**18.1019**	7.4741
$F_M(x_{218})$	0.2425	**6.4453**	9.6549	0.0767	0.9258	**18.1018**	11.7376
Case 3							
x_{212}	213.3452	3.3421	60.0000	9.6950			
$F_S(x_{212})$	**0.2437**	6.5210	9.1729	0.0772	**1.0300**	**18.1371**	7.9492
$F_M(x_{212})$	**0.2419**	6.4289	9.6057	0.0770	**0.9199**	**18.1371**	11.7847
Case 4							
x_{205}	217.3894	3.2880	60.7034	9.8649			
$F_S(x_{205})$	0.2414	6.4710	**9.1775**	0.0787	**1.0100**	**18.2529**	8.9651
$F(x_{205})$	0.2404	6.3740	**9.4653**	0.0787	**0.8806**	**18.2529**	11.7933

We can see from Figure 3.7 that both f_1 and f_5 improve their values with respect to the initial $F(x_0)$. However, at the end of the optimization process, we obtain the best value for f_5, while for f_1 the best value was reached in a previous step. It can be also appreciated in Figure 3.8, that f_2 and f_6 are clearly in conflict. Then, when we reduce the value of f_6, the value of f_2 increases. At the end of the optimization process, we obtain the best value for f_6 and the worst value for f_2.

On the other hand, in Figure 3.9 we observe that the functions f_1 and f_6 are directly in conflict, while for f_5 the value depends of both functions. At the end of the optimization process, we obtain the best value for f_6 and the worst value for f_1; for the case of f_5 the initial and the final values are similar, but along the steps such value has a lot of variation. Notice that, the result for this scenario is almost the same than the previous one.

Finally, from Figure 3.10 it is clear that PE reduces two of the three functions. We notice that the values of f_6 and f_3 are always reduced, while the change in f_5 is not constant. At the end of the optimization process, we obtain the best value for f_3 and f_6, while the best values for f_5 is obtained in a previous step. However, notice that, in some step, the values of f_1, f_5 and f_6 are improved with respect of the initial one.

As can be seen, the movement has been performed in all cases according to the desired direction. We have presented here the entire path of solutions. However, in a real decision making process, the DM can of course choose at any time either to accept a computed candidate solution, or to change the direction in which the steering has to be performed.



(a) Objectives

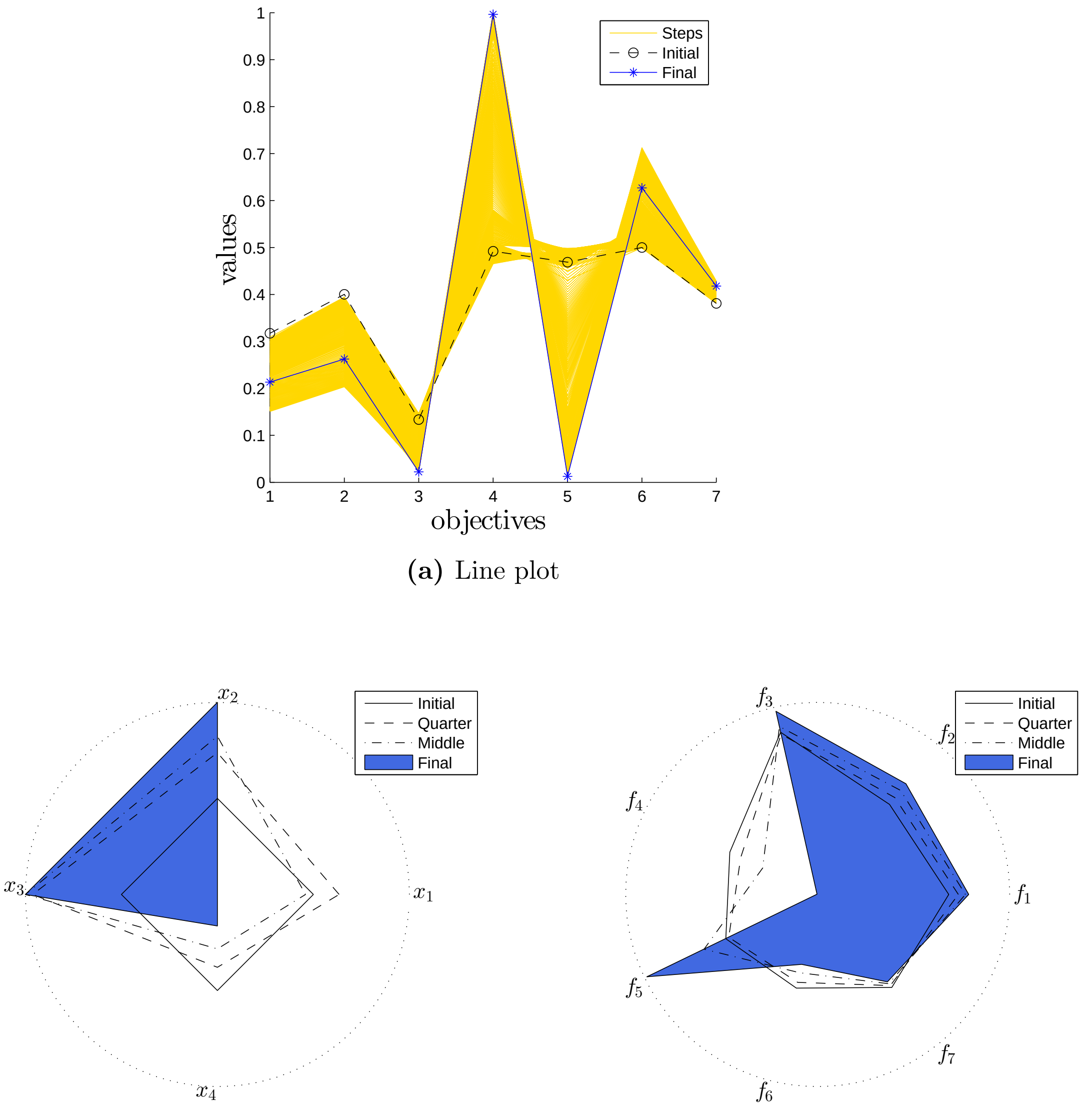
(b) x_1, x_2, x_3

(c) x_1, x_2, x_4

(d) x_1, x_3, x_4

(e) x_2, x_3, x_4

Figure 3.6: Example of PT and NSGA-III on the PIM model for f_1, f_5, and f_6

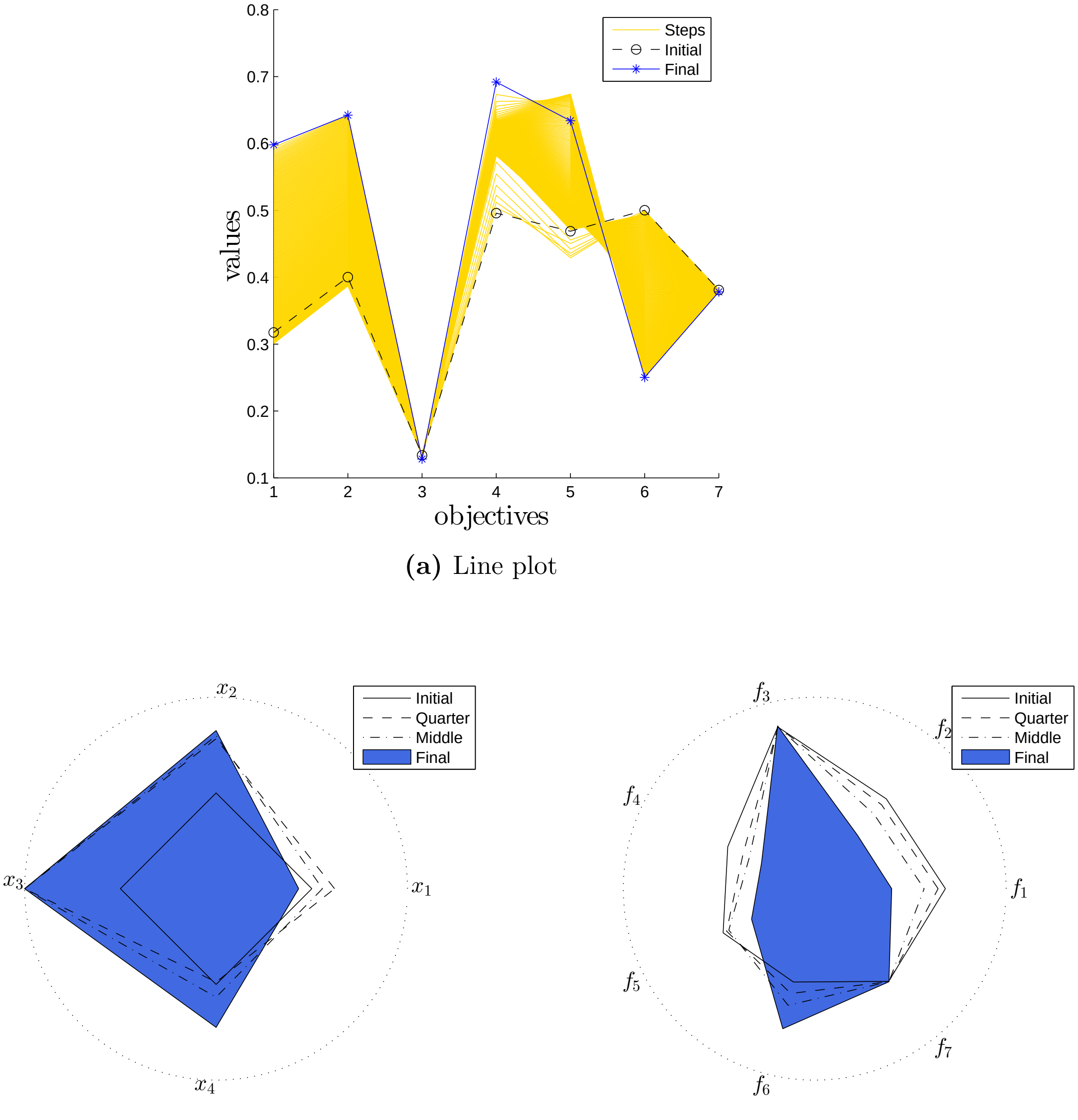


(a) Line plot

(b) Pareto Set

(c) Pareto Front

Figure 3.7: Graphical result on the complete PIM model for the first scenario (minimize f_1 and f_5).

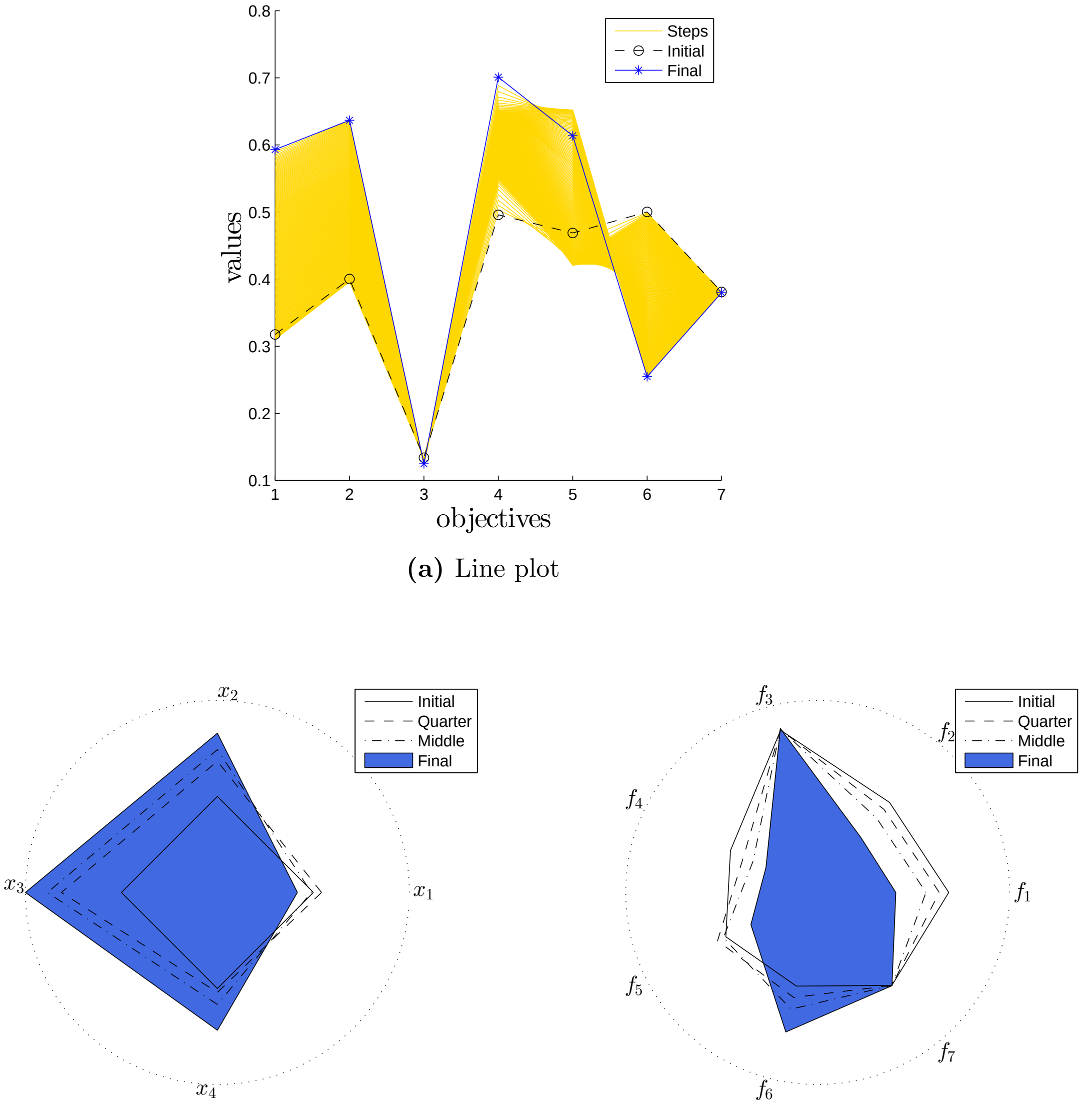


(a) Line plot

(b) Pareto Set

(c) Pareto Front

Figure 3.8: Graphical result on the complete PIM model for the second scenario (minimize f_2 and f_6).

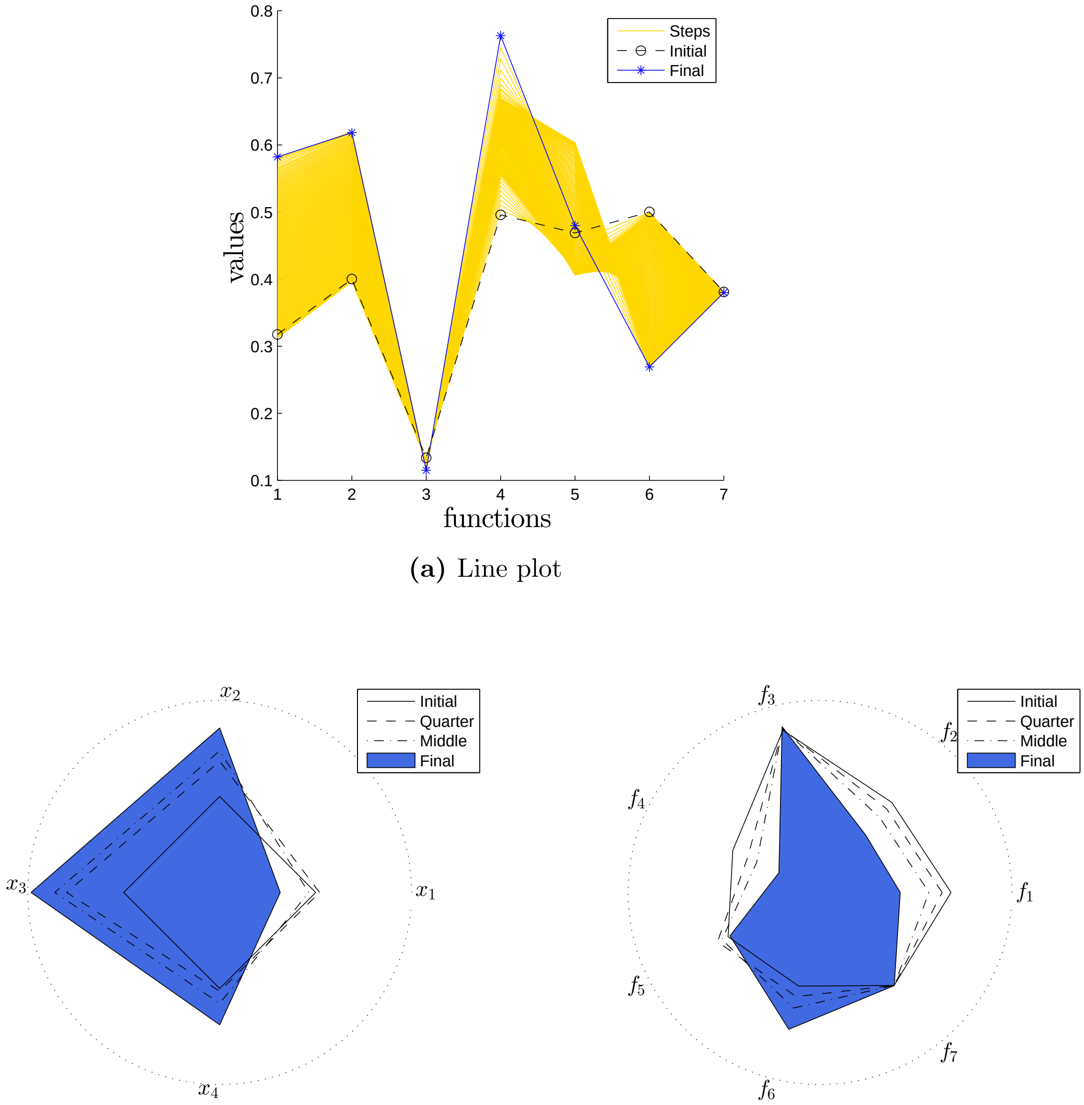


(a) Line plot

(b) Pareto Set

(c) Pareto Front

Figure 3.9: Graphical result on the complete PIM model for the third scenario (minimize f_1, f_5, and f_6).



(a) Line plot

(b) Pareto Set

(c) Pareto Front

Figure 3.10: Graphical result on the complete PIM model for the last scenario (minimize f_3, f_5, and f_6).

Chapter 4

Exploration in Objective Space

In this chapter, we present two approaches to deal with MOPs with different smoothness assumptions. The idea is to perform a steering in objective space similar to the continuous PE. In Section 4.1, we present a fine tuning method to deal with general MaOPs; while, in Section 4.2 we treat linear MaOPs.

The motivation of our proposals is to allow the DM to navigate along the Pareto front by starting from a given initial solution x_0, possibly coming from the output of any available optimization technique as with the continuous PE. This initial solution x_0 is to be viewed as a departure point in the objective space from where the DM can refine his/her preferences by discovering on-line the vicinity of x_0, and eventually finding new preferred points in the objective space. Consequently, we shall provide the DM with the necessary tools in order to explore a whole path of solutions being as near as possible to the PF and to locally explore the landscape of Pareto optimal solutions in an iterative manner, i.e., from one solution to a nearby one. For this purpose, the DM is required to provide a direction in the objective space, in which the search process shall be steered. Informally speaking, steering the search along the given direction means providing the DM with a sequence of candidate solutions x_i, $i = 1, \ldots, N$, that can be viewed as forming a path in the objective space and such that every solution x_i is improving the previous solution x_{i-1} with respect to the direction given by the DM in the objective space.

4.1 Fine tuning method and application to knapsack

For the target framework to work it is important to keep in mind the notion of Pareto optimality when performing a movement in objective space. For instance, assuming that the starting solution is a Pareto optimal solution, then it is obviously not possible to improve all the objectives simultaneously. Consequently, the DM can still define a direction which does not involve an optimal movement, because *no prior knowledge* on the shape of the Pareto front is assumed. In the rest of this section, we provide a step-by-step description of the proposed framework and the necessary algorithmic components for its proper realization. This shall also allow us to better highlight the different issues one has to address in such context.

4.1.1 Framework for the fine tuning method

We assume that the DM has a preferred direction d in objective space. More formally, given an initial solution x_0, we assume that the DM is interested in a solution x_1 which is in the vicinity of x_0 such that the following holds:

$$F(x_1) \approx F(x_0) + td, \tag{4.1}$$

where $t > 0$ is a given (typically small) step size.

As it is very unlikely that such a point x_1 exists where the exact equality in Equation (4.1) holds (note also that the Pareto front around $F(x_0)$ is not known), we propose to consider a 'best approximation' using an 'approximated' *reference point* Z_1 as follows:

$$Z_1 := F(x_0) + \bar{t}d, \tag{4.2}$$

where $\bar{t} > 0$ is a given, fixed (problem dependent) step size. Then, we propose to compute the 'closest' Pareto optimal solution to the reference point Z_1 in the objective space, which will hence constitute the next point x_1 to be presented to the DM. Notice that we still have to define a metric specifying the closeness of optimal points with respect to the reference point – this will be addressed later. Once x_1 is computed, the DM can consequently change his/her mind or not, by providing a new direction or by keeping the old one. The framework then keeps updating the sequence of reference points and providing the DM with the corresponding closest optimal solutions in an interactive manner. The proposed framework is hence able to provide the DM with a sequence of candidate solutions such that the respective sequence of objective vectors (ideally) performs a movement in the specified direction d.

In Algorithm 4.1, we summarize the high-level pseudo code of the proposed method. We first remark that the procedure RefPoint implements the idea of transforming the direction d provided by the DM into a reference point, which we simply define as follows:

$$Z_i = F(x_{i-1}) + \delta d, \tag{4.3}$$

where x_{i-1} is the previous (starting) solution and δ a parameter specifying the magnitude of the movement in objective space. Actually, δ can be viewed as the preferred

Algorithm 4.1 Fine Tuning Framework

```
Require: starting point $x_0$
Ensure: sequence $\{x_i\}$ of candidate solutions
  for $i = 1, 2, \ldots$ do
      Let $d \in \mathbb{R}^k$                          ▷ Search direction in objective space
      Let $\delta \in \mathbb{R}_+$                       ▷ Step size in objective space
      $Z_i = \textsc{RefPoint}(x_{i-1}, d)$               ▷ Reference point in step $i$
      $\text{SOP}_i = \text{G}(Z_i)$                      ▷ Next single objective problem to solve
      $x_i = \textsc{Emo\_Optimizer}(\text{SOP}_i)$       ▷ evolutionary search for the next solution
  end forreturn $X := \{x_1, x_2 \ldots\}$
```

Euclidian distance between two consecutive solutions $\|F(x_{i-1}) - F(x_i)\|$ in objective space. It hence defines the preferred step size of the required movement, which is kept at the discretion of the DM. Notice also that both the direction d and the step size δ can be changed interactively by the DM, which we do not include explicitly in the framework of Algorithm 4.1 for the sake of simplicity.

Given this reference point, one has to specify more concretely which solution should be sought for the decision maker. This is modeled by function G, which takes the current reference point into account and outputs a (single-objective) SOP to be solved. At last, the procedure Emo_Optimizer refers to the (evolutionary) algorithm that effectively computes the next solution to be presented to the DM. At this stage of the presentation, it is still not fully clear how to define function G and how to effectively implement the evolutionary solving procedure, which is at the core of this paper. Before going into the technical details of these crucially important issues, let us comment on Figure 4.1 showing two hypothetical scenarios in the two-objective case, chosen for the sake of a better visualization. For $F(\overline{x}_0)$, the reference point $\overline{Z}_1$ is feasible when choosing the direction $d = (1, -1)^T$. That is, there exists a point $\overline{x}$ such that $F(\overline{x}) = \overline{Z}_1$. We want a function $G(\overline{Z}_1)$ that prevents x to be actually chosen. Instead, the solution $\overline{x}_1$ should be a natural candidate since it is a Pareto optimal solution where $F(\overline{x}_1)$ is the closest element to $\overline{Z}_1$ in the Pareto front. The second scenario is for a given point x_0 such that Z_1 is infeasible. Here, it is clear that the solution of $G(Z_1)$ must be a Pareto optimal solution whose image $F(x_1)$ is the closest to the given reference point. Notice that in both cases, the Pareto Front is not known when defining function $G(Z)$.

In the following, we propose a possible answer for the definition of G, as well as some alternative (single- and multi-objective) evolutionary procedures for solving the corresponding SOP.

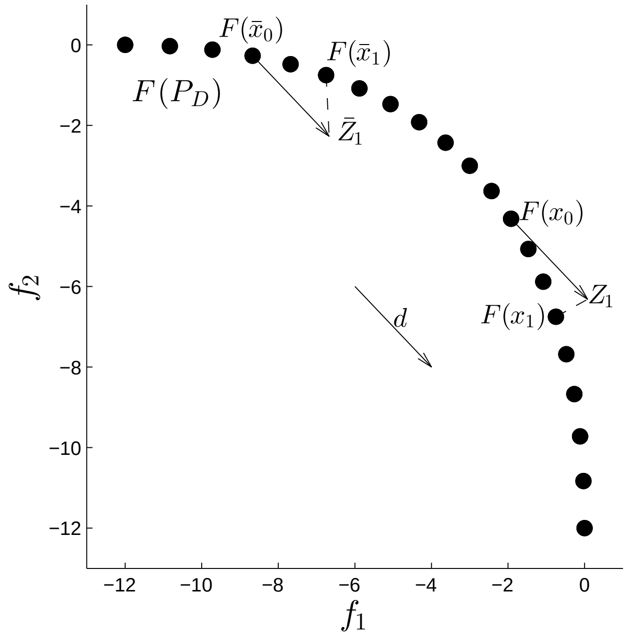


Figure 4.1: Illustrative example of fine tuning in objective space.

4.1.2 Framework instantiation

Since the direction provided by the DM could be arbitrary, and given that we do not assume any prior knowledge neither about the Pareto front, nor on the initial solution from where to steer the search; we propose the following modeling of function $G(x|Z)$, defining the next point to be computed by our framework. We rely on the WASF (see Section 2.3.1). The motivation of using such a function is that the optimal solution to Problem (2.31) is a Pareto optimal solution [Miettinen and Mäkelä, 2002], independently of the choice of the reference point. This is an interesting property of the WASF that allows us to deal with reference points that might be defined on the feasible or the infeasible region of the objective space. Notice that this is to contrast to other scalarizing functions, such as the widely-used Chebychev function, that constraints the reference point to be defined beyond the Pareto front.

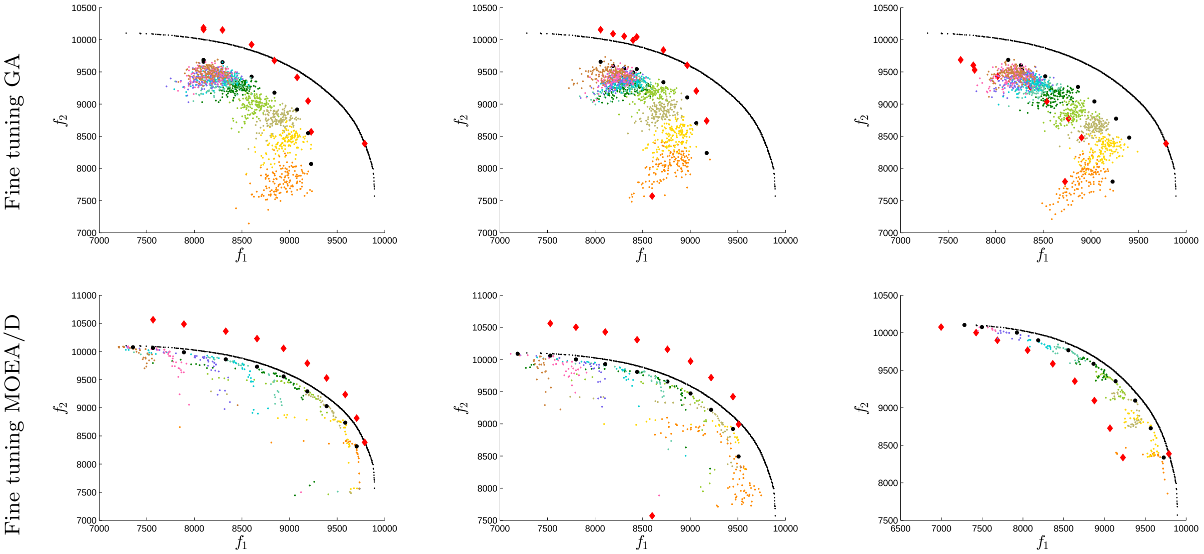
In our framework, the WASF is intended to capture the DM preferences, expressed by the reference point computed with respect to the DM's preferred direction. However, the weighting coefficient vector still has to be specified. It is known that for a given reference point, the solutions generated using different weight vectors are intended to produce a diverse set of solution in the objective space. We here choose to set the weight vector λ as $(1/k, 1/k, \ldots, 1/k)^T$, which can be viewed as one empirical choice implying a relative fairness among the objectives while approaching the reference point.

The evolutionary solving process

In order to solve the previously defined SOP, we investigate two alternative evolutionary approaches.

The first one consists in using a standard *Genetic Algorithm* (GA). More precisely, and with respect to the experimented knapsack problem, we use the same evolutionary mechanisms and parameters than [Alves and Almeida, 2007], i.e., a parent selection via a random binary tournament with probability 0.7, an elitist replacement strategy that keeps the best individual, a binary crossover operator with probability 0.5, a single point mutation, and an improve and repair procedure [Alves and Almeida, 2007] for handling the capacity constraints. However, the initial population of the GA is adapted with respect to the iterations of our proposed framework as follows. Each time the SOP defined by the WASF and the corresponding reference point is updated, we initialize the population with 1/4 of the best individuals from the previous iteration, that we complement with randomly generated individuals. In the first iteration of our framework, the initial population is generated randomly. In our preliminary experiments, this was important in order to obtain a good trade-off between quality and diversity within the evolutionary process. Actually, this observation leads us to consider the following alternative evolutionary algorithm, where population diversity is maintained in a more explicit manner, by using an MOEA for solving the target SOP.

More precisely, our second alternative solving procedure is based on an adaptation of the MOEA/D framework [Zhang and Li, 2007]. We recall that MOEA/D is based on the decomposition of a given MOP into multiple subproblems using different weight vectors, which are then solved cooperatively. In contrast to the original algorithm, where the entire Pareto front is approximated using an ideal reference point and a diverse set of weight vectors, typically generated in a uniform way in the objective space; we are here interested in a single solution with respect to the target reference point. Hence, we still consider a set of uniformly-distributed weight vectors, but we use the WASF where the reference point is fixed in order to focus the search process on the region of interest for the DM. At the end of the MOEA/D search process, we output the best-found solution for the weight vector $\lambda = (1/k, 1/k, \ldots, 1/k)^T$, which precisely corresponds to the target SOP defined with respect to the DM preferred direction. Similarly to the GA, our preliminary experiments revealed that the choice of the initial population for MOEA/D has an important impact on the quality of the target solution. Accordingly, apart from the first iteration where the initial population is generated at random, we choose to systematically initialize MOEA/D with the population obtained with respect to the previous reference point. Due to the explicit diversity of the MOEA/D population, this initialization strategy revealed a reasonable choice in our initial experiments.



$F(x_0) = (9789, 8387)^T$ $d = (0, 1)^T$

$F(x_0) = (8600, 7570)^T$ $d = (0, 1)^T$

$F(x_0) = (9789, 8387)^T$ $d = (-1, 0)^T$

Figure 4.2: Illustrative scenarios on the MOKP for the bi-objective case. The best-known PF approximation is in thin black points. Reference points are shown using red squared points. The output solutions are shown as circled black point. The population is depicted using a variable color scale. $\delta = 500$.

Illustrative scenarios

To exemplify the possible scenarios, we experiment in the following the tuning method on the *multiobjective (multi-dimensional) 0-1 knapsack problem* (MOKP) with different assumptions. This is in order to highlight the behavior of the framework under some possible representative scenarios and to identify the main raised issues.

We define the exemplary scenarios changing the input values of the fine tuning method, i.e., the initial optimal solution $F(x_0)$, the direction in objective space d and the step size δ. For each investigated scenario, we provide plots rendering the computed reference points, the projection of the selected solutions (x_i) in the objective space, and the final population of each of the two considered evolutionary algorithms, together with the best-known PF approximation. This is reported in Figure 4.2 for a bi-objective MOKP instance from [Zitzler and Thiele, 1999]. Notice that, since we are interested in the impact of the input parameters, we assume that the initial solution x_0 could be optimal or not, which implies that the first-obtained reference point can also be optimal or not. Thus, by simplicity we omit the first update of the reference point, and we consider that $Z_1 = F(x_0)$. At last, we consider 10 iterations of the proposed method, a population size of 150, and $7,500$ function evaluations when running the solution procedure at each iteration.

In Figure 4.2 (left), we consider a Pareto optimal solution as a starting point and a fixed direction vector (provided by the DM) corresponding to the scenario where the second objective is to be refined repeatedly. We can clearly see that running the proposed framework is able to gradually improve the output solutions and to effectively steer the search along the desirable input direction. However, the output solutions are not necessarily optimal, which we clearly attribute to the relatively few amount of computational effort used when running the evolutionary solving procedure. Interestingly, using the MOEA/D algorithm as a solving procedure (bottom) for the single-objective reference point target problem appears to work much better than the single-objective GA (top). We clearly attribute this to the diversity issues that the evolutionary process is facing when trying to find Pareto optimal solutions. This is confirmed in our second scenario depicted in Figure 4.2 (middle), where the initial solution is chosen to be a non-optimal one. This second scenario also demonstrates that the proposed approach behaves in a coherent manner even if the solution considered in each iteration is not optimal. Notice that these two scenarios consider the same preferred direction, which is actually pointing to regions where there exist some non-dominated points. In Figure 4.2 (right), we instead consider the scenario where a non-optimal direction $d = (-1, 0)^T$ is provided, that is a direction that points towards a dominated region of the objective space. This leads to the critical situation where the computed reference point might be dominated. Again, we notice that the proposed approach can handle this situation properly and that the MOEA/D-based solving procedure performs better than the GA.

4.1.3 Numerical results

In this section, we present some numerical results of our approach using the modified (fine tuning) MOEA/D as a solver, since it was shown to provide better performance than the fine tuning GA. In order to appreciate the behavior of the proposed method, we compare it against the original MOEA/D algorithm. However, since the original MOEA/D is intended to compute an approximation of the whole PF, a special care has to be taken.

First, the proposed method enables to only output a path of solutions based on the computed reference points. For this reason, we use IGD_Z (see Section 2.5.3) to compare our results. The value of IGD_Z can be straightforwardly computed for our approach using the set of reference points computed at each iteration and the best-available PF approximations as the discretization of the PF. For the original MOEA/D, we consider to first extract from the archive maintained by MOEA/D the nearest solutions (in objective space) to the same reference points computed by our approach. Then, these solutions are considered in order to compute an IGD_Z value for the original MOEA/D. By comparing the IGD_Z values for our method as well as for the original MOEA/D, our intent is to highlight the benefits that can be expected

Table 4.1: Parameters settings of the fine tuning method adopted for obtaining the numerical results reported in Table 4.2. Number of knapsacks (KS), items, population size (P), maximal number of functions evaluations for each reference point (ZEvs), number of considered reference points ($|Z|$), step size δ, initial reference point Z_0 and desirable direction d_k.

KS	**Items**	**P**	**ZEvs**	$\|Z\|$	δ	Z_0	d_k
2	250	150	7500	10	300	$(10000, 8000)^T$	$(0, 1)^T$
2	500	200	10000	10	300	$(16000, 19000)^T$	$(1, 0)^T$
3	100	351	17600	10	200	$(4056, 3314, 3228)^T$	$(0, 1, 1)^T$
3	100	351	17600	10	200	$(4056, 3314, 3228)^T$	$(0, 0, 1)^T$
4	500	455	17500	10	300	$(13643, 14224, 16968, 16395)^T$	$(1, 1, 0, 0)^T$
4	500	455	17500	10	300	$(16716, 16867, 14178, 13234)^T$	$0, 0, 1, 1$

Table 4.2: Numerical results. Number of knapsacks (KS), items, population size (P), maximal number of functions evaluations (Ev), and IGD_Z; minimum, average, standard deviation (in a small font) and a maximum of 20 independent runs were adopted for our experiments. The comparison of results is done with respect to IGD_Z.

KS	**Items**	**P**	**Ev**	**IGD**$_Z$							
				Finite Tuning				original *MOEA/D*			
2	250	150	75000	69.86	80.73	(7.02)	90.22	**40.34**	**47.51**	(4.73)	**59.01**
2	500	200	100000	241.45	271.04	(15.34)	291.85	**153.12**	**173.79**	(10.89)	**192.08**
3	100	351	176000	35.84	**41.62**	(3.56)	**47.93**	**34.05**	46.82	(5.13)	57.03
3	100	351	176000	**24.10**	**30.85**	(4.82)	41.29	25.63	32.54	(4.35)	**40.79**
4	500	455	175000	**184.73**	**228.05**	(26.95)	**289.81**	365.29	494.72	(66.79)	577.90
4	500	455	175000	**144.85**	**198.48**	(20.12)	**231.05**	311.34	463.83	(70.58)	589.30

when *locally* steering the search along a preferred direction in an interactive way, with respect to computing a *global* approximation set form which we steer the search *a posteriori*. It is however worth noticing that such a comparison is only conducted for the sake of illustrating the accuracy of our approach and its effective implementation which should not be considered as an alternative to existing (global) multi-objective optimization algorithms.

In the following, we consider some benchmark instances of the considered MOKP[1], as specified in Table 4.1, which also summarizes the different parameters settings used for the proposed method. MOEA/D was adopted using the same settings as in its original paper [Zhang and Li, 2007]. Notice that, overall, the same number of function evaluations are used for both the original MOEA/D and the proposed method. Table 4.2 shows the obtained results for the consider scenarios over 20 independent runs for each instance and algorithm.

We notice that, for $k = 2$, the original MOEA/D is able to obtain better results than the proposed Fine Tuning method. This is because MOEA/D can generate

[1] http://www.tik.ee.ethz.ch/sop/download/supplementary/testProblemSuite/

points close to the entire PF when the number of objectives is limited. However, we can observe that, the larger the number of objectives, the better the Fine Tuning approach. This is because MOEA/D requires more approximation points and function evaluations in order to cover the entire PF as the dimensionality grows, while the Fine Tuning approach is able to naturally focus on certain regions of the PF. We remark that this fact also improves the execution time, because the Fine Tuning approach does not require any external archive, while for MOEA/D to output a high-quality global PF approximation, an archive is actually used for the MOKP.

4.2 Pareto Explorer for Linear MaOPs

When both the objectives and the constraints are linear, we have a linear MOP; respectively, a linear MaOP in case $k > 3$. Such problems can be expressed as

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & Px \\ \text{s.t.} \quad & Bx \leq b, \end{aligned} \tag{4.4}$$

where $P \in \mathbb{R}^{k \times n}$, $B \in \mathbb{R}^{m \times n}$, $x \in \mathbb{R}^n$ with $0 \leq x$, and $b \in \mathbb{R}^m$.

In the following we assume that we are given a (Pareto) optimal solution x_0 of a linear MaOP and that the DM is interested in a local exploration of x_0 with respect to $F(x_0) = Px_0$ along the Pareto set/front. For this, we suggest in the following two different strategies, (i) one that allows to choose a direction in image space, and (ii) another one that allows to change the weights of the different objectives.

4.2.1 Change in Objective Space

The first method is based on the assumption that the DM has a preferred direction $d_k \in \mathbb{R}^k$, to explore in the *objective space*. That is, given a particular point Px_0 on the Pareto front, he/she is interested on generating new solutions x_{new}, such that their objective values are ideally given by

$$F(x_{new}) \approx F(x_0) + td_k, \tag{4.5}$$

where $t > 0$.

In order to conserve the linearity of the problem, we consider an equivalent formulation (see [Steuer and Choo, 1983]) of Equation (2.31) as follows:

$$\begin{aligned} \min_{(x,\alpha)\in\mathbb{R}^{n+1}} \quad & \alpha - \rho \sum_{i=1}^{k} f_i(x) \\ \text{s.t. } & w_i(f_i(x) - z_i) \le \alpha \\ & i = 1, \dots, k, \\ & x \in S. \end{aligned} \tag{4.6}$$

Since the shape of the Pareto front is not known, it is also unclear (and rather unlikely) that such solutions indeed exist. Instead, one could numerically trace a solution path $\{x_i\}$ through those objective values that best fit (4.5), via defining the sequence of reference points

$$\begin{aligned} Z_0 &:= F(x_0) \\ Z_{i+1} &:= Z_i + t_i d_z, \quad i = 0, 1, \dots, \end{aligned} \tag{4.7}$$

where each $t_i > 0$, and solving Problem (4.6) for each reference point. Now, the linear MaOPs (Problem (4.6)) can be expressed in the following way:

$$\begin{aligned} \min_{\hat{x}\in\mathbb{R}^{n+1}} \quad & \hat{c}^T \hat{x} \\ \text{s.t.} \quad & \hat{B}\hat{x} \le b, \\ & \hat{P}\tilde{x} \le \hat{Z}_i, \end{aligned} \tag{4.8}$$

where

$$\hat{c} := [-\rho e P \,|\, 1], \tag{4.9}$$

$\hat{c} \in \mathbb{R}^{n+1}$, $e = (1, \dots, 1)^T \in \mathbb{R}^k$ $P \in \mathbb{R}^{k\times n}$ as in (4.4):

$$\hat{x} := (x^T, \alpha)^T, \tag{4.10}$$

$\hat{x} \in \mathbb{R}^{n+1}$.

$$\hat{B} := [B \,|\, 0_m], \tag{4.11}$$

$B \in \mathbb{R}^{m\times n}$ defined as in (4.4), $0_m = (0, \dots, 0)^T \in \mathbb{R}^m$, and $b \in \mathbb{R}^m$ defined as in (4.4).

$$\hat{P} := [WP \,|\, -e], \tag{4.12}$$

$\hat{P} \in \mathbb{R}^{k\times(n+1)}$, $W \in \mathbb{R}^{k\times k}$ a diagonal matrix with $W_{i,i} = w_i,\ i = 1, \dots, k$; and $\hat{Z} \in \mathbb{R}^k$ as

$$\hat{Z}_i := W Z_i. \tag{4.13}$$

In the particular case that all the weights w_i are equal, i.e.,

$$w_1 = \cdots = w_k,$$

which is equivalent to an unbiased sum of the objectives, the problem gets simplified as follows: we can consider W as the $\mathbb{R}^{k\times k}$ identity matrix and we have then for the constraints:

$$\begin{aligned} \hat{P}\tilde{x} \leq \hat{Z} &\Leftrightarrow [WP \,|\, -e]\hat{x} \leq WZ \\ &\Leftrightarrow [P \,|\, -e]\hat{x} \leq Z. \end{aligned} \tag{4.14}$$

The above way is how the constraints are considered in this approach.

Proceeding in this manner one obtains a sequence of candidate solutions such that the respective sequence of images performs (ideally) a movement in the d_k-direction. Algorithm 4.2 shows the pseudo code of this method. The choice of t_{i-1} defines the distance between two consecutive solutions $\|F(x_{i-1})-F(x_i)\|$, in objective space; and is thus problem dependent. One problem that is still remaining is to detect when to stop the process. We assume here that the DM performs a local search around x_0, i.e., he/she will stop after a few iterations. If this is not the case, the search will be stopped at an iteration step $i+1$ such that $x_{i+1} \approx x_i$, i.e., when the process generates very similar or the same candidates. This case occurs when the candidates reach the boundary of the Pareto set.

Algorithm 4.2 PE: Movement in Objective Space

```
Require: Initial optimal solution $x_0$, desired direction $d_k$ in objective space.
Ensure: Sequence $X = \{x_1, x_2, \ldots\}$ of candidate solutions.
$Z_0 := Px_0$
for $i = 0, 1, \ldots$ do
    choose step size $t_i > 0$
    set $Z_{i+1} := Px_i + t_i d_k$
    solve (4.8) using $Z_{i+1}$ starting from $x_i$ to obtain $x_{i+1}$
end for
```

Example For illustration purposes we consider the following bi-objective problem (i.e., $k = 2$):

$$\begin{aligned} \min_{x\in\mathbb{R}^2} F(x) &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x \\ \text{s.t.} & \\ \begin{pmatrix} -7 & -11 \\ -6 & -2 \end{pmatrix} x &\leq \begin{pmatrix} -77 \\ -30 \end{pmatrix} \\ 0 \leq x_1 &\leq 11 \\ 0 \leq x_2 &\leq 15. \end{aligned} \tag{4.15}$$

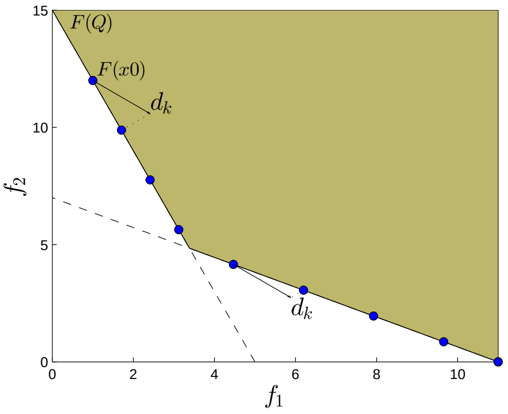


Figure 4.3: Illustrative example of the steering in objective space for linear MOPs.

The Pareto front of MOP (4.15) is convex and consists of two line segments (compare to Figure 4.3). When choosing

$$x_0 = (1, 12)^T$$

with

$$Px_0 = (1, 12)^T$$

and

$$d_k = (1, -1)^T,$$

then the application of Algorithm 1 performs a movement right down the Pareto front. Note that for the first three iterations the solutions are dominating the reference points, while afterwards, dominated solutions are obtained. Hence, the the true Pareto front does not have to be known when choosing the direction d_k. This holds for MaOPs of any value of k.

4.2.2 Change in Weight Space

The second approach for the guided space exploration is based on the assumption that the DM has a certain idea about the importance of preference for each objective; and he/she would like to make a related change in the corresponding *weight space*. More precisely, given a vector w of weights, the task is to minimize Problem (4.6) (respectively Problem (4.8)) for the given weight w.

The change in weight space can be realized as follows: assume a convex weight $w^{(0)}$ is given; and also a vector $\Delta w \in \mathbb{R}^k$ which represents the desired changes, with the next property

$$\sum_{i=1}^{k} \Delta w_i = 0. \tag{4.16}$$

Then, the new weight vector can be computed via

$$w^{(1)} := w^{(0)} + t\Delta w, \tag{4.17}$$

where $t > 0$ is a given step size. It is easy to see that if t is small enough, then $w^{(1)}$ is also a convex weight.

One problem that remains is that the initial weight vector $w^{(0)}$ is not known exactly unless the initial solution x_0 is obtained by solving (4.6) directly. On the other hand, a good approximation to the utopian point Z^* is necessary to get points along all the PF.

Thus, at each iteration of the method, Equation (4.6) is solved for the current $w^{(i)}$ and Z^*. Algorithm 4.3 shows the pseudo code of this second approach. The algorithm has to be stopped if one of the weights becomes non-positive.

Algorithm 4.3 PE: Change in Weight Space

```
Require: starting point $x_0$, initial weight $w^{(0)}$, vector of changes $\Delta w$, tolerance $\epsilon$ for
  maximal change in objective space.
Ensure: Sequence $X = \{x_1, x_2, \ldots\}$ of candidate solutions.
  get an approximation of $Z^*$
  for $i = 1, 2, \ldots$ do
      choose $t_i > 0$
      set $w^{(i+1)} := w^{(i)} + t_i \Delta w$
      solve (4.6) starting with $x_i$ using $w^{(i)}$ as weight vector and $\epsilon$ as tolerance to
  obtain $x_{i+1}$
  end for
```

Example We revisit the bi-objective problem (4.15) with x_0 as in Example 1. The obtained sequence of objective vectors

$$Px_i$$

for

$$w^{(0)} = (0.95, 0.05)^T,$$

with

$$\Delta w = (-0.5, 0.5)^T \quad \text{and} \quad t_i = 0.15,$$

is shown in Figure 4.4. Again, we obtain a movement right down the Pareto front which makes sense as the change of weights represents an emphasis on the second objective, while the first objective gets less attention.

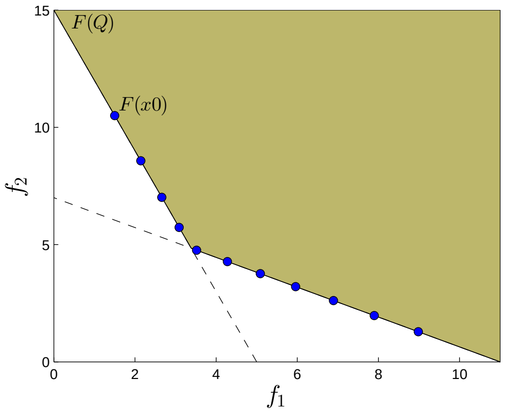


Figure 4.4: Illustrative example of the change in weight space for linear MOPs.

4.2.3 Numerical Results

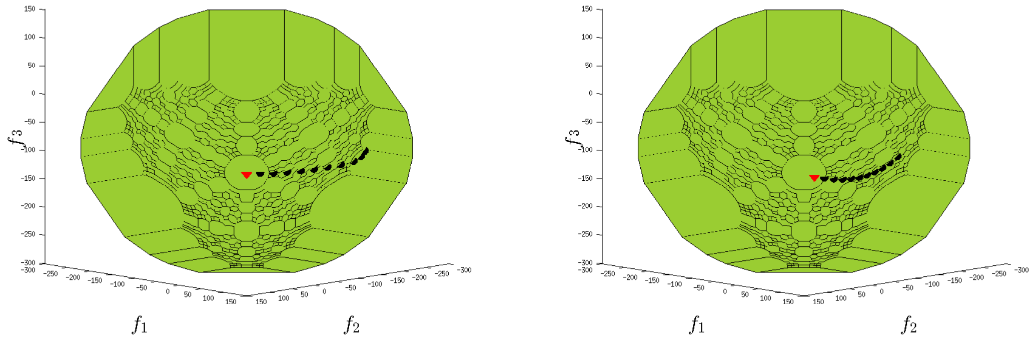
Here we present two possible scenarios on the scalable benchmark problem *bensolvedron* (see [Löhne and Weißing,]). This is a problem with k objectives (where k can take any integer value) and $n = (k + 2p)^k$ variables and constraints. We have chosen here to consider the problem for $k = 3$ (which allows us to show the entire Pareto front) and for $k = 5$. In both cases, we tried to get a similar path of solutions for both approaches. We stress that it is impossible to show a comparison to other methods as those either compute single solutions or aim at computing the entire solution set. For both cases, a comparison is unfair. The further investigation of our method, however, will be subject of future studies.

3-objective case

For our first scenario we fix $k = 3$ and $p = 2$ leading to a 3-objective problem, with $n = 343$ decision variables and the same number of constraints. We have chosen a point x_0 with objective vector

$$Px_0 = (-160, -160, -160)^T.$$

For the movement in objective space, we have chosen $d_k = (0, -2, -1)^T$, and step size $t_i = 15$. To obtain a similar solution path for the change of weights method, we have chosen $w^{(0)} = (1/3, 1/3, 1/3)^T$, $d_w = (0.3, -0.2, -0.1)^T$ and $t_i = 0.1$. For each approach, we have computed 10 iterations; the results are shown in Figure 4.5. The Pareto front has been computed with BenSolve ([Löhne and Weißing,]). The computational time to compute the entire set on a computer with 6.9 GB of RAM and processor AMD A8-4555M was around two seconds (1.965 s). The computation of the 10 solutions took 0.80 and 0.94 seconds for the two approaches.



(a) Movement in objective space.

(b) Change of weights.

Figure 4.5: Graphical result on the bensolvedron problem with three objectives.

5-objective case

Next we consider $k = 5$ and $p = 0$, i.e., we have a 5-objective problem with $n = 3,125$ decision variables and constraints. We have taken a point x_0 with

$$Px_0 = (-160, -160, -160, -160, -160)^T$$

as initial solution, which is not Pareto optimal. For the movement in objective space, we have chosen

$$d_k = (0, -2, -1, 0, 1)^T$$

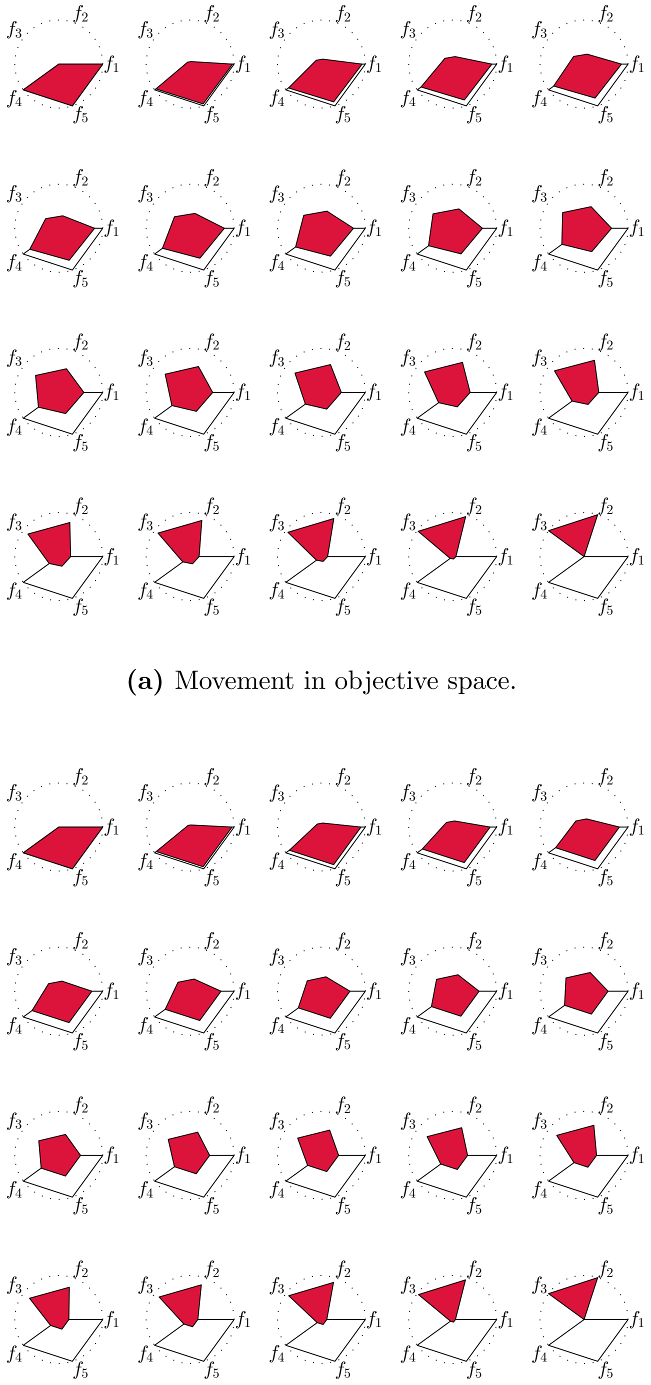
and $t_i = 15$. For the change of weights, we have taken

$$w^{(0)} = (0.2, 0.2, 0.2, 0.2, 0.2)^T,$$

and

$$\Delta w = (0.1, -0.2, -0.1, 0.1, 0.1)^T$$

with $t_i = 0.01$. We computed 20 steps for each approach. The results are shown in Figure 4.6. In this case, we are not able to show the Pareto front. We have stopped the execution of `BenSolve` after a running time of 6 hours, and it was not able to deliver the Pareto set within this time, while the computation of the 20 solutions took 27.99 and 40.21 seconds for the each of the two approaches.



(a) Movement in objective space.

(b) Change of weights.

Figure 4.6: Result of the 5-objective case.

Chapter 5

MOEA-PT

In this chapter, we present a hybrid of a MOEA with the PT [Cuate et al., 2019]. As it is explained in Section 2.6.1, the PT method is a continuation strategy that can efficiently perform movements along the Pareto front of a given MOP. However, this reconstruction process is carried out locally, which involves that we must provide a reduced set of relevant approximated solutions, i.e., with reasonably good convergence and dispersion over the front. Therefore, before using the PT, we need a MOEA to produce this first set of promising solutions. This is closely related to the first phase of PE, as its aim is also to obtain a reduced set of candidate solutions. The difference is that with PE, we only need to select one element of the resulting set as an initial point for the steering phase; while for the PT, we need to take advantage of every point of the set in order to approximate the entire PS.

For the treatment of Bi-Objective Optimization Problems (BOPs), we have decided to use a micro-GA, that is based on NGSA-II. On the other hand, for MOPs with more than three objectives, a modified implementation of NSGA-II [Deb et al., 2002], is developed. In a second and last phase, PT takes over and refines the obtained solutions. In the following, we describe the two stages of the resulting hybrids M-NSGA-II/PT) and ϵ-NSGA-II/PT.

5.1 First stage: Rough Approximation via Micro-NSGA-II

The aim of the MOEA to be implemented is to generate an approximated Pareto set that contains only a few solutions since the construction of the real front will be subsequently performed by the PT. However, these solutions should be diverse enough to identify all the components of a possibly disconnected front. Finally, the MOEA should be able to handle equality constraints efficiently. Therefore, the MOEA used in the first step should meet the following characteristics:

- The number of solutions in the roughly approximated set is small, in order to reduce the computational burden of the local search (PT). Indeed, in case of a completely connected front, one single approximated solution might allow to build the entire Pareto front.
- The MOEA should promote diversity, since the rough approximation produced should cover all the extent of the Pareto front and identify all the different components, in case of a disconnected front.
- The MOEA must be able to handle equality constraints. As mentioned before, a severely constrained problem might cause diversity issues that should be overcome by the MOEA.

Note that the two first features are conflicting, since the number of elements of the approximated front should be small enough to avoid unnecessary computations. Nonetheless, there must be enough approximated solutions to ensure the identification of all the components, in case of having a disconnected Pareto front. To deal with 2 or 3 objective problems such as those treated in this section, we observed that 20 points in the rough approximation represents a good trade-off between these two requirements. This means that the MOEA should work either with small populations or maintaining a small external archive.

Decomposition-based algorithms constitute a viable option. Actually, some preliminary experiments were performed with MOEA/D, but the use of small populations drastically increases the velocity of information transfer within the population. As a consequence, using too many neighbors (3 or more) led to very few different solutions in the final population. On the other hand, with too few neighbors, the algorithm is not able to converge to the real Pareto front. Finally, those convergence troubles encouraged the use of a dominance-based algorithm, NSGA-II, which can efficiently handle a two-objective problem and allows the easy integration of a constraint-handling mechanism.

In order to balance the cost of the two stages, the algorithm handles a small population (preliminary tests showed that 20 individuals allow a sufficiently good convergence) and the crowding distance operator is used to maintain diversity. Regarding constraint handling, the Constraint Dominance Principle (CDP) is combined with ϵ-constraint[Takahama and Sakai, 2006], to avoid the risk of premature convergence towards the first feasible solutions found by the algorithm. CDP implements the standard feasibility rules: if two solutions are infeasible, the one with the lower constraint violation is selected; if one solution is feasible and the other one is infeasible, the feasible one wins; finally, in case that both solutions are feasible, the decision is taken according to the dominance criterion.

In addition, according to the ϵ-constraint strategy, constraints are first relaxed at the beginning of the run, so that a solution x such that $\Phi(x) \leq \epsilon$ is considered as feasible (where $\Phi(x)$ represents the total amount of constraint violation of x). Then, ϵ is gradually reduced, having slightly infeasible solutions competing with feasible ones and allowing diversity preservation during the run. A decreasing schedule of ϵ was proposed in the framework of single-objective optimization in [Takahama and Sakai, 2006], where ϵ decreases according to a polynomial function until a critical generation T_c is reached. Then, ϵ is set to 0 and the constraint handling technique reduces to the above-mentioned CDP:

$$\epsilon = \begin{cases} \epsilon(0)\,(1{-}t/T_c)^{cp} & \text{if } 0 < t < T_c \\ 0 & \text{if } t \geq T_c, \end{cases} \tag{5.1}$$

where t is the generation number, cp is a parameter controlling the speed of the decrease and $\epsilon(0)$ is the constraint relaxation level at the first generation. This parameter is computed as the total constraint violation of x^θ, which is the θ-th solution in the first population, sorted in decreasing order of the total constraint violation Φ: $\epsilon(0) = \Phi(x^\theta)$.

Finally, an additional parameter was introduced in order to improve diversity: parent selection is performed with tournaments implementing the CDP extended with ϵ-constraint. However, the resulting winner of the tournament is considered only with probability p_f: in other cases (i.e., with probability $1 - p_f$), the winner individual is randomly chosen. The entire process is shortly described, for the reader convenience, in Algorithm 5.1.

5.2 Second stage: Refinement via PT

The main task at this stage is to appropriately process the resulting archive P provided by the MOEA. The main challenge here is to avoid unnecessary effort, e.g. via computing non-optimal KKT points along local Pareto fronts that are already dominated by previously computed solutions. While this is relatively easy for $k = 2$ objectives (see [Cuate et al., 2019]), this task becomes more complicated with higher values of k.

5.2.1 BOPs

PT could spend a lot of additional function evaluations computing non-optimal KKT points. We can solve this and other issues using PT for BOPs [Martín and Schütze, 2014] as described below. After the first stage, and before using PT, we need a post-processing procedure for archive P as in the following steps:

Algorithm 5.1 ϵ-NGSA-II

```
$P \leftarrow pop\_init()$
Evaluate each individual $x^i \in P$ to obtain $F(x^i)$ and $\phi(x^i)$
Compute $\epsilon(0)$ and set $\epsilon = \epsilon(0)$
for $t \leftarrow 1$ to $MaxGen$ do
    $P' \leftarrow crossover(P)$    ▷ Parent selection through tournament and $\epsilon$-constraint
    $P'' \leftarrow mutation(P')$
    $Q \leftarrow P \cup P''$
    $Q^F \leftarrow Feasible(Q, \epsilon)$, $Q^I = Infeasible(Q, \epsilon)$
    $Q^F \leftarrow FastNonDominatedSorting(Q^F)$
    $Q^I = SortConstraintViolation(Q^I)$
    Fill $P$ with $Q^F$, using crowding distance if necessary
    if $|Q^F| < PopSize$ then
        Complete $P$ with $Q^I$
    end if
    Update $\epsilon$ through equation 5.1
end for
$P \leftarrow reduce(P)$    ▷ return 20 solutions
Return $P$ and $F(P)$
```

1. Improve every point in P by the modification of the Newton method (2.56).

2. Remove all dominated points in the new archive (possible local fronts).

3. Sort the final archive P, according to f_1, in such a way that $f_1(p^{(1)}) < \ldots < f_1(p^{(m)})$, where $p^{(i)} \in P,\ i = 1, \ldots, m$.

Now, we can use PT as follows: we take the first element $p^{(1)} \in P$ as the starting point for PT with a **left-up** movement ($\mu^{(2)}$), and we compute as many solutions as possible (until there is no conflict with the other points of P). Then, we perform the **right-down** movement ($\mu^{(1)}$) starting at $p^{(1)}$, but from this case, we have to consider the value of the next element in P (and also the previous one in case we having) in order to avoid extra function evaluations.

In general, let x_d be the current solution for the **right-down** movement of PT starting from $p^{(i)},\ i = 1, \ldots, m-1$, τ as in Eq. (2.21), and $\theta \in (0, \tau)$; then we have the following stopping criteria:

- $\|F(x_d) - F(p^{(i+1)})\|_2 < \theta$. That is, we reach the next point in P. In case $x_d \prec p^{(i+1)}$, we delete $p^{(i+1)}$ from P and we continue with the right-down movement (compute a new x_d). Otherwise, we stop and select $p^{(i+1)}$ as a new starting point.

- $f_2(x_d) < f_2(p^{(i+1)})$. When the first condition is not satisfied, this condition means that $x_d \prec p^{(i+1)}$. If that is the case, we delete $p^{(i+1)}$ from P and continue with the **right-down** movement (compute a new x_d).
- No improvements in the f_2 direction could be achieved (PT's stopping condition). If the PT stops, then we select $p^{(i+1)}$ as a new starting point (we move to a different connected component of the Pareto front).

Additionally, for $p^{(i)}$, $i = 2, \ldots, m$, we have also to perform and verify the **left-up** movement. Here, we assume that a previous right-down movement was made. Let x_u be the current solution for the **left-up** movement of PT starting from $p^{(i)}$, $i = 2, \ldots, m$, x_d the last solution obtained by the right-down movement starting from $p^{(i-1)}$, τ as in Eq. (2.21), and $\theta \in (0, \tau)$; then we have the following stopping criteria:

- $\|F(x_u) - F(x_d)\|_2 < \theta$. This condition prevents the computation of extra solutions in previously considered regions of the Pareto front.
- $f_2(x_u) > f_2(x_d)$. When the first condition is not satisfied, this condition means that $x_d \prec x_u$. If that is the case, then we stop and we continue with the right-down movement for $p^{(i+1)}$.
- No improvements in the f_1 direction could be achieved (PT's stopping condition). If the PT stops, then we continue with the right-down movement for next point $p^{(i+1)}$.

5.2.2 General MOPs

The following procedure works for a general k. Before PT can be executed, the following post-processing has to be done on P:

1. Let τ be the desired minimal distance between two solutions in objective space. In this first step, go over P and eliminate elements that are too close to each other (if needed). This leads to the new archive $\tilde{P}$.
2. Apply the Newton method (2.56) to all elements of $\tilde{P}$. Remove all dominated solutions, and elements that are too close to each other as in the first step. This leads to the archive $\bar{P}$.
3. To obtain a "global picture" of the part of the Pareto front that will be computed by PT, construct a partition of a potentially interesting subset S of the image space into a set of hyper-cubes (or k-dimensional boxes) with radius $\approx \tau$. This partition can be easily constructed via using a binary tree whose root represents S (see [Dellnitz et al., 2005] for details, where, however, the partition is used

in decision variable space). S is a box that is constructed out of $\bar{P}$ as follows: denote by m_i and M_i the minimal and maximal value of the i-th objective value of all elements in $\bar{P}$, respectively. Then the i-th element of the center of S is set to $(m_i + M_i)/2$ and its i-th element of the radius to $(M_i - m_i)/2$. In the computations, we will only allow storing one candidate solution within each of these boxes in the archive A to guarantee a spread of the solutions.

Then, in the first step, the element $p^{(1)} \in \bar{P}$ is chosen as the starting point for PT, where $f_1(p^{(1)}) = m_1$. An external archive A will be created that will be the reference for PT and that will be the set of solutions that will be returned after the application of ϵ-NSGA-II/PT. At the beginning, it is $A := \{p^{(1)}\}$. In parallel, a box collection C will be created that contains all the (unique) boxes out of the above partition that contain all the elements of A. In the first step, C will set to the box that contains $p^{(1)}$. The application of PT leads to a sequence of candidate solutions x_i, $i = 1, \ldots, s$. For each x_i the following steps are performed:

1. If x_i is dominated by any element of A, then the current application of PT is stopped.
2. Else, it is checked if the unique box that contains x_i is already contained in C. If this is not the case, add this box to C and add x_i to A. Else, decline x_i and proceed with x_{i+1}.

After this, take the element from $\bar{P}\backslash\{p^{(1)}\}$ with the smallest value of f_2 as the starting point for PT and proceed as above. Proceed in this manner, sorting in a cyclic way according to each objective, until all elements $p \in \bar{P}$ have been chosen as starting points for PT.

5.3 Proposed Test Problems

Here, we propose two bi-objective test problems, Eq1-ZDT1 and Eq2-ZDT1, and one three-objective problem, Eq-Quad. All problems are scalable in the number of decision variables, and for all problems the inclusion of the equality constraint(s) has an influence on the location of the Pareto set.

5.3.1 Eq1-ZDT1

The original ZDT1 is a bi-objective problem with box constraints that can be defined for an arbitrary number n of decision variables. Meanwhile, the proposed

Eq1-ZDT1 problem is stated as follows

$$\begin{aligned} f_1(x) = &\ x_1 \\ f_2(x) = &\ g(x)\left(2 - \sqrt{\tfrac{f_1(x)}{g(x)}}\right) \end{aligned} \tag{5.2}$$

$$\text{s.t.} \quad h(x) = (x_1 - 0.5)^2 + (x_2 - 0.4)^2 - 0.25 = 0 \tag{5.3}$$

where

$$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i^2 \tag{5.4}$$

As we can see, the Eq1-ZDT1 (5.2) is also a scalable bi-objective problem in the number of variables, which changes the box constraints by an equality constraint (with the implicit inequality constraint that $x_1 \geq 0$ so that f_2 is defined). The constraint of this problem (5.3) defines a kind of "hyper-cylinder", where the variables x_1 and x_2 are placed on a circle, while the remaining variables $x_i,\ i = 3, \ldots, n$ can take any value.

In the following, we will provide the Pareto set for Eq1-ZDT1. For this, we need the set $P_h \subset Q$ defined as

$$\begin{aligned} P_h := \big\{ & x \in \mathbb{R}^n : (x_1 - 0.5)^2 + (x_2 - 0.4)^2 = 0.25, \\ & x_i = 0,\ i = 3, \ldots, n.\big\} \end{aligned} \tag{5.5}$$

Theorem 5.1. *Let P_h be defined as in* (5.5) *and $n = 30$. Then the subset $\mathcal{P}_{Eq1} \subset P_h$ given by*

$$\begin{aligned} & \mathcal{P}_{Eq1} = \{x \in \mathbb{R}^n : x_1 \in [0, \gamma], \\ & x_2 = 0.4 - \sqrt{0.25 - (x_1 - 0.5)^2},\ x_i = 0,\ i = 3, \ldots, n\}. \end{aligned} \tag{5.6}$$

where $\gamma \approx 0.977336$ is the Pareto set of Eq1-ZDT1.

Proof. a) First, we prove that $\nexists\, y \in Q \setminus P_h$ such that $y \prec x$, where $x \in P_h$.

Suppose that $\exists\, y \in \mathbb{R}^n \setminus P_h$ such that $h(x) = 0$ and $y \prec x, \forall x \in P_h$. First, let

$$x := (y_1, y_2, 0, \ldots, 0)^T, \tag{5.7}$$

with $(y_1 - 0.5)^2 + (y_2 - 0.4)^2 = 0.25$. Then, let $\Delta := (\Delta_1, \ldots, \Delta_n)^T \in \mathbb{R}^n \setminus \emptyset$ with $\Delta_1 = \Delta_2 = 0$; as $y \in \mathbb{R}^n \setminus P_h$ we can choose y as follows:

$$y := x + \Delta = (y_1, y_2, \Delta_3, \ldots, \Delta_n)^T, \tag{5.8}$$

Now, from (5.2) note that

1. For the first objective we have that $f_1(y) = f_1(x) = y_1$.
2. For the second objective we have that

$$\begin{aligned} g(x) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i^2 \\ &= 1 + \frac{9}{n-1}\left(y_2^2 + \sum_{i=3}^{n} 0\right) \\ &= 1 + \frac{9}{n-1} y_2^2, \\ g(y) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} y_i^2 \\ &= 1 + \frac{9}{n-1}\left(y_2^2 + \sum_{i=3}^{n} \Delta_i^2\right), \\ \Rightarrow g(x) &< g(y). \end{aligned}$$

Then,

$$\begin{aligned} \frac{y_1}{g(x)} > \frac{y_1}{g(y)} &\Rightarrow \sqrt{\frac{y_1}{g(x)}} > \sqrt{\frac{y_1}{g(y)}} \\ &\Rightarrow -\sqrt{\frac{y_1}{g(x)}} < -\sqrt{\frac{y_1}{g(y)}} \\ &\Rightarrow 2 - \sqrt{\frac{f_1(x)}{g(x)}} < 2 - \sqrt{\frac{f_1(y)}{g(y)}}. \end{aligned}$$

Finally,

$$\begin{aligned} g(x) < g(y) \Rightarrow \\ g(x)\left(2 - \sqrt{\frac{f_1(x)}{g(x)}}\right) < g(y)\left(2 - \sqrt{\frac{f_1(y)}{g(y)}}\right) \\ \Rightarrow f_2(x) < f_2(y), \end{aligned}$$

which contradicts the hypothesis. Thus $\nexists\, y \in P_D \setminus P_h \,|\, x \prec y$ with $x \in P_h$.

b) Now we show that the points $\mathcal{P}_{Eq1}$ are not dominated by each other and they dominate all the points in the set $P_h \setminus \mathcal{P}_{Eq1}$.

Let $x, x' \in P_h$ such that $x_1 = x'_1$ and $|x_2| < |x'_2|$. Notice that we can express x_2 in terms of x_1 as

$$x_2 = 0.4 \pm \sqrt{0.25 - (x_1 - 0.5)^2}, \tag{5.9}$$

and it is clear that the points of the form $(x_1, 0.4 + \sqrt{0.25 - (x_1 - 0.5)^2})$ are dominated by the points $(x_1, 0.4 - \sqrt{0.25 - (x_1 - 0.5)^2})$ (that is, the inferior half of the circle), then $g(x) < g(x') \Rightarrow f_2(x) < f_2(x')$.

Now, we can write $f_2(x)$ with $x \in P_{h_d}$ in terms of x_1 as

$$f_2(x_1) = \left(\frac{(n-1) + 9C^2(x_1)}{n-1}\right)\left(2 - \frac{x_1}{(n-1) + 9C^2(x_1)}\right). \tag{5.10}$$

where, $C(x_1) = 0.4 - \sqrt{0.25 - (x_1 - 0.5)^2}$.

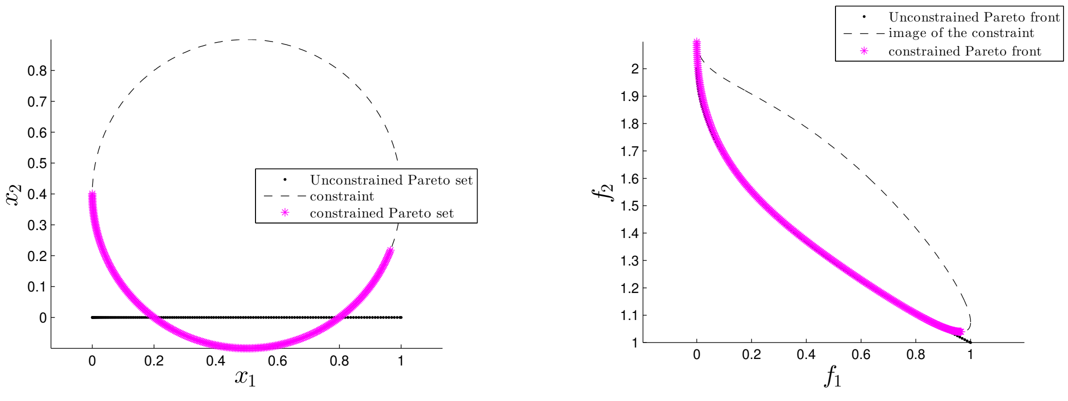
Computing the derivative of Eq. (5.10) we have

$$f_2'(x_1) = -\frac{(5(n-1) + 180x_1 - 90)\sqrt{0.25 - (x_1 - 0.5)^2} - 72x_1 + 36}{5(n-1)\sqrt{0.25 - (x - 0.5)^2}}. \tag{5.11}$$

The derivative of f_2 has only one root at $\gamma = x_1^* \approx 0.977336$. Also, note that for a point a we have that if $a \in [0, \gamma]$, then $f_2'(a) < 0$. On the other hand, if $a \in [\gamma, 1]$, then $f_2'(a) > 0$. Hence, $f_2(x)$ is monotonically decreasing, and consequently, points in $\mathcal{P}_{Eq1}$ are not dominated by each other.

Finally, by *(a)* and *(b)* we have that $\mathcal{P}_{Eq1}$ is the Pareto set of Eq1-ZDT1. □

To obtain γ for the formulation of the Pareto set we needed to consider f_2' which depends on n. The proof is analog for other values of n with changing value of γ. Some of these values can be found in Table 5.1.



(a) Projection of the Pareto set onto the x_1x_2-plane

(b) Pareto front

Figure 5.1: Pareto set and front of the Eq1-ZDT1 with $n = 30$.

5.3.2 Eq2-ZDT1

Via adding box constraints to Eq1-ZDT1, we can define the Eq2-ZDT1 problem as follows

$$\begin{aligned} f_1(x) = &\ \ x_1 \\ f_2(x) = &\ \ g(x)\left(2 - \sqrt{\tfrac{f_1(x)}{g(x)}}\right) \end{aligned} \tag{5.12}$$

$$\text{s.t.} \qquad h(x) = 0 \tag{5.13}$$

$$0 \leq x_i \leq 1,\ i = 1, \dots, n, \tag{5.14}$$

where $h(x)$ and $g(x)$ are defined as in (5.3) and (5.4), respectively.

Next, we will provide the analytical Pareto set for Eq2-ZDT1.

Theorem 5.2. *For $n = 30$, $x \in \mathbb{R}^n$, the Pareto set for the Eq2-ZDT1 problem (see Figure 5.2) is given by*

$$\begin{aligned} \mathcal{P}_{Eq2} := \{x \in \mathbb{R}^n : x_1 \in \mathcal{I}_1 \cup \mathcal{I}_2 \cup \mathcal{I}_3,\ x_2 = I(x_1), \\ x_i = 0,\ i = 3, \dots, n.\} \end{aligned} \tag{5.15}$$

where $\mathcal{I}_1 := [0, 0.2]$, $\mathcal{I}_2 := [\eta, 0.8)$, $\mathcal{I}_3 := [0.8, \gamma]$, $\eta \approx 6.700214$, *and*

$$I(x_1) := \begin{cases} 0.4 - \sqrt{0.25 - (x_1 - 0.5)^2}, & x_1 \in \mathcal{I}_1 \cup \mathcal{I}_3 \\ 0.4 + \sqrt{0.25 - (x_1 - 0.5)^2}, & x_1 \in \mathcal{I}_2 \end{cases} \tag{5.16}$$

Proof. a) Let P_h be defined as in (5.5), and P_{Eq1} as in (5.6) and then first part of this proof is analogs the previous analysis for Eq1-ZDT1.

b) As second step, we need to remove all the points in P_{Eq1} that do not satisfy the box constraints. In particular, as $x_i = 0,\ i = 3, \dots, n$, we focus on x_1 and x_2. For $x_1, x_2 \in P_h$, we have that $x_1 \in [0, 1]$ and $x_2 \in [-0.1, 0.4]$, i.e., some values of x_2 do not satisfy the lower bound.

We can express x_1 as follows:

$$x_1 = 0.5 + \sqrt{0.25 - (x_2 - 0.4)^2}, \tag{5.17}$$

thus, for $x_2 = 0$ we can find the values of x_1 that define $\mathcal{I}_1$ and $\mathcal{I}_3$ via:

$$x_1 = 0.5 \pm 0.3 \Rightarrow \mathcal{I}_1 = [0, 0.2]\ \mathcal{I}_3 = [0.8, \gamma]$$

After removing the non-feasible points from P_{Eq1} we have a gap in Pareto set/front. Now, notice that, some points $x \in P_h : x_2 = 0.4 + \sqrt{0.25 - (x_1 - 0.5)^2}$ (that is, $x \notin P_{Eq1}$), could be within the gap. That is, we have to find the values of $x \in P_h \setminus P_{Eq1}$ such that $f_2(x_1) \in [f(0.8), f(0.2)]$.

For this we consider:

$$\bar{f}_2(x_1) = \left(\frac{(n-1)+9C_2^2(x_1)}{n-1}\right)\left(2 - \frac{x_1}{(n-1)+9C_2^2(x_1)}\right). \qquad (5.18)$$

where, $C_2(x_1) = 0.4 + \sqrt{0.25-(x_1-0.5)^2}$.

Notice that $[f_2(0.8), f_2(0.2)] \subset [\bar{f}_2(0.8), \bar{f}_2(0.2)]$ and $\bar{f}_2$ is a continuous function, then for the intermediate value theorem $\exists\, x_{1,0.8}, x_{1,0.2}$ such that $\bar{f}_2(x_{1,0.2}) = f_2(0.2)$ and $\bar{f}_2(x_{1,0.8}) = f_2(0.8)$, respectively.

For $n = 30$, such values are $x_{1,0.8} \approx 0.9773356$ and $x_{1,0.2} \approx 0.670021$. Then $\mathcal{I}_2 = [\eta, 0.8)$, with $\eta = x_{1,0.2}$, as $x_{1,0.8} > 0.8$ and $f_2(0.8) < \bar{f}_2(0.8)$.

Finally, by *(a)* and *(b)* we have that $\mathcal{P}_{Eq2}$ is the Pareto set of Eq2-ZDT2. □

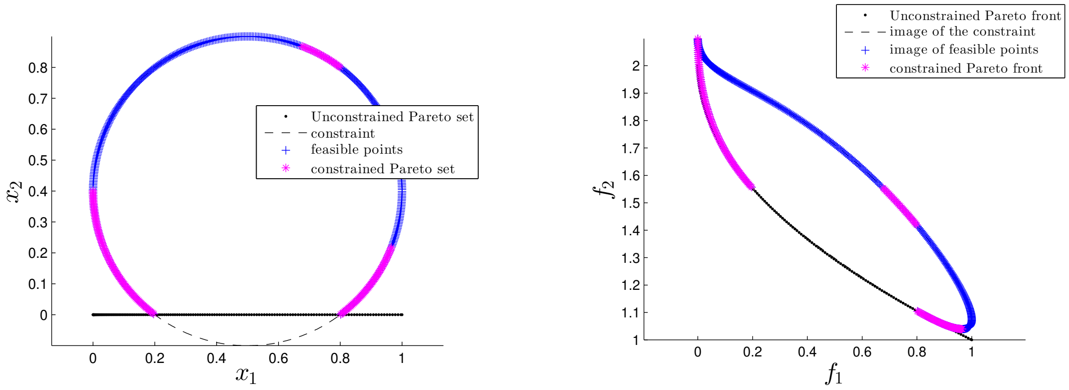
As we can observe, the values of γ and η depend on n (see Theorems 5.1 and 5.2). We refer to Table 5.1 for these values for other dimensions of the decision variable space. See Figure 5.2 for Pareto set and front of Eq2-ZDT2.

Table 5.1: Values of γ and η for different dimensions n of the decision variable space. γ and η are used to describe the Pareto sets of Eq1-ZDT1 and Eq2-ZDT1.

n	γ	η
16	0.954380	0.863336
17	0.957029	0.848048
18	0.959445	0.832853
19	0.961656	0.817805
20	0.963686	0.802946
21	0.965554	0.788312
22	0.967278	0.773932
23	0.968874	0.759830
24	0.970353	0.746025
25	0.971727	0.732530
26	0.973006	0.719359
27	0.974199	0.706518
28	0.975314	0.694012
29	0.976357	0.681847
30	0.977336	0.670021
31	0.978253	0.658536
32	0.979116	0.647389

5.3.3 Eq-Quad

Finally, we propose a modification of the problem taken from [Martín and Schütze, 2018] which has three quadratic objectives and two equality constraints:



(a) Projection of the Pareto set onto the x_1x_2-plane

(b) Pareto front

Figure 5.2: Pareto set and front of the Eq2-ZDT1 with $n = 30$.

$$f_j(x) = \|x - a^{(j)}\|_2^2, \quad j = 1, \dots, k,$$

where $x \in \mathbb{R}^n$, $k = 3$, and $a^{(1)} = (1, -1.4, -0.4)^T$, $a^{(2)} = (-1.4, 1, -0.4)^T$, and $a^{(3)} = (0.4, 0.4, 0.8)^T$.

subject to

$$\begin{aligned} h1(x) &= r^2 - x_3^2 - \left(R - \sqrt{x_1^2 + x_2^2}\right)^2 = 0 \\ h2(x) &= x_1 + x_2 - x_3 = 0 \\ &\quad -1.5 \leq x_1 \leq 1 \\ &\quad -1.5 \leq x_2 \leq 1 \\ &\quad 0 \leq x_3 \leq 1. \end{aligned}$$

Figure 5.3 shows the constraints and the Pareto set for $n = 3$. As can be seen, the Pareto set consists of two connected components that can be both expressed by curves (and which are hence 1-dimensional).

5.4 Numerical Results

Here, we present some numerical results that compare the behavior of some state-of-the-art MOEAs against the proposed ϵ-NSGA-II/PT. As test functions we have chosen to take the three test problems proposed above, Eq1-Quad from [Martín and Schütze, 2018], the CZDT test suite as well as a modification of a problem from Das and Dennis problem (D&D) stated in [Martín and Schütze, 2018]. A point x is considered to be feasible if $\|h(x)\| < \varepsilon$, where we have taken $\varepsilon = 1e - 04$, which is common in evolutionary computation.

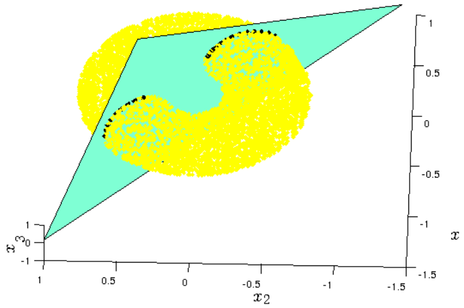


Figure 5.3: Constraints and feasible region for the Torus problem

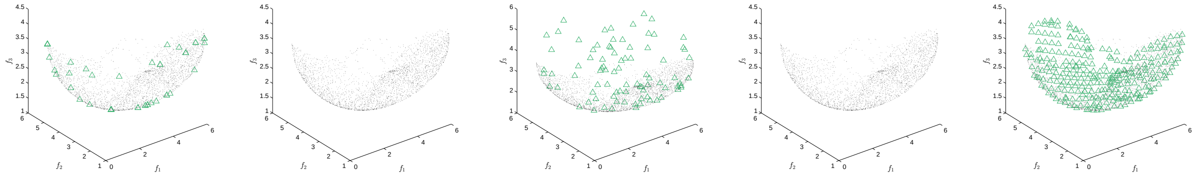
Our experiments have shown that the new hybrid needs between $15,000$ and $17,000$ function evaluations (FEs) to obtain good results for the bi-objective problems and $110,000$ to $140,000$ FEs for the three-objective problems. In order to make the comparison fair, we have set a final budget of $20,000$ FEs for all the selected MOEAs on the bi-objective problems and $150,000$ FEs for the three-objective problems. For ϵ-NSGA-II/PT, we have split the budget for the bi-objective problems into $15,000$ FEs for ϵ-NSGA-II and the remaining $5,000$ FEs for PT (and $100,000$ plus $50,000$ FEs for three-objective problems). For the realization of PE, we have used Automatic Differentiation to compute the gradients, which allows to express the cost of the continuation method in terms of FEs. For all experiments, we have executed 30 independent runs. We performed the Wilcoxon Test for the statistical significance to validate the results. For this, we consider the value $\alpha = 0.05$. Based on the test results, for the comparison between ϵ-NSGA-II/PT and any of the chosen MOEAs the symbol "$\uparrow$" means that the obtained results are statistically significant. The symbol "$--$" means that no Δ_2 value could be computed for the algorithm. This was the case if a MOEA detected no feasible solutions for at least 25 runs.

Figures 5.4-5.5 show the results for Eq1-Quad and Eq2-Quad, respectively; while, the figures for the bi-objective problems are shown in Figures 5.6-5.9. The theoretical PF is marked with dots (.), while approximations from the algorithms are marked with triangles ($\triangle$). The Pareto fronts for the other test problems have been omitted due to space limitations, however, Table 5.2 shows the indicator values, Δ_2 and I_F, for all test problems. Smaller values of Δ_2 correspond to better qualities of the approximated solution, while larger values for I_F indicate more feasible solutions in the approximation. The best values are displayed in boldface for each problem and each indicator. We can see that the new hybrid algorithm significantly outperforms the other algorithms in nine out of the ten test functions. In some cases, the MOEAs are not able to find any feasible solution within the given FE budget. ϵ-NSGA-II/PT, however, loses against c-MOEA/D on Eq2-Quad. This due to the fact that the real Pareto front is disconnected, and in most of the runs the first stage of our proposal was not able to find adequate solutions near both components. Therefore, in the

Table 5.2: Values of Δ_2 and I_F on the selected test problems.

	Method	Δ_2	I_F		Method	Δ_p	F. Ratio
CZDT1	ϵ-NSGA-II/PT	**0.0038**	**1.0000**	**D & D**	ϵ-NSGA-II/PT	**0.3442**	**1.0000**
	(std.dev)	(0.0002)			(std.dev)	(0.6553)	
	c-MOEA/D	–	0.0000		c-MOEA/D ↑	4.5168	0.0270
	(std.dev)	(–)			(std.dev)	(2.1485)	
	e-MOEA/D	–	0.0000		e-MOEA/D	–	0.0000
	(std.dev)	(–)			(std.dev)	(–)	
	GDE3	–	0.0000		GDE3	–	0.0000
	(std.dev)	(–)			(std.dev)	(–)	
	NSGA-II	–	0.0000		NSGA-II	–	0.0000
	(std.dev)	(–)			(std.dev)	(–)	
CZDT2	ϵ-NSGA-II/PT	**0.0038**	**1.0000**	**Eq1-ZDT1**	ϵ-NSGA-II/PT	**0.0158**	**1.0000**
	(std.dev)	(0.0002)			(std.dev)	(0.0015)	
	c-MOEA/D	–	0.0000		c-MOEA/D ↑	0.4088	0.5060
	(std.dev)	(–)			(std.dev)	(0.2504)	
	e-MOEA/D	–	0.0000		e-MOEA/D ↑	0.1683	0.3787
	(std.dev)	(–)			(std.dev)	(0.0488)	
	GDE3	–	0.0000		GDE3 ↑	3.0997	0.6653
	(std.dev)	(–)			(std.dev)	(0.5521)	
	NSGA-II	–	0.0000		NSGA-II ↑	–	0.0013
	(std.dev)	(–)			(std.dev)	(–)	
CZDT3	ϵ-NSGA-II/PT	**0.0156**	**1.0000**	**Eq2-ZDT1**	ϵ-NSGA-II/PT	**0.1251**	**1.0000**
	(std.dev)	(0.0164)			(std.dev)	(0.0428)	
	c-MOEA/D	–	0.0000		c-MOEA/D ↑	0.6624	0.4700
	(std.dev)	(–)			(std.dev)	(0.2215)	
	e-MOEA/D	–	0.0000		e-MOEA/D ↑	0.7800	0.4617
	(std.dev)	(–)			(std.dev)	(0.1235)	
	GDE3	–	0.0000		GDE3 ↑	3.6144	0.8873
	(std.dev)	(–)			(std.dev)	(0.5234)	
	NSGA-II	–	0.0000		NSGA-II ↑	2.4662	0.0037
	(std.dev)	(–)			(std.dev)	(1.6368)	
CZDT4	ϵ-NSGA-II/PT	**0.0031**	**1.0000**	**Eq1-Quad**	ϵ-NSGA-II/PT	**0.1261**	**1.0000**
	(std.dev)	(0.0016)			(std.dev)	(0.0043)	
	c-MOEA/D	–	0.0000		c-MOEA/D ↑	0.5714	0.2533
	(std.dev)	(–)			(std.dev)	(0.0953)	
	e-MOEA/D	–	0.0000		e-MOEA/D ↑	3.1760	0.0014
	(std.dev)	(–)			(std.dev)	(0.6012)	
	GDE3	–	0.0000		GDE3 ↑	0.9133	0.2666
	(std.dev)	(–)			(std.dev)	(0.0931)	
	NSGA-II	–	0.0000		NSGA-II	–	0.0001
	(std.dev)	(–)			(std.dev)	(–)	
CZDT6	ϵ-NSGA-II/PT	**0.0884**	**1.0000**	**Eq2-Quad**	ϵ-NSGA-II/PT	1.9969	**1.0000**
	(std.dev)	(0.0180)			(std.dev)	(1.0378)	
	c-MOEA/D	–	0.0000		c-MOEA/D ↑	**0.4737**	0.1583
	(std.dev)	(–)			(std.dev)	(0.2000)	
	e-MOEA/D	–	0.0000		e-MOEA/D	–	0.0000
	(std.dev)	(–)			(std.dev)	(–)	
	GDE3	–	0.0000		GDE3 ↑	2.8142	0.0047
	(std.dev)	(–)			(std.dev)	(1.1008)	
	NSGA-II	–	0.0000		NSGA-II	–	0.0000
	(std.dev)	(–)			(std.dev)	(–)	

second stage, we were in most cases only able to compute one of the two components. However, note that all the solutions of the final approximation are feasible in all the independent runs.



(a) cMOEAD

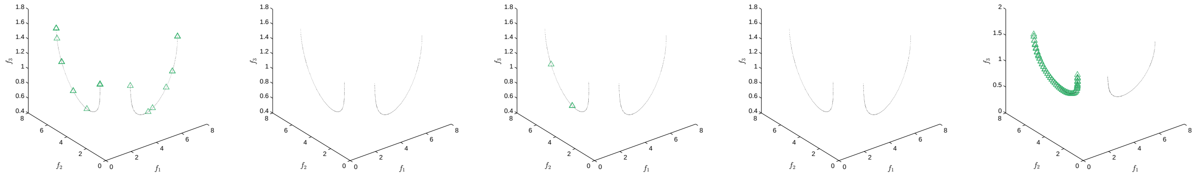
(b) eMOEAD

(c) GDE3

(d) NSGA2

(e) ϵ-NSGAII/PT

Figure 5.4: Pareto front approximations on the Eq1-Quad for the selected MOEAs using a budget of $150,000$ function evaluations.



(a) cMOEAD

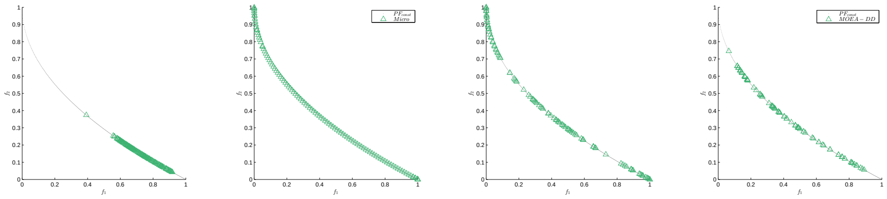
(b) eMOEAD

(c) GDE3

(d) NSGA2

(e) ϵ-NSGAII/PT

Figure 5.5: Pareto front approximations on the Eq2-Quad for the selected MOEAs using a budget of $150,000$ function evaluations.



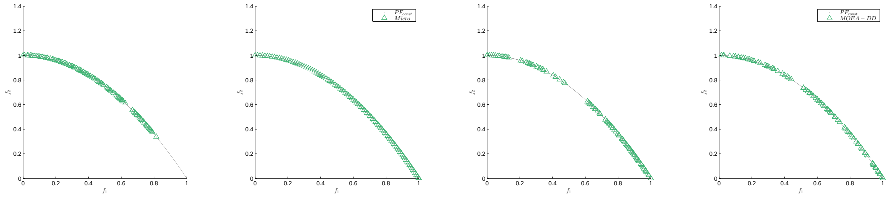
(a) NSGA-II

(b) M-NSGA-II/PT

(c) NSGA-II$_MPP$

(d) MOEA/D/D

Figure 5.6: Pareto front approximations on the C-ZDT1 problem for the selected MOEAs with $20,000$ function evaluations.



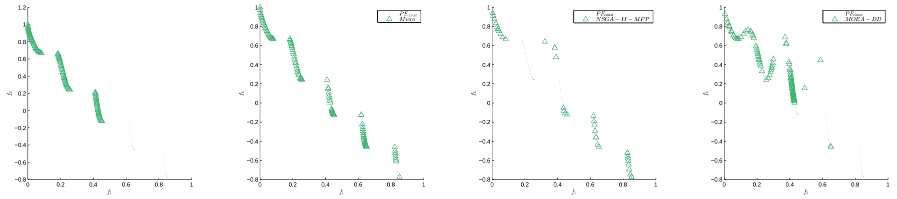
(a) NSGA-II

(b) M-NSGA-II/PT

(c) NSGA-II$_{M}PP$

(d) MOEA/D/D

Figure 5.7: Pareto front approximations on the C-ZDT2 problem for the selected MOEAs with $20,000$ function evaluations.



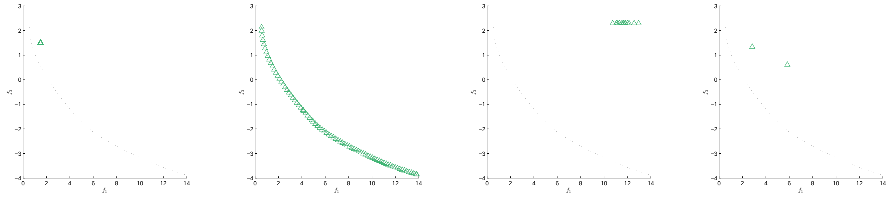
(a) NSGA-II

(b) M-NSGA-II/PT

(c) M-NSGA-II$_{M}PP$

(d) MOEA/D/D

Figure 5.8: Pareto front approximations on the C-ZDT3 problem for the selected MOEAs with $20,000$ function evaluations.



(a) NSGA-II

(b) M-NSGA-II/PT

(c) M-NSGA-II$_{M}PP$

(d) MOEA/D/D

Figure 5.9: Pareto front approximations on the D&D problem for the selected MOEAs with $20,000$ function evaluations.

Chapter 6

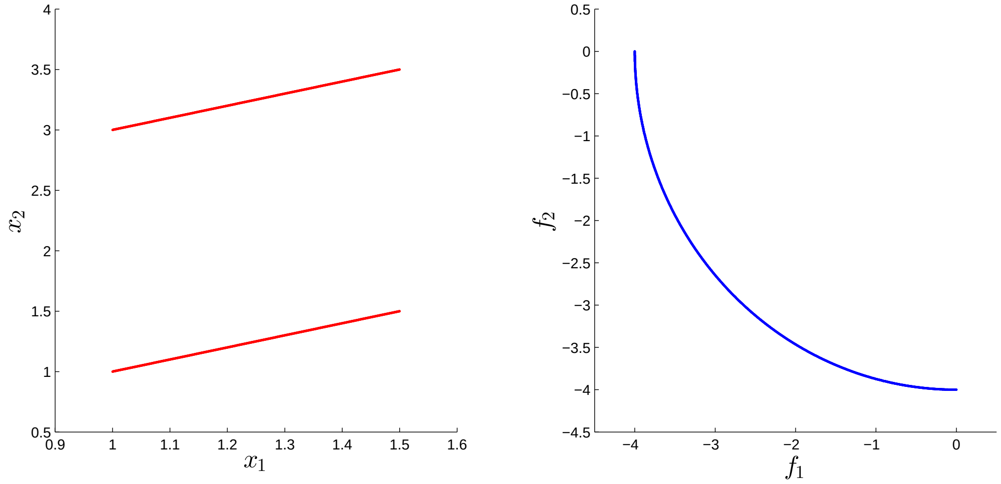
Exploration in Decision Space

State-of-the-art MOEAs that measure the approximation quality of their outcome entirely in objective space work typically well if there is a 1:1 relationship between Pareto set and Pareto front (that is, if for every $y \in F(P_D)$ there exists exactly one $x \in P_D$ such that $F(x) = y$). That is, if a good[1] finite size approximation of the Pareto front is found by the MOEA, the corresponding finite size approximation of the Pareto set is, in many cases, also satisfying. This, however, does not hold any more if there is an $m : 1$ relationship between Pareto set and front (i.e., if there are multiple $x_i \in P_D$ such that $F(x_i) = y$ for a $y \in F(P_D)$). If, for instance, there are several connected components of the Pareto set that map to the same part of the Pareto front, a good Pareto front approximation does not imply a good (or at least satisfying) approximation of the Pareto set. To see this, consider the hypothetical bi-objective problem that is shown in Figure 6.1. The Pareto set of this problem consists of two disjunct connected components that map both to the same Pareto front (that is, every $y \in F(P_D)$ has exactly two pre-images). Figure 6.2 shows four possible approximations in decision and objective space. As can be seen, the approximation quality is very high for all sets in objective space, while this is not the case for the Pareto set approximations. Out of them, only the last one is "complete" according to the given discretization. MOEAs that merely measure their outcomes in objective space cannot distinguish between those solutions, and consequently, the Pareto set approximation is left to chance. MOPs of this kind are termed Type III problems in [Rudolph et al., 2007].

To overcome this problem, we propose to develop a density estimator that aims to obtain a good distribution, both in objective and decision space. Usually, a classical density estimator groups the population considering only the objective values. Such classification is commonly used to define selection criteria for its elements, giving them certain **reference value** based on its distribution in objective space. According to

[1] The goodness can be measured e.g. by any existing performance indicator.

the design of each algorithm, the individual with either lower or higher reference value is chosen. The idea is to define a relationship between this reference value in objective space and a certain measurement in decision space. In this way, the first grouping phase identifies promising solutions in objective space; meanwhile, the second phase favors solutions with the most different values in the decision space. Our goal is to properly represent the trade-off between these two aspects.



(a) Pareto Set **(b)** Pareto Front

Figure 6.1: Pareto set and Pareto front of a hypothetical bi-objective problem where the Pareto set consists of two connected components that both map to the entire Pareto front.

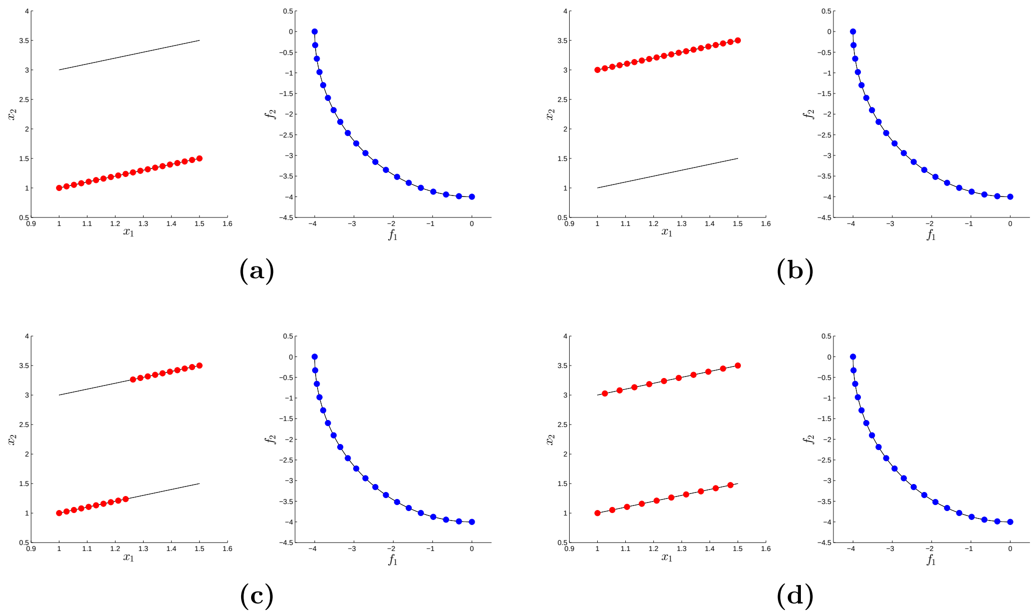
6.1 Proposed Framework

In the following, we describe the Variation Rate, a heuristic that can preserve diversity in decision space. The obtained points can be used in a next step by the PT or PE methods for a reconstruction of all the optimal points or for an exploration in decision space, respectively.

6.1.1 Using the Averaged Distance in Decision Variable Space

Here, we discuss why we think that the usage of the averaged distance is an adequate measure in decision space that serves our purpose.

Let $I = \{x_1, \ldots, x_s\} \subset \mathbb{R}$ be a finite set, then the averaged distance $\bar{d}$ between



(a)

(b)

(c)

(d)

Figure 6.2: Four different Pareto set/front approximations, where all Pareto front approximations are good (e.g, in the Hausdorff sense), but where only in case (d) the Pareto set approximation is complete.

each element $x_i \in I$ and the rest of the elements in I is given by

$$\bar{d}(x_i, I) = \frac{1}{s-1} \sum_{\substack{j=1 \\ j \neq i}}^{s} d(x_i, x_j), \tag{6.1}$$

where $d(x_i, x_j)$ is the desired metric for the distance between the two elements x_i and x_j in the decision space, which can vary according to the encoding or the adopted norm. In this work, we consider the Euclidean distance, i.e., $d(x_i, x_j) = \|x_i - x_j\|_2$. Though the averaged distance is defined for every finite set $I \in \mathbb{R}^n$, we will apply it on sets where the values of their images, $F(x_i),\ i = 1, \ldots, s$, are close to each other.

As an illustrative example, let's consider again the Type III bi-objective problem whose Pareto set and front are shown in Figure 6.3. The set I is given by the three points $\blacktriangledown$, $\blacklozenge$, and $\bigstar$. All three images are relatively close to the given reference point Z. Let's assume for simplicity that the distances of all three images to Z are given by one (i.e., $\|Z - F(\blacktriangledown)\| = \|Z - F(\blacklozenge)\| = \|Z - F(\bigstar)\| = 1$). Further, we assume that for the distances in variable space we have $d(\blacklozenge, \bigstar) = 0.4$, $d(\blacklozenge, \blacktriangledown) = 2$, and $d(\blacktriangledown, \bigstar) = 2.2$. Then, we obtain

$$\bar{d}(\bigstar, I) = \frac{2.6}{2} = 1.3, \qquad \bar{d}(\blacklozenge, I) = \frac{2.4}{2} = 1.2, \qquad \text{and} \qquad \bar{d}(\blacktriangledown, I) = \frac{4.2}{2} = 2.1$$

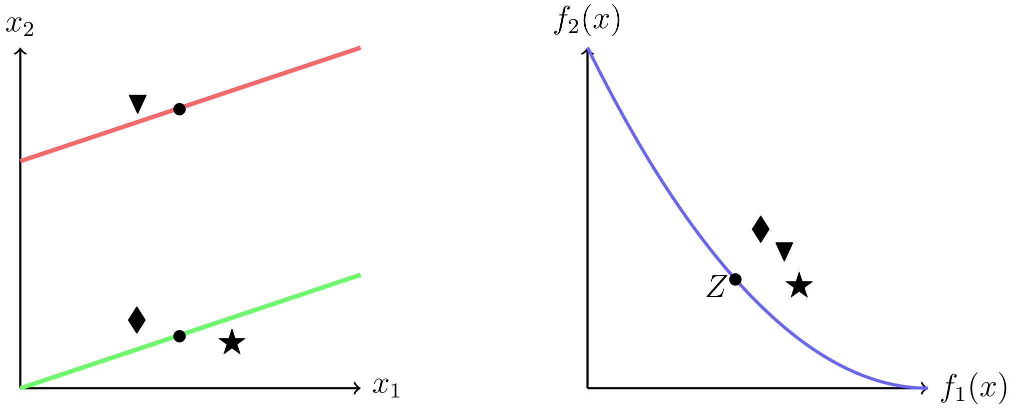


Figure 6.3: Illustrative example of the VR.

Notice that the point with the *biggest* average distance in variable space is also the most different individual in this space. In other words, elements with the maximum average distance in decision space have the desired behavior for Type III problems.

However, is it not sufficient only to take into account the average distance of the variables as a selection criteria. Our problem now is how to select an individual that has both a good quality in objective space as a good distribution in variable space. We discuss this issue in the following.

6.1.2 Variation Rate

As explained before, the generic selection criterion of most MOEAs prefers individuals with "the best" reference value in objective space, and this could be a maximum or a minimum value according to the selection procedure. For instance, the selection criterion of the NSGA-II prefers individuals with the biggest crowding distance, while the niching procedure of the NSGA-III favors individuals with the least distance to an induced line. In our case, we have to consider both the reference value provided by the classical selection criterion, as well as the average distance in decision variable space to solve Type III problems.

We first consider selection mechanisms that prefer small reference values. For this, let I be a set of points in decision space whose images are close to each other, let v_i be the reference value for each $x_i \in I$ and let $\bar{d}(x_i, I)$ be defined as in (6.1). Then the variation rate r_i for each element x_i is stated as follows:

$$r_i = \frac{v_i}{\bar{d}(x_i, I)} \tag{6.2}$$

This makes sense as for Type III problems the elements of the neighborhood I will have a similar reference value in objective space, while the average distances will be larger for the most different solutions in decision space. Hence, its quotient (the variation rate) will tend to be smaller than for the rest of the quotients. Thus, through the variation rate, we have a way to relate the objective and the decision spaces in order to choose the best individual in each group.

Next, we address selection mechanisms that prefer large reference values. For this, we have two options. The first one is to edit the selection criterion to prefer small values in order to use the variation rate. The second alternative is to use a product instead of a quotient. We decided to conserve the essence of each MOEA, and therefore, we implemented the second option in this work, to what we call *the inverse variation rate*. More precisely, for a set I as above with reference values v_i for each $x_i \in I$ and $\bar{d}(x_i, I)$ as defined as in (6.1), the inverse variation rate $\tilde{r}_i$ is defined as follows:

$$\tilde{r}_i = v_i \cdot \bar{d}(x_i, I), \tag{6.3}$$

Using this definition, elements with the largest values are hence preferred.

In order to illustrate these two approaches, we go back to the example shown in Figure 6.3 where we already have the values of $\bar{d}$ and v_i for all the three elements of the set I. Suppose that we have to select two out of the three individuals $\bigstar$, $\blacktriangledown$, and $\blacklozenge$.

If we work with a MOEA which has a selection criterion that prefers individuals with the least distance to Z in objective space, then the only option we have is randomly choosing them (since $v_\blacktriangledown = \|Z - F(\blacktriangledown)\| = 1$, $v_\blacklozenge = \|Z - F(\blacklozenge)\| = 1$, and $v_\bigstar = \|Z - F(\bigstar)\| = 1$). However, if we use the variation rate, then the values change to:

$$r_\bigstar = \frac{1}{1.3} \approx 0.7692 \qquad r_\blacklozenge = \frac{1}{1.2} \approx 0.8333, \qquad \text{and} \qquad r_\blacktriangledown = \frac{1}{2.1} \approx 0.4762,$$

In this way, we select $\blacktriangledown$ as the preferred solution and the second one is $\bigstar$, which preserves individuals in both of the disconnected regions in decision space.
The desired two-element population is hence given by

$$P = \{\blacktriangledown,\ \bigstar\}$$

Otherwise, if we work with a MOEA which has a selection criterion that prefers individuals with the largest distance to Z in objective space (maybe in order to preserve diversity), then the option we have again is randomly choosing two of them. However, if we use the inverse variation rate, that is:

$$\tilde{r}_\bigstar = 1 \cdot 1.3 = 1.3, \qquad \tilde{r}_\blacklozenge = 1 \cdot 1.2 = 1.2, \qquad \text{and} \qquad \tilde{r}_\blacktriangledown = 1 \cdot 2.1 = 2.1,$$

then it also leads ▼ and ★ as the selected individuals.

The desired two-element population is again given by

$$P = \{▼,\ ★\}$$

Observe that in both cases we conserve one individual in each disconnected component of the Pareto set. It is something that we can not guarantee with the use of a standard approach.

Notice that for all MOPs with a 1:1 relationship of the Pareto set and the Pareto front it is expected that solutions in the same neighborhood have similar reference values in objective space and also a similar average distance in decision space. Thus, it is also likely that making the quotient or the product of these values does not significantly affect the original selection criterion. This will be shown in Section 4 on several classical benchmark problems.

We can now state a general framework. A pseudocode of the Variation Rate is shown in Algorithm 6.1.

Algorithm 6.1 Framework to include the Average Distance in Decision Variable Space within any MOEA

```
Require: Parameters of the selected MOEA
Ensure: Final population $P_t$
$t \leftarrow 0$
$P_0 \leftarrow$ InitializePopulation()
while the stop criterion is not satisfied do
    $M_t \leftarrow$ VariationOperator($P_t$)
    $V_t \leftarrow$ SelectProcedure($P_t \cup M_t$)
    $P_t \leftarrow$ SelectByVariationRate($V_t$)
    $t \leftarrow t + 1$
end while
```

In Algorithm 6.1, the procedure `SelectByVariationRate` takes the reference values V_t in objective space provided by `SelectProcedure` (we assume it is based on a classical selection criterion), and then it updates such values according to the variation rate or the inverse variation rate in order to improve the selection mechanism to deal with Type III problems.

As we can see, this framework can be used in principle within any MOEA, however, the particular use of the variation rate or the inverse variation rate will depend on the given MOEA. In the following, we explain how to adapt the Variation Rate for four of the most representative MOEAs.

6.1.3 Integration into NSGA-II

The first algorithm that we consider is the classical NSGA-II, which has been used successfully for the treatment of a large number of applications. This is a domination-based multi-objective evolutionary algorithm; that is, this method directly applies the Pareto dominance relation and an elitism strategy to preserve the best individuals along the optimization process. The elitism operator is incorporated via a special parent selection based on two mechanisms: **fast-non-dominated-sorting** and **crowding distance**. The former conserves best individuals based on the Pareto dominance relation, whereas the latter is used to promote the preservation of the diversity.

We consider the classification in fronts performed by the non-dominated-sorting as our neighborhood structure because the crowding distance is applied only in the last front that can contribute elements to the next population. This means that we have to integrate the diversity in decision space into the crowding distance. The crowding distance procedure sorts the elements in the last front according to the values of an objective, then the crowding distance of an individual p_i is the average distance in objective space from the previous and the next individuals (according to the induced order), that is, the individuals p_{i-1} and the p_{i+1}. In order to preserve the extreme individuals, the crowding distance of the first and the last element is set as a big value. This means that the crowding distance prefers elements with big values, and hence, we use the inverse variation rate.

The pseudocode of the modification of the NSGA-II with the variation rate (VR-NSGA-II) is shown in Algorithm 6.2. The differences between the NSGA-II and the VR-NSGA-II are line 12 and 20. In line 12, we compute the inverse variation rate values of the last front, while, in line 20, we perform the selection according to these values.

6.1.4 Integration into NSGA-III

We consider this algorithm here because it is able to properly deal with MOPs with many objectives. This algorithm is similar to its predecessor, the NSGA-II in the variation operators and in the classification of the fronts via the non-dominated-sorting; however, the crowding distance is replaced by a more sophisticated procedure.

Here, the idea is to take advantage of the **association** method of the NSGA-III, which defines a “neighborhood” structure in a very convenient way for our purpose. The association method assigns each element of F_j (the last front by classifying after the non-dominated-sorting) to the nearest induced line by some weight $w_i \in Z$, where Z is a set of reference points. Each weight can have more than one associated element, forming a neighborhood.

Algorithm 6.2 Pseudocode of VR-NSGA-II

```
Require: Population size ($P_s$), crossover probability ($P_c$), mutation probability ($P_m$)
Ensure: Final Population
Population ← InitializePopulation($P_s$)
FastNondominatedSorting(Population)
Selected ← SelectParentsByRank(Population, $P_s$)
Children ← CrossoverAndMutation(Selected, $P_c$, $P_m$)
while ¬StopCondition() do
    Union ← Merge(Population, Children)
    Fronts ← FastNondominatedSorting(Union)
    Parents ← $\emptyset$
    $Front_L$ ← $\emptyset$
    for $Front_i$ ∈ Fronts do
        $V_t$ ←CrowdingDistanceAssignment($Front_i$)
        $R_t$ ←InverseVariationRateAssigment($V_t$)
        if Size(Parents)+Size($Front_i$) > $P_s$ then
            $Front_L$ ← $i$
            break
        else
            Parents ← Merge(Parents, $Front_i$)
        end if
        if Size(Parents)¡$P_s$ then
            $Front_L$ ← SortByRankAndInverseVariationRate($Front_L$)
            for $P_1$ to $P_{P_s - \text{Size}(Front_L)}$ do
                Parents ← $Pi$
            end for
        end if
        Selected ← SelectParentsByRankAndDistance(Parents, $P_s$)
        Population ← Children
        Children ← CrossoverAndMutation(Selected, $P_c$, $P_m$)
    end for
end while
return Children
```

In the original NSGA-III, **niching** is realized by sorting the obtained groups in the association stage according to its cardinality in ascending order. The element with the least distance to the induced line in each group is selected, and the algorithm continues with the next group until the population is filled. Thus, we modify the **niching** method. To include the diversity in decision space, our new niching procedure does not prefer the element with the least distance value. Instead, it prefers the one with the smallest *variation rate*.

The pseudocode of an iteration of the VR-NSGA-III algorithm with variation rate is shown in Algorithm 6.3. We only show the iteration as the complete code is basically identical to NSGA-II with a different selection mechanism. Here, the main difference of the variation rate version is in line 14; while the original NSGA-III uses ninching as part of its selection criterion, the VR-NSGA-III employs the variation rate, as it is stated in line 14.

Algorithm 6.3 Iteration of the VR-NSGA-III

```
Require: Reference points $Z$, current population $P_t$
Ensure: Next population $P_{t+1}$
 1: $S_t = \emptyset,\ i = 1$
 2: $Q_t$ = apply variation operators to $P_t$
 3: $M_t = P_t \cup Q_t$
 4: $(F_1, F_2 \ldots, )$ = non-dominated-sorting($M_t$)
 5: while $|S_t| \leq N$ do
 6:     $t = S_t \cup F_i$
 7:     $i = i + 1$
 8: end while
 9: Add first fronts to $P_{t+1}$
10: $F_i$ := last added front
11: Normalize $F_i$
12: Associate elements of $F_i$ with each $Z$
13: $V_t$ := Niching of $F_i$
14: $R_t$ := SelectByVariationRate($V_t$)
15: $P_{t+1} : S_t \cup R_t$
```

6.1.5 Integration into MOEA/D

MOEA/D is part of the Decomposition-Based Evolutionary Algorithms, which transform the original multi-objective optimization problem into a set of single-objective optimization problems that are simultaneously solved. In particular, this method takes a set of weights to define neighborhoods. The set of nearest weights defines one neighborhood and the best individuals are selected based on the value of a certain aggregative function. MOEA/D considers the weighted aggregation of objectives as an elitist mechanism. Furthermore, the neighborhood structure promotes the mating of close solutions. Different aggregative functions can be used in the MOEA/D framework. However, individuals with the least values are selected. In this work, we employ the Tchebycheff function, which is the most popular approach.

In order to include variation rate to the MOEA/D, we modify the selection criterion. Instead of preferring individuals with the least aggregative function value, we use the least value of the variation rate. For this, we employ the neighborhood structure of the original MOEA/D.

The pseudocode of VR-MOEA/D is shown in Algorithm 6.4. The change in this algorithm is very subtle In line 9, we compute the variation rate of the elements of the neighborhood and the offspring ($B(i) \cup y$) instead of only computing the values of the aggregative function.

Algorithm 6.4 Pseudocode of VR-MOEA/D

```
Require: $N$ number of solutions and weight vectors; $T$ neighborhood size.
Ensure: Final Population
 1: Initialize $N$ weight vectors $\lambda^1\lambda^2, \cdots, \lambda^N$
 2: Set $N$ subproblems defined by the $N$ weight vectors
 3: Set $N$ neighborhoods $B(i) = \{w_{i,1}, \cdots, w_{i,T}\}$, where $w_{i,j} = \lambda^j$ are the closest
    weight vectors to $\lambda^i$
 4: $\{x^1, \cdots, x^N\} \leftarrow$ InitializePopulation($N$)
 5: while $\neg$ StopCondition() do
 6:     for $i \in N$ do
 7:         Randomly select two solutions from $B(i)$ to generate an offspring $y$
 8:         Apply variation operators to $y$
 9:         Compute the values $V_r$ of $B(i) \cup y$ via the aggregative function $g$.
10:         Compute Variation Rate $V_t$ of the elements of $B(i) \cup y$
11:         for $x$ in $B(i)$ do
12:             if $r_y$ ¡ $r_x$ then
13:                 Replace $x$ with $y$
14:             end if
15:         end for
16:     end for
17: end while
18: return Children
```

6.1.6 Integration into SMS-EMOA

SMS-EMOA is an indicator-based algorithm, which means that it uses in its selection criterion the value of a certain performance indicator. In case of SMS-EMOA, it is the hypervolume indicator.

This algorithm is similar to the NSGA-II but it replaces the crowding distance by the contribution to the hypervolume of each individual in the last front. That is, the individuals of the last front with the biggest contribution to the hypervolume are preferred.

In this case, to adapt the SMS-EMOA we consider the inverse variation rate, as the original mechanism criterion of this algorithm prefers high values. Again, we use the last front as our neighborhood structure.

The pseudocode of an iteration of this method is shown in Algorithm 6.5 as the rest of the algorithm is basically the NSGA-II. We observe that, in line 12, we modify the values of the hypervolume contributions with the variation rate and we use them for the selection mechanism.

Algorithm 6.5 Iteration of the VR-SMS-EMOA

```
Require: Current population $P_t$
Ensure: Next population $P_{t+1}$
 1: $S_t = \emptyset,\ i = 1$
 2: $Q_t$ = apply variation operators to $P_t$
 3: $M_t = P_t \cup Q_t$
 4: $(F_1, F_2 \ldots, )$ = non-dominated-sorting($M_t$)
 5: while $|S_t| \leq N$ do
 6:     $t = S_t \cup F_i$
 7:     $i = i + 1$
 8: end while
 9: Add first fronts to $P_{t+1}$
10: $F_i$ := last added front
11: $V_t \leftarrow$ ComputeHypervolumeContributions($F_i$)
12: $R_t$ := SelectByVariationRate($V_t$)
13: $P_{t+1} : S_t \cup R_t$
```

6.2 Numerical Results

In this section, we show some numerical results and comparisons to the state-of-the-art to demonstrate the benefit and strength of the variation rate. To this end, we first compare the original version of each algorithm against its corresponding version that uses the variation rate (respectively, the inverse variation rate) on some widely used (non-Type III) benchmark problems. This is done in order to show that the performance of each algorithm is not significantly affected for standard problems. In the next step, we test again the original and variation rate versions of the selected MOEAs on some Type III problems, where the advantage of the variation rate becomes apparent.

The benchmark problems that we use for the first part of these experiments are the well known test problems DTLZ 1-4 [Deb et al., 2005], IDTLZ 1 and IDTLZ2 [Jain and Deb, 2014], as well as the test problems WFG 1-5 [Huband et al., 2006]. For the second part, we use the six following Type III problems.

The first Type III problem is taken from [Deb and Tiwari, 2008], which is defined as follows:

$$f_1(x) = \sum_{i=1}^{n} \sin(\pi x_i), \quad f_2(x) = \sum_{i=1}^{n} \cos(\pi x_i), \tag{6.4}$$

where $0 \leq x_i \leq 6$. This problem, denoted as OMNI1 in this work, has a total of 243 different disconnected components that form the Pareto set, and all of these components map to the same Pareto front.

The second problem, also taken from [Deb and Tiwari, 2008], is defined as follows:

$$f_1(x) = \sin\left(\pi \sum_{i=1}^{n} x_i\right), \quad f_2(x) = \cos\left(\pi \sum_{i=1}^{n} x_i\right), \tag{6.5}$$

where $0 \leq x_i \leq 1$, and $i = 1, 2, \ldots, 6$. This problem is denoted as OMNI2 in this work. Let $y = \sum_{i=1}^{6} x_i$ be the sum of the variables, then the Pareto set consists of the points x where $1 \leq y \leq 1.5$ or $3 \leq y \leq 3.5$. In addition, here, both connected components map to the same Pareto front. That is, every point on the Pareto front can be obtained in different infinite ways via the combinations of the variables mentioned.

The third Type III problem is the application stated in [Schütze et al., 2013, Sun et al., 2018], where subdivision techniques have been used to tackle the problem. It is stated as follows: for f_1, f_2 $\mathbb{R}^5 \to \mathbb{R}$, it is:

$$\begin{aligned} f_1(x) &= \sum_{i=1}^{n} x_i, \\ f_2(x) &= 1 - \prod_{i=1}^{n} (1 - w_i(x_i)), \end{aligned} \tag{6.6}$$

where

$$w_i(z) = \begin{cases} 0.01 \cdot \exp\left(-\left(\frac{z}{20}\right)^{2.5}\right), & \text{for } i = 1, 2, \\ 0.01 \cdot \exp\left(-\frac{z}{15}\right), & \text{for } 3 \leq i \leq 5. \end{cases} \tag{6.7}$$

Finally, we consider the methodology from [Rudolph et al., 2007] to construct three more problems, denoted in this work as RPH1, RPH2, and RPH3. These are bi-objective problems with two variables. In order to properly define them, we use the following functions.

First, we define the objective functions for the RPH1-3 problems

$$\begin{aligned} f_1(x) = (x_1 + a)^2 + x_2^2, \\ f_2(x) = (x_1 - a)^2 + x_2^2, \end{aligned} \tag{6.8}$$

where $x \in \mathbb{R}^2$ and $a \in \mathbb{R}^+$. The variants of the RPH problems are obtained with the following functions.

Let $t_1(x)$ and $t_2(x)$, with $x \in \mathbb{R}^2$, be the tile identifiers that are determined via:

$$\begin{aligned} \hat{t}_1(x) &= \operatorname{sgn}(x_1) \cdot \left\lceil \frac{|x_1| - \left(a + \frac{c}{2}\right)}{2a + c} \right\rceil, \\ \hat{t}_2(x) &= \operatorname{sgn}(x_2) \cdot \left\lceil \frac{|x_2| - \frac{b}{2}}{b} \right\rceil, \end{aligned} \tag{6.9}$$

which restrict the problem to nine tiles using the relation $t_i = \operatorname{sgn}(\hat{t}_i(x)) \cdot \min\{|\hat{t}_i|, 1\}$, with $i = 1, 2$.

Then, RPH1 is defined as $f_i^{(1)}(x) = f(\hat{x}(x))$, where $\bar{x} : \mathbb{R}^2 \to \mathbb{R}^2$ is defined by the following transformation:

$$\begin{aligned} \hat{x}_1(x_1) &= x_1 - t_1 \cdot (c + 2a), \\ \hat{x}_2(x_2) &= x_2 - t_2 \cdot b. \end{aligned} \tag{6.10}$$

For the RPH1-3 problems, we fix the constant values $a = 4$, $b = 10$, and $c = 4$.

The RPH2 problem is defined as the RPH1, but it rotates the variables. That is, for an angle θ, we have

$$r(x) = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} x, \tag{6.11}$$

and then $f_i^{(2)}(x) = f_i^{(1)}(r(x))$. In this work, we use $\theta = \frac{\pi}{4}$.

Finally, via the following transformation $d : \mathbb{R}^2 \to \mathbb{R}$

$$d(x) = x_1 \cdot \left(\frac{x_2 - L + \epsilon}{U - L} \right), \tag{6.12}$$

for some small $\epsilon > 0$ and where U and L denote the upper and lower bound of the search space, respectively; we can define the RPH3 as: $f_i^{(3)}(x) = f_i^{(2)}(d(x), x_2)$, which is a rotated and transformed problem. In this work, we use $\epsilon = 0.1$ for the RPH3 problem, while $L = -20$ and $U = 20$ are the the upper and lower bounds of each variable for the RPH1-3 problems.

We used the PlatEMO platform [Tian et al., 2017] to run our tests. The parameters settings of all the used algorithms are shown in Table 6.1. For all experiments, we executed 30 independent runs. The numerical results with the mean and standard deviation of the hypervolume and Δ_p indicators are shown in Tables 6.2 and 6.3. In these tables, we have put in boldface the best value between each pair of algorithms (the original version and the version with variation rate). We also performed the Wilcoxon test [Gibbons and Chakraborti, 2011] as our statistical significance test to validate the results. For this, we considered the value $\alpha = 0.05$. We put in gray the cell where such difference has statistical significance according to this test.

Table 6.1: Parameter configuration for each algorithm. Mutation probability m_p, crossover probability c_p, neighborhood size T, and number of reference points $\#Z$.

Parameter	**NSGA-II**	**NSGA-III**	**MOEA/D**	**SMS-EMOA**
m_p	$1/n$	$1/n$	0.1	$1/n$
c_p	0.8	0.8	1.0	0.8
T	-	-	20	-
$\#Z$	-	200	-	-

From the tables, we obtain that, for the classical benchmark problems, the original version of the selected MOEAs has a better Δ_p value than the variation rate version in 27 out of the 44 combinations, where only 19 out of these values have statistical significance, which is an expected result. However, it is important to notice that the variation rate versions do not always lose. According to the Δ_p indicator values, the variation rate versions are better in 17 out of 44 cases, but only in two with statistical significance.

For the hypervolume indicator, something similar happens. Here, the original version of the MOEAs is better than the variation rate version in 33 out of 44 runs, with statistical significance in 21 cases, while the variation rate version wins in 11 out of 44 cases for this indicator, where three of them have statistical significance.

In total, from the 88 possible combinations (algorithms, indicators and problems), we have statistical significance in 45 cases; this means that, almost 50% of the time, it is not possible to say that the original version is different than the variation rate version. Moreover, in the cases when we have statistical significance, we can see in Table 6.2 that the averaged values are very similar.

We observe the advantages of the variation rate versions with the Type III problems (see Table 6.3). For these problems, we use the Δ_p both in objective and decision space (we denote this in the table as Obj. Δ_p and Var. Δ_p, respectively). We observe a similar behavior in objective space; here, 16 out of 24 possible combinations are better for the original MOEAs (but only 6 out of these 34 have statistical significance). However, in decision space, the variation rate versions are better than the original versions; we have that 18 out of 24 combinations have better Δ_p values, almost all of them with statistical significance (only, in one case, we can not reject the null hypothesis).

Graphical results are shown in Figures 6.4–6.9; we plot the original MOEA and its corresponding variation rate version with the best value for each problem (according to the median of all runs using the Var. Δ_p).

In Figure 6.4, we can see that the obtained distribution is better for the variation rate version; it looks similar to that obtained by the original algorithm. However, we have to recall that this problem has 243 different disconnected components in the Pareto set, and some of them are overlapping in the plot. In Figure 6.5, we notice that the variation rate version can obtain points at the two regions in this representation (we plot y on both axes, where $y = \sum_{i=1}^{6} x_i$), while the original version is only able to compute points in one of them. On the other hand, in Figure 6.6, we can see that both the original and the variation rate versions can approximate the disconnected components of this problem well (except by the MOEA/D algorithm). Nonetheless, the variation rate version is better in this problem, according to the values of Table 6.3.

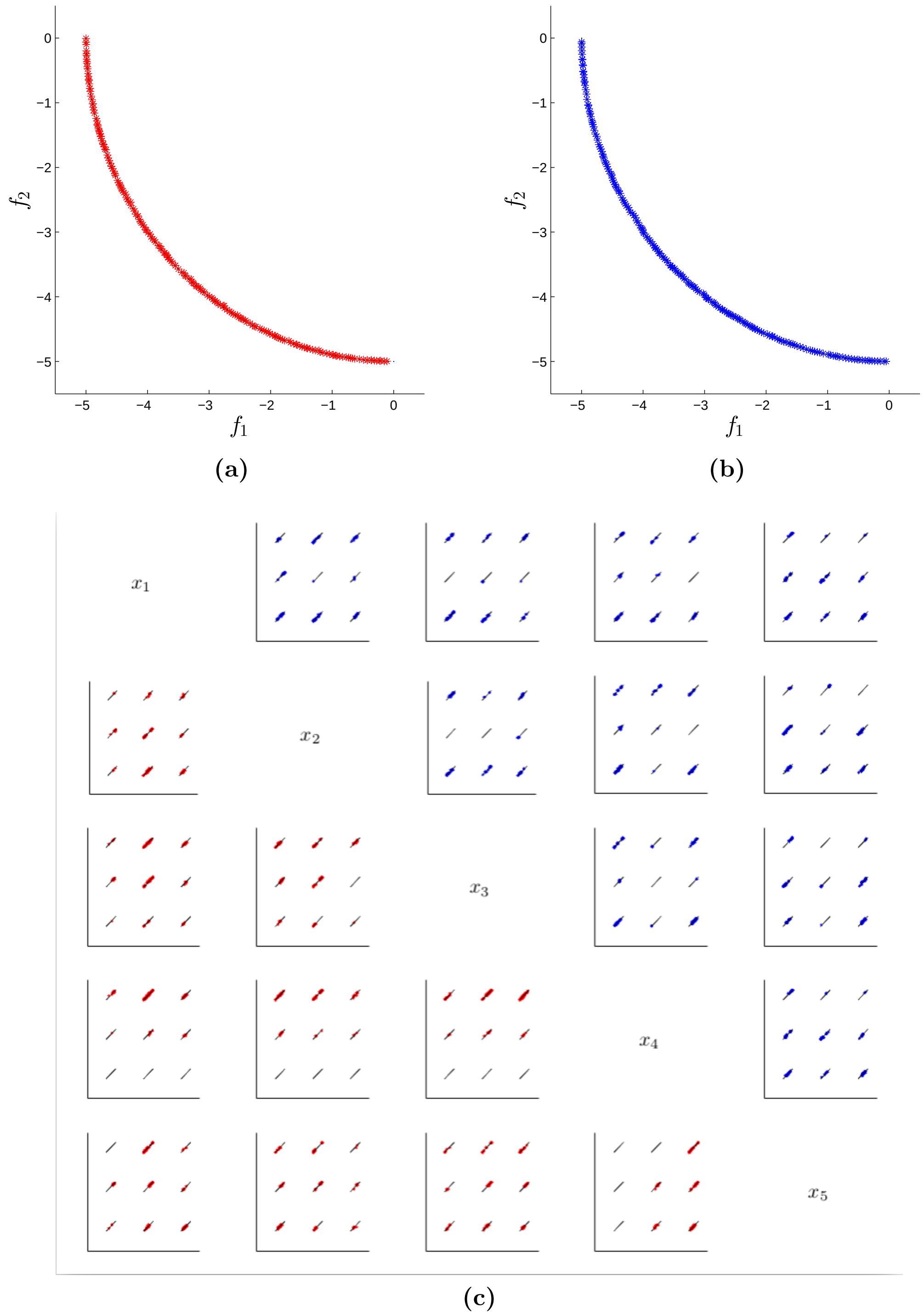
For the RPH problems, we can see in Figure 6.7 that the variation rate version of the NSGA-II can obtain four out of the nine disconnected components of the Pareto set, while the original version only gets three out of them; moreover, the distribution in decision space is also improved. In Figure 6.8, we can see again a similar behavior between the variation rate and the original version of the NSGA-III algorithm, which is also confirmed by the values of Table 6.3, where the variation rate version wins in three out of the four baseline algorithms for the RPH2 problem, but only in one has statistical significance. Finally, for the RPH3 problem, we can see in Figure 6.9 how the variation rate can significantly improve the performance of the MOEA/D algorithm in its distribution in decision space, as the original version only obtains points in one out of the nine disconnected components of the Pareto set, while the variation rate version obtains five out of them.

Table 6.2: Numerical results for the original and variation rate version of some MOEAs in standard benchmark test problems. We show the mean and standard deviation (up and down in the cell, respectively). We put in boldface the best value and in gray the cells with statistical significance according to the Wilcoxon test.

Problem	**Ind.**	**NSGA-II**		**NSGA-III**		**MOEAD**		**SMSEMOA**	
		Original	**VR**	**Original**	**VR**	**Original**	**VR**	**Original**	**VR**
DTLZ1	Δ_p	**0.0155** 0.0004	0.0181 0.0006	0.0105 0.0000	0.0105 0.0000	0.0105 0.0000	0.0105 0.0000	**0.0358** 0.0639	0.0719 0.1213
	HV	**0.8473** 0.0013	0.8454 0.0012	**0.8576** 0.0003	0.8575 0.0006	0.8571 0.0005	**0.8572** 0.0004	**0.8359** 0.0081	0.8351 0.0079
DTLZ2	Δ_p	0.0410 0.0008	0.0410 0.0014	0.0277 0.0000	0.0277 0.0000	**0.0277** 0.0000	0.0280 0.0000	0.0539 0.0041	**0.0536** 0.0041
	HV	0.5630 0.0018	**0.5641** 0.0015	0.5813 0.0000	0.5813 0.0001	**0.5814** 0.0000	0.5793 0.0001	**0.5593** 0.0013	0.5588 0.0020
DTLZ3	Δ_p	0.0666 0.0780	**0.0572** 0.0855	0.0301 0.0021	**0.0295** 0.0015	0.0316 0.0027	**0.0308** 0.0019	**0.0687** 0.0522	0.1386 0.2410
	HV	0.5577 0.0052	**0.5598** 0.0074	0.5714 0.0060	**0.5730** 0.0044	0.5660 0.0068	**0.5677** 0.0053	0.5526 0.0066	**0.5456** 0.0213
DTLZ4	Δ_p	**0.0400** 0.0007	0.0450 0.0015	0.0277 0.0000	0.0277 0.0000	0.1359 0.2069	**0.0293** 0.0000	**0.0719** 0.1096	0.0725 0.1095
	HV	0.5650 0.0013	**0.5676** 0.0013	0.5812 0.0001	**0.5813** 0.0001	0.5287 0.0971	**0.5721** 0.0002	0.5521 0.0483	**0.5522** 0.0481
IDTLZ1	Δ_p	**0.0156** 0.0005	0.0187 0.0011	0.1542 0.2459	**0.0774** 0.1318	**0.0177** 0.0000	0.0181 0.0001	**0.0371** 0.0708	0.0422 0.0984
	HV	**0.2301** 0.0017	0.2268 0.0018	**0.2331** 0.0007	0.2326 0.0006	**0.2287** 0.0001	0.2275 0.0002	**0.2218** 0.0035	0.2217 0.0038
IDTLZ2	Δ_p	**0.0394** 0.0008	0.0406 0.0012	0.0412 0.0008	0.0412 0.0010	0.0475 0.0008	**0.0456** 0.0007	0.0513 0.0035	**0.0510** 0.0048
	HV	**0.5495** 0.0011	0.5425 0.0019	**0.5550** 0.0020	0.5533 0.0017	**0.5565** 0.0001	0.5548 0.0002	**0.5410** 0.0023	0.5408 0.0018
WFG1	Δ_p	**0.1626** 0.0224	0.2488 0.0405	**0.2143** 0.0266	0.2392 0.0342	**0.2495** 0.0176	0.7113 0.0929	0.3413 0.0353	**0.3408** 0.0283
	HV	**0.9257** 0.0095	0.8582 0.0249	**0.8740** 0.0157	0.8629 0.0198	**0.8892** 0.0237	0.5933 0.0428	0.8697 0.0170	**0.8725** 0.0152
WFG2	Δ_p	**0.1270** 0.0072	0.1329 0.0083	0.0830 0.0012	0.0830 0.0013	**0.2166** 0.0335	0.2222 0.0320	0.1405 0.0069	**0.1377** 0.0067
	HV	**0.9356** 0.0009	0.9334 0.0010	**0.9393** 0.0006	0.9392 0.0006	**0.9203** 0.0061	0.9200 0.0049	**0.9309** 0.0015	0.9307 0.0011
WFG3	Δ_p	**0.5261** 0.0241	0.7354 0.0234	0.7229 0.0180	**0.7200** 0.0224	**0.8110** 0.0261	0.8788 0.0462	**0.4541** 0.0488	0.4633 0.0653
	HV	**0.4131** 0.0013	0.4080 0.0023	**0.4013** 0.0016	0.4006 0.0018	**0.3826** 0.0163	0.3613 0.0142	**0.3672** 0.0052	0.3649 0.0033
WFG4	Δ_p	**0.1641** 0.0044	0.1968 0.0053	**0.1142** 0.0007	0.1147 0.0007	**0.1361** 0.0028	0.2240 0.0053	0.2349 0.0099	**0.2301** 0.0136
	HV	**0.5502** 0.0026	0.5328 0.0029	**0.5751** 0.0008	0.5741 0.0008	**0.5550** 0.0028	0.5180 0.0029	0.5336 0.0026	**0.5339** 0.0031
WFG5	Δ_p	**0.1857** 0.0038	0.1927 0.0041	**0.1385** 0.0004	0.1388 0.0003	**0.1515** 0.0013	0.1549 0.0014	0.2191 0.0149	**0.2181** 0.0131
	HV	**0.5178** 0.0032	0.5148 0.0029	**0.5396** 0.0003	0.5393 0.0002	**0.5225** 0.0023	0.5200 0.0029	**0.5168** 0.0021	0.5166 0.0027

Table 6.3: Numerical results for the original and variation rate version of some MOEAs in Type III test problems. We show the mean and standard deviation (up and down in the cell, respectively). We put in boldface the best value and in gray the cells with statistical significance according to the Wilcoxon test.

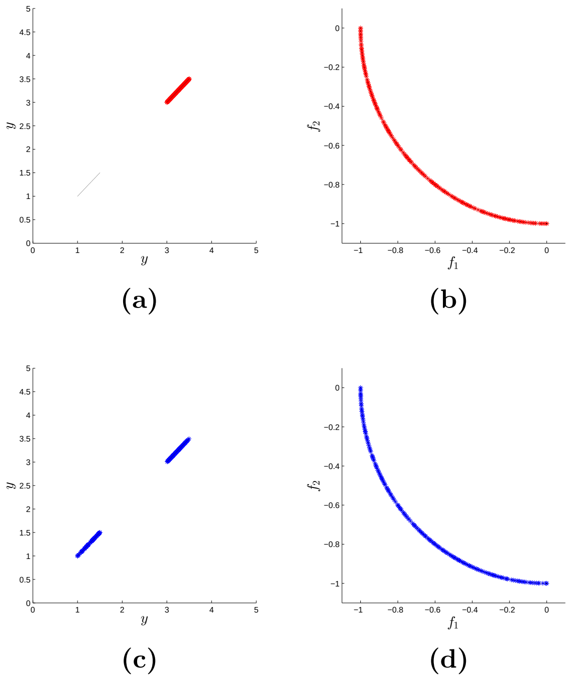
Problem	**Ind.**	**NSGA-II**		**NSGA-III**		**MOEAD**		**SMSEMOA**	
		Original	**VR**	**Original**	**VR**	**Original**	**VR**	**Original**	**VR**
OMNI1	Obj Δ_p	**0.0218** 0.0006	0.0220 0.0005	**0.0293** 0.0030	0.0306 0.0046	**0.1605** 0.0145	0.2363 0.0547	0.0210 0.0017	**0.0206** 0.0012
	Var Δ_p	2.2071 0.2928	**1.8804** 0.1603	**1.8876** 0.1852	1.9313 0.2480	4.3767 0.6295	**3.2976** 0.7931	2.2525 0.2978	**2.0409** 0.2544
OMNI2	Obj Δ_p	**0.0038** 0.0001	0.0039 0.0001	0.0048 0.0000	0.0048 0.0000	**0.0237** 0.0000	0.0278 0.0076	0.0039 0.0001	0.0039 0.0000
	Var Δ_p	1.1649 0.0360	**0.7206** 0.0075	0.8714 0.0956	**0.7576** 0.0312	1.2561 0.1039	**0.9800** 0.1302	0.9073 0.0853	**0.7366** 0.0071
SCM1	Obj Δ_p	0.3149 0.0135	**0.3138** 0.0123	0.3599 0.0198	**0.3594** 0.0303	**81.3852** 0.0043	81.3853 0.0042	0.2852 0.0297	**0.2681** 0.0252
	Var Δ_p	**3.7266** 0.1461	3.7751 0.1440	**3.5737** 0.1586	3.6209 0.1468	**38.5910** 0.0020	38.5912 0.0022	2.5058 0.0309	**2.4572** 0.0261
RPH1	Obj Δ_p	**0.1636** 0.0042	0.1653 0.0040	**0.4323** 0.0028	0.4334 0.0042	**2.6365** 0.0027	3.3847 0.7119	0.1551 0.0023	**0.1535** 0.0026
	Var Δ_p	6.8061 2.5349	**4.2771** 0.7180	1.6001 1.3496	**1.0450** 0.8738	8.0157 2.8617	**5.2802** 0.7714	**2.7075** 1.9102	5.3446 0.4255
RPH2	Obj Δ_p	**0.1725** 0.0027	0.1745 0.0051	**0.4302** 0.0109	0.4365 0.0141	**2.6459** 0.0175	3.1017 0.6411	0.1580 0.0038	**0.1574** 0.0035
	Var Δ_p	**2.6822** 0.9276	3.0836 0.7577	1.4661 0.5815	**1.2051** 0.6150	11.3906 3.0919	**6.6796** 1.5272	1.2333 0.5085	**1.1097** 0.5499
RPH3	Obj Δ_p	**0.1701** 0.0040	0.1732 0.0047	**0.4314** 0.0182	0.4340 0.0140	**2.6572** 0.0360	3.1012 0.6169	**0.1555** 0.0038	0.1564 0.0032
	Var Δ_p	6.3361 1.7299	**2.8003** 0.8529	3.0233 1.9181	**2.1521** 0.9408	8.6873 2.0764	**4.7500** 1.5926	2.8545 1.1084	**1.6417** 0.5711



(a) **(b)**

(c)

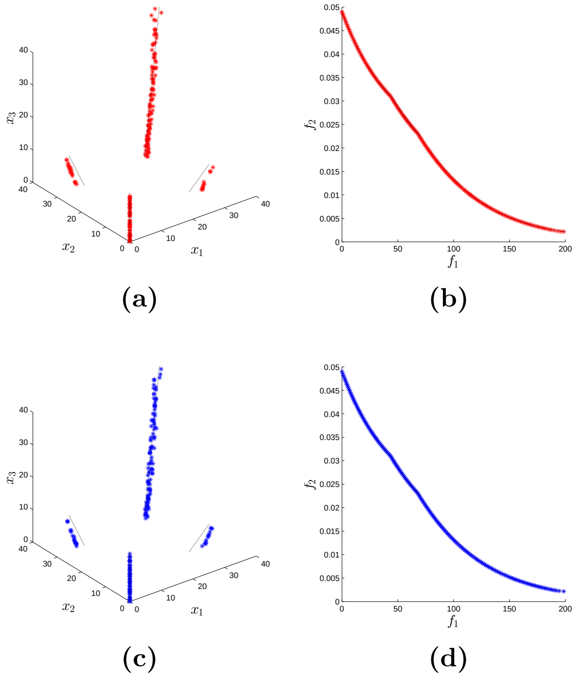
Figure 6.4: Graphical results of the run with the median values on the OMNI1 function with the NSGA-II algorithm. (**a**) Objective Space Original; (**b**) Objective Space VR; (**c**) Decision Space, pairwise plot of each variable. The left-down and red marks correspond to the original algorithm, while the right-up and blue ones are the VR version.



(a) **(b)**

(c) **(d)**

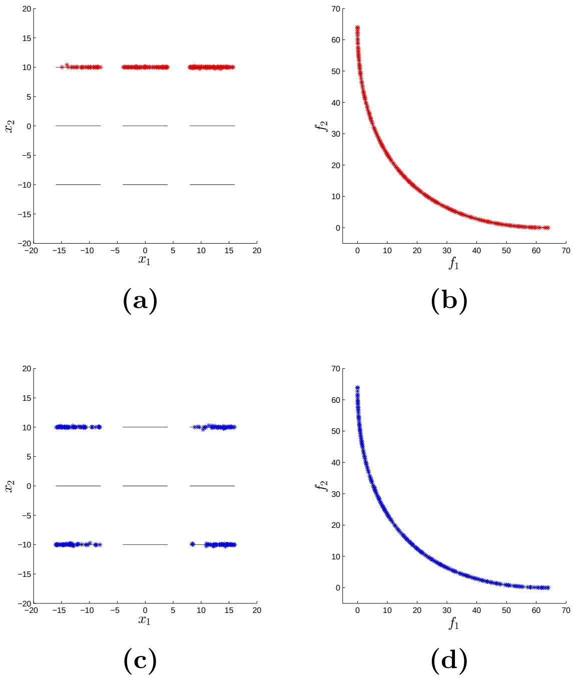
Figure 6.5: Graphical results of the run with the median values on the OMNI2 function with the NSGA-II algorithm. (**a**) Decision Space Original; (**b**) Objective Space Original; (**c**) Decision Space VR; (**d**) Objective Space VR.



(a) **(b)**

(c) **(d)**

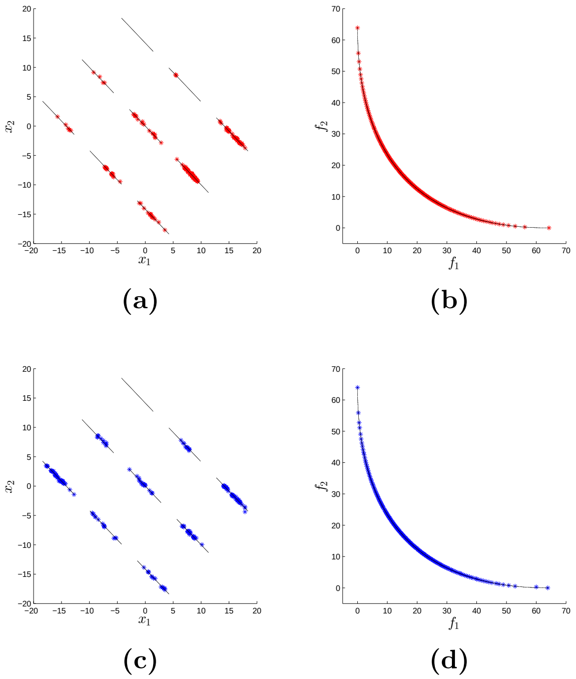
Figure 6.6: Graphical results of the run with the median values for the SCM1 function for the SMS-EMOA algorithm. (**a**) Decision Space Original; (**b**) Objective Space Original; (**c**) Decision Space VR; (**d**) Objective Space VR.



(a) **(b)**

(c) **(d)**

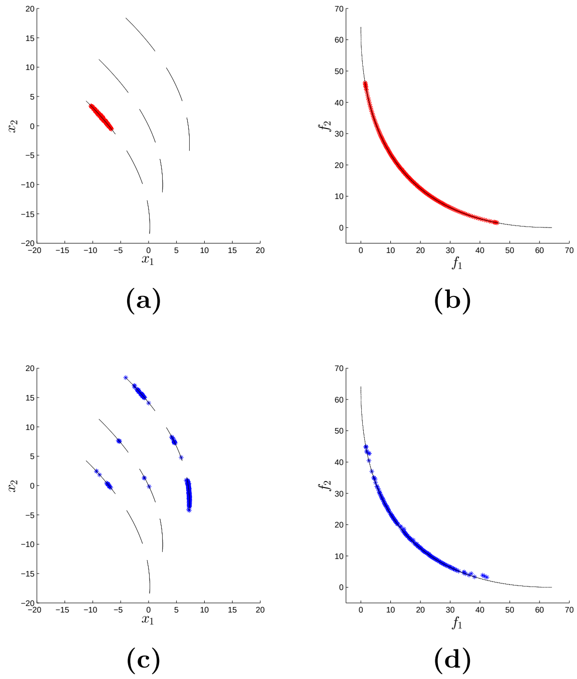
Figure 6.7: Graphical results of the run with the median values on the RPH1 function with the NSGA-II algorithm. (**a**) Decision Space Original; (**b**) Objective Space Original; (**c**) Decision Space VR; (**d**) Objective Space VR.



(a) **(b)**

(c) **(d)**

Figure 6.8: Graphical results of the run with the median values on the RPH2 function with the NSGA-III algorithm. (**a**) Decision Space Original; (**b**) Objective Space Original; (**c**) Decision Space VR; (**d**) Objective Space VR.



(a) **(b)**

(c) **(d)**

Figure 6.9: Graphical results of the run with the median values on the RPH3 function with the MOEA/D algorithm. (**a**) Decision Space Original; (**b**) Objective Space Original; (**c**) Decision Space VR; (**d**) Objective Space VR.

Chapter 7

A New Benchmark suite for Equality Constrained MOPs

As we stated in Section 2, the PE and PT methods can deal with equality constrained MOPs. This is an important feature of these methods, as MOEAs have a lot of problems to properly deal with this kind of problems (we show this with the numerical results of this chapter). Moreover, in the literature there is not a benchmark for Equality Constrained MOPs. For this reason, we propose a procedure to construct this kind of problems, that are scalable in variables, objectives, and constraints. We describe our benchmark in this chapter.

7.1 Hyper-spheres as equality constraints

For the construction of our test problems, we will use hyper-spheres. In the following, we discuss how they can be used so that the resulting MOP is scalable both in the number of decision variables and the number of constraints.

7.1.1 Hyper-spheres

The hyper-sphere or m-sphere

$$S^m = \{x \in \mathbb{R}^{m+1} \,:\, \|x\|_2^2 = r\}, \tag{7.1}$$

with $m \geq 0$, defines a m-dimensional manifold that is embedded on $(m+1)$-dimensional Euclidean space. We can use the fact that the intersection of two m-spheres is a $(m-1)$-sphere under certain conditions in order to construct an object of dimension $m-1$. For this, let $S_1^m, S_2^m \subset \mathbb{R}^{m+1}$ be two m-spheres with centers $c, c' \in \mathbb{R}^{m+1}$ and radii $r, r^{'} \in \mathbb{R}$. Further, suppose that $c_i = c_i',\ i = 1, \ldots, m$ and $0 < |c_{m+1} - c_{m+1}'| < \min\{r, r'\}$. Then,

$$\begin{aligned}(x_1 - c_1)^2 + \ldots + (x_{m+1} - c_{m+1})^2 &= r^2 && (7.2)\\ (x_1 - c_1')^2 + \ldots + (x_{m+1} - c_{m+1}')^2 &= r'^2. && (7.3)\end{aligned}$$

From (7.2) and (7.3) it follows that

$$2(c_{m-1}' - c_{m-1})x_{m+1} = r^2 - r'^2 - c_{m+1}^2 + c_{m+1}'^2, \quad (7.4)$$

so that any point in the intersection $S_1^m \cap S_2^m$ must lie in the hyperplane

$$x_{m+1} = \frac{r^2 - r'^2 - c_{m+1}^2 + c_{m+1}'^2}{2(c_{m+1}' - c_{m+1})}, \quad (7.5)$$

and its equation is

$$(x_1 - c_1)^2 + \ldots + (x_m - c_m)^2 = r_m^2, \quad (7.6)$$

where r_m^2 may be found from either Eq. (7.2) or (7.3). In particular, if we consider two spheres S_1^m and S_2^m with the same radius $r \in \mathbb{R}^+$ and centers $c \in \mathbb{R}^{m+1}$ and $c' := c + re_{m+1}$, respectively, where $e_{m+1} \in \mathbb{R}^{m+1}$ is the $(m+1)$-canonical vector. Then

$$r_m^2 = \frac{3}{4}r^2, \quad (7.7)$$

and this $(m-1)$-sphere lies in

$$x_{m+1} = c_{m+1} + \frac{r}{2}. \quad (7.8)$$

Following this way, we can add any number of spheres to reduce the dimension. For instance, we know that $\hat{S}^{k-3} = \hat{S}^{k-2} \cap S^{k-2}$, where $\hat{S}^{k-2} \neq S^{k-2}$. Now, the next sphere to intersect must be a S^{k-3} sphere, i.e, $\hat{S}^{k-4} = \hat{S}^{k-3} \cap S^{k-3} = (\hat{S}^{k-2} \cap S^{k-2}) \cap S^{k-3}$, and so on. We can express this recursively as follows

$$\hat{S}^{i-1} = \hat{S}^i \cap S^i, \quad (7.9)$$

where

$$\hat{S}^i = S^{k-2} \cap \left(\bigcap_{j=k-2}^{i+1} S^j \right). \quad (7.10)$$

Let $c' \in \mathbb{R}^{k-1}$ denote the "reference center" vector and let $r \in \mathbb{R}^+$ be the "reference radius". Further, let $S^{i-1} \in \mathbb{R}^i$ be the spheres

$$(x_1 - c_1)^2 + \ldots + (x_i - c_i)^2 = r_i^2, \quad (7.11)$$

where $c_j = c'_j$, $j = 1, \ldots, i-1$, $c_i = c'_i + r_{i+1}$; then, for the intersection S^{i-1} as in (7.9), we have

$$r_i^2 = \frac{3}{4} r_{i+1}^2, \tag{7.12}$$

with $r_{k-1} = r$ and $i = k-1, k-2, \ldots, 1$.

7.1.2 Embedding into higher dimensions

In order to be scalable in the number of decision variables, it is important that our spheres can be placed in $\mathbb{R}^n$ for any value of n. Let S^{m-1} be a sphere with center $c \in \mathbb{R}^m$ and radius $r \in \mathbb{R}^+$ as in (7.9), such that $S^{m-1} \in [0,1]^m \subset \mathbb{R}^m$, where $0 < m < k \leq n$. Then, the following degenerate sphere expression fulfills:

$$\left(\sum_{i=1}^{m} (x_i - c_i)^2 - r^2 \right)^2 + \sum_{j=m+1}^{n} (x_j - a_{j-m})^2 = 0, \tag{7.13}$$

where $a \in \mathbb{R}^{n-m}$. This comes because every term in the left hand side of (7.13) should be zero. In fact, although each constraint depends on $x \in \mathbb{R}^n$, its dimension is given by the dimension of the sphere.

In order to be scalable in the number p of equality constraints, we define

$$h_i(x) := \sum_{i=1}^{k-i} (x_i - c_i)^2 - r_i^2 = 0, \tag{7.14}$$

where $x \in \mathbb{R}^n$, $i = 1, \ldots, p$ and r_{i+1} defined as in Eq. (7.12). Notice that, $h_i(x)$ is basically the equation of the S^{k-i-1} sphere, but here every $x_j \in [0,1]$, $j = k-i+1, \ldots, n$ can take any value.

7.2 Equality constrained MOPs

Here, we propose the equality constrained MOPs, called Eq-DTLZ and Eq-IDTLZ, which utilize the structure of constraints as discussed in the previous section. All test problems are scalable in the number of decision variables (n) and objectives (k) as well as in the number p of equality constraints. We provide the analytical Pareto set for all of these problems.

The Eq-(I)DTLZ problems are based on the DTLZ problems, that were selected due to their properties, which allows performing a study on the behavior of some MOEAs under different conditions when we include some equality constraints. These properties are as follows:

DTLZ1 Multi-objective problem with a linear Pareto-optimal front. The difficulty in this problem is to converge to the hyper-plane. The search space contains $(11^z - 1)$ local Pareto-optimal fronts, where $|x_M| = z$, each of which can be attracted by a MOEA.

DTLZ2 This function can also be used to investigate a MOEA's ability to scale up its performance in a large number of objectives. The Pareto front is concave. For $k > 3$ the Pareto-optimal solutions must lie inside the first quadrant of the unit sphere in a three-objective plot with f_k as one of the axes.

DTLZ3 The above problem introduces $(3^z - 1)$ local Pareto-optimal fronts, and one global Pareto-optimal front. All local Pareto fronts are parallel to the global Pareto front and an MOEA can get stuck at any of them. The Pareto front is concave, scalable, and multi-frontal.

DTLZ4 This function is a modification of DTLZ2. This modification allows a dense solution set to exist near the $f_k - f_1$ plane. It is used to investigate a MOEA's ability to maintain a good distribution of solutions. The Pareto-optimal front is concave and separable.

IDTLZ1-2 Such functions are modified so that the corresponding Pareto-optimal front is inverted. The problem is such that minimizing each objective function has a unique solution. It is called an inverted function because it is in disagreement with our defined hyper-plane for which the maximum (and not minimum) point of each objective among all points on the hyper-plane is a unique point.

7.2.1 Eq-DTLZ

We will refer to the Pareto set of DTLZ 1-4 [Deb et al., 2005] as $P_B \subset \mathbb{R}^n$ which is stated as

$$\begin{aligned} P_B := \{x \in \mathbb{R}^n : x_i \in [0,1],\ i = 1, \dots, k-1, \\ x_j = 0.5,\ j = k, \dots, n\}. \end{aligned} \tag{7.15}$$

We can see from (7.15) that the Pareto set lies in the $(k-1)$-dimensional hypercube, i.e., the DTLZ 1-4 problems are non-degenerated. In other words, the dimension of the Pareto set and the Pareto front is $k - 1$. The idea of Eq-DTLZ is to include some equality constraints in such a way that for every added constraint we can reduce the dimension of the Pareto set/front by one.

For the DTLZ problems, we can state a defined in Eq. (7.13) as $a = \{a_i = x_{m+i},\ i = 1, 2, \dots, k - m - 1,\ a_j = 0.5,\ j = k, \dots, n\}$, where each x_{m+i} is defined as in Eq. (7.8). Notice that, we can write any constraint (the spheres $S^{k-2}, \dots S^m$) using Eq. (7.13) with $m = k - 2, \dots, m$, respectively. On the other hand, we have

$S^{m-1} \subseteq P_B \Rightarrow \mathcal{P}_D = S^{m-1}$, that is, the Pareto set is formed for all the vectors that satisfy Eq. (7.13) and this is a $(m-1)$-manifold. Hence, if we take a $(k-2)$-sphere $S^{k-2} \in [0,1]^{k-1} \subset \mathbb{R}^{k-1}$ as our first constraint, then the dimension of the Pareto set will be $k-2$, and the Pareto set will be formed for all the points such that

$$\sum_{i=1}^{k-1} (x_i - c_i)^2 = r^2, \tag{7.16}$$

where $c \in \mathbb{R}^{k-1}$ is the center of the sphere and $r \in \mathbb{R}$ is the radius. We can proceed analogously for a general p and obtain our first benchmark.

The definitions of the test problems Eq-DTLZ 1-4 can be seen in Table 7.1.

Pareto sets for Eq-DTLZ 1-4

Let $H : \mathbb{R}^n \to \mathbb{R}^p$

$$H(x) := (h_1(x), h_2(x), \ldots, h_p(x))^T, \tag{7.17}$$

be the map formed by the p equality constraints. From our previous analysis, we can characterize the solutions $x \in H^{-1}(0)$ as follows,

$$x = (v_1, \ldots, v_m, x'_{m+1}, \ldots, x'_{k-1}, x_k, \ldots, x_n)^T, \tag{7.18}$$

where $v \in \mathbb{R}^m$ is a point in the S^{m-1} sphere, x'_{m+i}, $i = 1, \ldots, k-m-1$, is computed as in Eq. (7.8) and $x_j \in [0,1]$, $j = k, \ldots, n$. This means that $\mathcal{P}_D \neq H^{-1}(0)$ and we need to find which vectors of $H(x)$ form the Pareto set.

Theorem 7.1. *Let P_B be defined as in (7.15) and $H(x)$ as in (7.17). Then $P_C = P_B \cap H^{-1}(0)$ is the Pareto set of Eq-DTLZ 1-4.*

Proof. Assume $\exists\, x \in P_C$ such that $\exists\, y \in H(x) \setminus P_U : y \prec x$. Since $y \in H(x) \setminus P_U$, we can define y as follows:

$$y := (v_1, \ldots, v_m, x'_{m+1}, \ldots, x'_{k-1}, 0.5 + \Delta_1, \ldots, 0.5 + \Delta_{n-k})^T, \tag{7.19}$$

where $\Delta \in \mathbb{R}^{n-k+1} \setminus \emptyset$; and we choose x as

$$x = (v_1, \ldots, v_m, x'_{m+1}, \ldots, x'_{k-1}, 0.5, \ldots, 0.5)^T. \tag{7.20}$$

Now, we have

1. Eq-DTLZ 1. For this problem, we have

$$f_i(x) = (1 + g(x))\,(1 - x_{k-1}) \prod_{i=1}^{k-i} x_i \tag{7.21}$$

where

$$\begin{aligned} g(x) = 100\Big(\ &|\ x\ | + \\ &\sum_{i=k}^{n} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\Big), \end{aligned}$$

then note that $g(x)$ reaches its global minimum at $x_j = 0.5$ for $j = k, \ldots, n$. Therefore

$$\begin{aligned} g(x) &< g(y), \quad \Rightarrow (1 + g(x)) < (1 + g(y)) \\ \Rightarrow f_i(x) &< f_i(y),\ \forall i = 1, \ldots, k. \end{aligned}$$

which contradicts the hypothesis.

2. Eq-DTLZ 2. For this problem, we have

$$f_i(x) = (1 + g(x)) \sin(0.5\pi x_{k-1}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i) \tag{7.22}$$

where

$$g(x) = \sum_{i=k}^{n} (x_i - 0.5)^2,$$

then

$$\begin{aligned} g(x) &= \sum_{i=k}^{n} (0.5 - 0.5)^2 = 0, \\ g(y) &= \sum_{i=k}^{n} (0.5 + \Delta_i - 0.5)^2 = \sum_{i=k}^{n} \Delta_i^2 > 0, \\ \Rightarrow g(x) &< g(y), \quad \Rightarrow (1 + g(x)) < (1 + g(y)) \\ \Rightarrow f_i(x) &< f_i(y),\ \forall i = 1, \ldots, k. \end{aligned}$$

which contradicts the hypothesis.

3. Eq-DTLZ 3. For this problem, we have

$$f_i(x) = (1 + g(x)) \sin(0.5\pi x_{k-1}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i) \tag{7.23}$$

where

$$g(x) = 100\Big(\mid x \mid + \sum_{i=k}^{n} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\Big).$$

Note that $g(x)$ is the same function as in Eq-DTLZ 1. That is, the proof follows analogously to those shown before.

4. Eq-DTLZ 4. For this problem, we have

$$f_i(x) = (1 + g(x)) \sin(0.5\pi x_{k-1}^{\alpha}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i^{\alpha}) \tag{7.24}$$

with $g(x)$ defined as in Eq-DTLZ 2 and the α parameter allows for a meta-variable mapping in Eq-DTLZ 2 $x_i \to x_i^{\alpha}$, the authors suggest $\alpha = 100$. As $g(x)$ is the same function as in Eq-DTLZ 2, the proof follows analogously.

Finally, for all cases we conclude that $\forall y \in H(x) \setminus P_U$, $\exists x \in P_C$ such that $x \prec y$. On the other hand, recall that all the points in P_B are non-dominated with respect to each other; therefore, $P_C \subset P_B$ is also a set of non-dominated solutions. □

7.2.2 Eq-IDTLZ

For these functions, the following transformation is applied to each objective $\tilde{f}_i(x)$ for DTLZ 1-4:

$$\tilde{f}_i(x) := 0.5(1 + g(x)) - f_i(x). \tag{7.25}$$

Hereby, $g(x)$ is as for the DTLZ functions. These problems have the same Pareto set as the DTLZ 1-4 problems, with the difference that the Pareto fronts are rotated.

Extending the above approach to inverted DTLZ 1-4 problems, we obtain the following result

Theorem 7.2. *Let P_B be as defined in* (7.15) *and $H(x)$ as in* (7.17)*. Then $P_C = P_B \cap H^{-1}(0)$ is the Pareto set of Eq-IDTLZ 1-4.*

Proof. Recall that for the inverted DTLZ 1-4 problems

$$f_i(x) = 0.5(1 + g(x)) - f_i(x),$$

but $g(x)$ is the same as the original DTLZ 1-4 problems. So, by the proof of Theorem 7.2.1, we know that

$$\begin{aligned} (1+g(x)) &< (1+g(y)) \\ \Rightarrow 0.5(1+g(x)) &< 0.5(1+g(y)), \end{aligned} \tag{7.26}$$

and that

$$f_i(x) < f_i(y) \;\; \forall\, i = 1, \ldots, k. \tag{7.27}$$

Then, by (7.26) and (7.27), we have that

$$0.5(1+g(x)) - f_i(x) < 0.5(1+g(y)) - f_i(y) \;\; \forall\, i = 1, \ldots, k. \tag{7.28}$$

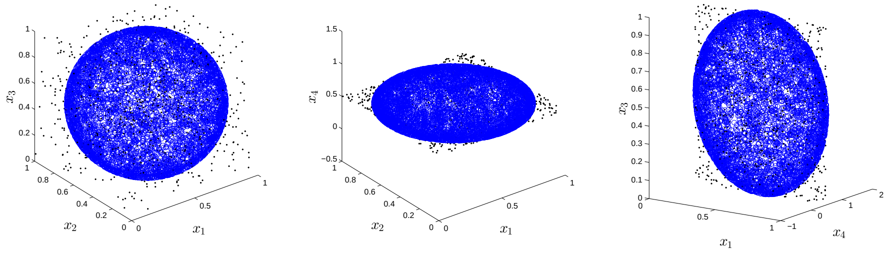
Therefore, $\forall y \in H(x) \setminus P_I$, $\exists x \in P_C$ such that $x \prec y$. On the other hand, recall that all the points in P_I are non-dominated with respect to each other; thus, $P_C \subset P_I$ is also a set of non-dominated solutions. □

The definitions of the test problems Eq-DTLZ 1-4 and Eq-DTLZ 1-2 can be seen in Table 7.1.

7.2.3 Examples

In the following, we present some examples of the Pareto sets of the proposed MOPs. Figure 7.1 shows the Pareto set of Eq-(I)DTLZ 1-4 for $k = 4$, $n = 13$, and $p = 1$. Analogously, Figure 7.2 corresponds to the Pareto set of all proposed MOPs for $k = 4$, $n = 13$, and $p = 2$. We can see that the dimension of the Pareto set is reduced when we add constraints, that is, for the same problem we have a sphere when $p = 1$ and a circle when $p = 2$. Notice that, as we consider $k = 4$ for the examples shown in Figures 7.2 and 7.1, then we can represent the main characteristics of all projections with the three cases that we show (remember that, for unconstrained problems, the dimension of the Pareto set is $k-1$; in particular, for these examples, we have a three-dimensional object given by $x_i \in [0, 1]$, $i = 1, 2, 3$, and $x_j = 0.5$, $j = 4, \ldots, 13$). That is, we can see projections of the three principal variables x_1, x_2, and x_3 (Figures 7.1a and 7.2a), two of them (Figures 7.1b and 7.2b), or one of them (Figures 7.1c and 7.2c). It is worth noticing that, in all the cases where none of the above variables are involved, we have that $x_i = 0.5$ for $i = 4, \ldots 13$.

Thus, with this approach, we can reduce the dimension of both the Pareto set and Pareto front via the choice of $H(x)$ (see Table 7.2). In the following, we present an example that shows how to build a constrained MOP with this proposal. Given a MOP with $k = 5$ and $n = 10$, we set $p = 3$, meaning that we want 3 constraints.

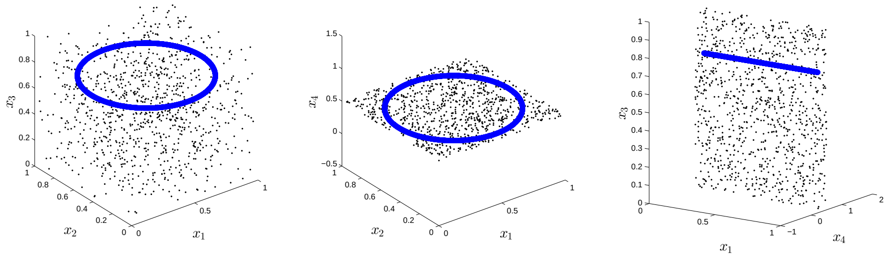


(a)

(b) x_1, x_2, x_4

(c) x_1, x_3, x_4

Figure 7.1: Different projections of the analytic Pareto set with $k = 4$, $n = 13$ and one equality constraint. Every plot shows the box constrained Pareto set of the problem (dot region), and the blue disk corresponds to the equality constrained Pareto set.



(a)

(b) x_1, x_2, x_4

(c) x_1, x_3, x_4

Figure 7.2: Different projections of the analytic Pareto set with $k = 4$, $n = 13$ and two equality constraints. Every plot shows the box constrained Pareto set of the problem (dot region), and the blue ring corresponds to the equality constrained Pareto set.

Then, we can define the center and radius as $c = (0.5, \dots, 0.5)^T \in \mathbb{R}^{10}$, and $r = 0.4$, which produce the set of equality constraints $H(x) := (h_1(x), h_2(x), h_3(x))^T$, where:

$$
\begin{aligned}
h_1(x) &= \sum_{i=1}^{4} (x_i - 0.5)^2 - 0.16 = 0, \\
h_2(x) &= \sum_{i=1}^{3} (x_i - 0.5)^2 + (x_4 - 0.9)^2 - 0.16 = 0 \\
h_3(x) &= \sum_{i=1}^{2} (x_i - 0.5)^2 + \Big(x_3 - (0.5 + \sqrt{0.012})\Big)^2 \\
&\quad -0.012 = 0.
\end{aligned}
$$

Table 7.1: Eq-DTLZ and Eq-IDTLZ benchmark problems

Function		Definition			Box
Eq-DTLZ	**Eq-DTLZ1**	$f_i(x) = (1 + g(x))\,(1 - x_{k-1}) \prod_{i=1}^{k-i} x_i,$ where, $i = 1, \ldots, k,$ $g(x) = 100 \left(\mid x \mid + \sum_{i=k}^{n} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right)$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
	Eq-DTLZ2	$f_i(x) = (1 + g(x)) \sin(0.5\pi x_{k-1}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i),$ where, $i = 1, \ldots, k,$ $g(x) = \sum_{i=k}^{n} (x_i - 0.5)^2$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
	Eq-DTLZ3	$f_i(x) = (1 + g(x)) \sin(0.5\pi x_{k-1}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i),$ where, $i = 1, \ldots, k,$ $g(x) = 100 \left(\mid x \mid + \sum_{i=k}^{n} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right)$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
	Eq-DTLZ4	$f_i(x) = (1 + g(x)) \sin(0.5\pi x_{k-1}^{\alpha}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i^{\alpha}),$ where, $i = 1, \ldots, k,$ $g(x) = \sum_{i=k}^{n} (x_i - 0.5)^2$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
Eq-IDTLZ	**Eq-IDTLZ1**	$f_i(x) = 0.5(1 + g(x)) - (1 + g(x))\,(1 - x_{k-1}) \prod_{i=1}^{k-i} x_i,$ where, $i = 1, \ldots, k,$ $g(x) = 100 \left(\mid x \mid + \sum_{i=k}^{n} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right)$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
	Eq-IDTLZ2	$f_i(x) = 0.5(1 + g(x)) - (1 + g(x)) \sin(0.5\pi x_{k-1}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i),$ where, $i = 1, \ldots, k,$ $g(x) = \sum_{i=k}^{n} (x_i - 0.5)^2$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
	Eq-IDTLZ3	$f_i(x) = 0.5(1 + g(x)) - (1 + g(x)) \sin(0.5\pi x_{k-1}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i),$ where, $i = 1, \ldots, k,$ $g(x) = 100 \left(\mid x \mid + \sum_{i=k}^{n} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right)$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	
	Eq-IDTLZ4	$f_i(x) = 0.5(1 + g(x)) - (1 + g(x)) \sin(0.5\pi x_{k-1}^{\alpha}) \prod_{i=1}^{k-i} \cos(0.5\pi x_i^{\alpha}),$ where, $i = 1, \ldots, k,$ $g(x) = \sum_{i=k}^{n} (x_i - 0.5)^2$			$0 \le x_j \le 1$ $j = 1, \ldots, n.$
		$k = 3$	$p = 1$	$h_1(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 - 0.4^2 = 0$	
		$k = 4$	$p = 1$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.4^2 = 0$	
			$p = 2$	$h_1(x) = \sum_{i=1}^{3} (x_i - 0.5)^2 - 0.5^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - 0.5)^2 + (x_3 - 1)^2 - 0.5^2 = 0$	

Table 7.2: Equality constraints for different values of k and p and the general case.

		$H_{k,p}(x)$
$k=3$	$p=1$	$h_1(x) = \sum_{i=1}^{2} (x_i - c_i)^2 - r^2 = 0$
$k=4$	$p=1$	$h_1(x) = \sum_{i=1}^{3} (x_i - c_i)^2 - r^2 = 0$
	$p=2$	$h_1(x) = \sum_{i=1}^{3} (x_i - c_i)^2 - r^2 = 0,$ $h_2(x) = \sum_{i=1}^{2} (x_i - c_i)^2 + (x_3 - (c_3 + r))^2 - r^2 = 0$
$k=5$	$p=1$	$h_1(x) = \sum_{i=1}^{4} (x_i - c_i)^2 - r^2 = 0$
	$p=2$	$h_1(x) = \sum_{i=1}^{4} (x_i - c_i)^2 - r^2 = 0,$ $h_2(x) = \sum_{i=1}^{3} (x_i - c_i)^2 + (x_4 - (c_4 + r))^2 - r^2 = 0$
	$p=3$	$h_1(x) = \sum_{i=1}^{4} (x_i - c_i)^2 - r^2 = 0,$ $h_2(x) = \sum_{i=1}^{3} (x_i - c_i)^2 + (x_4 - (c_4 + r))^2 - r^2 = 0$ $h_3(x) = \sum_{i=1}^{2} (x_i - c_i)^2 + (x_3 - (c_3 + r_1))^2 - r_1^2 = 0,$ where $r_1 = \sqrt{\frac{3}{4}r^2}$.
k	$p < k$	$h_i(x) = \sum_{i=1}^{k-i} (x_i - c_i)^2 - r_i^2 = 0,$ where $r_0 = r$.

7.3 Performance of MOEAs on Eq-(I)DTLZ

In this section, we present some numerical examples that show the behavior of some state-of-the-art MOEAs on the new benchmark problems. For this, we have selected NSGA-II [Deb et al., 2002], NSGA-III [Jain and Deb, 2014], Adaptive NSGA-III [Jain and Deb, 2014], MOEA/ D/ D [Li et al., 2015], and GDE3 [Kukkonen and Lampinen, 2005]. NSGA-II, NSGA-III and Adaptive NSGA-III uses feasibility rules in the selection process. In case the selection process considers two infeasible individuals, a penalty function is used to determine which individual violates more the constraints. GDE3 is an extension of differential evolution for global multi-objective optimization. This method handles constraints using the same principles as NSGA-

II. Finally, MOEA/D/D combines dominance and decomposition-based approaches for solving many-objective optimization problems. MOEA/D/D also uses feasibility rules and the penalty function proposed in NSGA-II, but it is not only used in the selection process. It is present also in the update procedure of the algorithm, and this consideration helps with the preservation of diversity of the population.

For all experiments we used the PlatEMO [Tian et al., 2017] framework, where we executed 30 independent runs using $50,000$, $100,000$ and $150,000$ function calls for MOPs with $k = 3$; and using $200,000$, $300,000$ and $500,000$ function calls for MOPs with $k = 4$. We stress that there exist some recent algorithms that deal with constraints such as CMOEA/D-DE-SR, CMOEA/D-DE-CDP [Jan and Khanum, 2013], and PPS-MOEA/D [Fan et al., 2019b]. However, we have to omit a comparison here since for none of the mentioned algorithms an implementation in PlatEMO was available, and have to leave this for future work.

Table 7.3 contains the algorithm parameter values used for the experiments. The performance indicators Δ_2 [Schütze et al., 2012] and the hypervolume HV [Zitzler et al., 2007] are used to measure the algorithm effectiveness. Tables 7.4 and 7.5 show the results for Eq-DTLZ 1-4 with $k = 3$ and $k = 4$, respectively. In Table 7.4 we can see the results for the Eq-IDTLZ 1-2 with $k = 3$; and finally, Table 7.5 shows the results for the Eq-IDTLZ 1-2 with $k = 4$. For all these tables, Column # shows the averaged number of feasible solutions at the end of each run. For this, we consider a solution x to be feasible if $\|H(x)\|_2 \leq 1e - 4$.

From the tables, as well as from the figures we can observe that the approximation qualities are not satisfying. This is likely a result from the fact that none of these MOEAs have been designed so far to incorporate equality constraints.

In Figure 7.3, we see the result for GDE3, which has, on average, 90 feasible solutions (see Table 7.4). However, most of these solutions are in the "tube" and their images are very far away from the real Pareto front (in the range that we selected, neither of them were displayed). On the other hand, we have the result of NSGA-II and NSGA-III, which are the MOEAs with the best indicator values for Eq-DTLZ1. In Figure 5.5, we show again GDE3, which has a similar performance, i.e., the feasible solutions are within the "tube"; and the two MOEAs with the best Δ_p values for Eq-DTLZ2, NSGA-II and NSGA-III.

Table 7.3: Parameters of the selected MOEAs.

MOEA	**Parameter**	**Value**
NSGA-II	Population size	100
	Crossover probability	0.8
	Mutation probability	$\frac{1}{n}$
	Distribution index for crossover	20
	Distribution index for mutation	20
NSGA-III ANSGA-III	Population size	92
	Reference points	91
	Crossover probability	1
	Mutation probability	$1/n$
	Distribution index for crossover	20
	Distribution index for mutation	20
MOEA\ D \ D	Population size	91
	# weight vectors	91
	Crossover probability	1
	Mutation probability	$1/n$
	Distribution index for crossover	30
	Distribution index for mutation	20
	Penalty parameter of PBI	5
	Neighborhood size	20
	Probability used to select in the neighborhood	0.9
GDE3	Population size	100
	CR	0.2
	F	0.2
	Distribution index for mutation	20

Table 7.4: Results for Eq-DTLZ 1-4 and Eq-IDTLZ 1-2 with $k = 3$ and $p = 1$ for some MOEAs. (# is average of the number of feasible solutions at the end of each run).

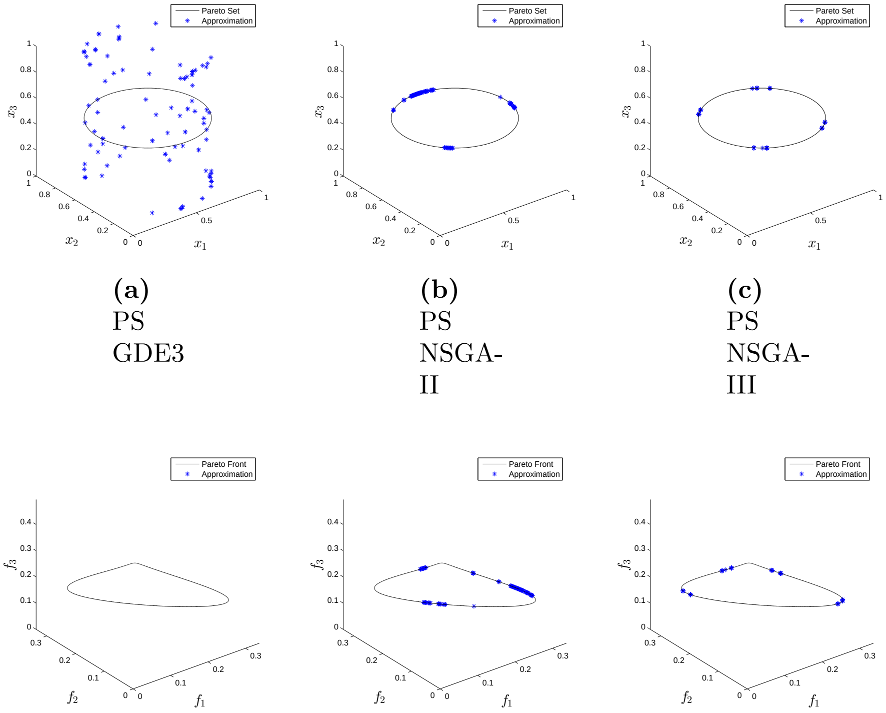
		50,000			100,000			150,000		
	Method	Δ_p	HV	#	Δ_p	HV	#	Δ_p	HV	#
Eq-DTLZ1	ANSGA-III	0.2323	9.2858e-03	1.00	0.1795	1.2811e-02	1.00	0.1832	1.2651e-02	1.00
	(std.dev)	(0.0719)	(0.0041)		(0.0722)	(0.0042)		(0.0767)	(0.0048)	
	GDE3	166.7477	0.0000e+00	0.41	149.6107	0.0000e+00	0.70	125.1485	0.0000e+00	0.91
	(std.dev)	(6.9725)	(0.0000)		(7.4593)	(0.0000)		(16.1758)	(0.0000)	
	MOEADD	2.2988	1.5387e-02	1.00	1.6887	1.5652e-02	1.00	0.6970	1.6555e-02	0.98
	(std.dev)	(3.9807)	(0.0047)		(4.4745)	(0.0049)		(2.4116)	(0.0054)	
	NSGA-II	0.1808	1.2619e-02	1.00	0.1826	1.3100e-02	1.00	0.1666	1.3606e-02	1.00
	(std.dev)	(0.0692)	(0.0049)		(0.0600)	(0.0043)		(0.0743)	(0.0048)	
	NSGA-III	0.1877	1.1123e-02	1.00	0.1731	1.2500e-02	1.00	0.1727	1.4026e-02	1.00
	(std.dev)	(0.0731)	(0.0042)		(0.0623)	(0.0042)		(0.0748)	(0.0053)	
Eq-DTLZ2	ANSGA-III	0.4164	1.7425e-01	1.00	0.4924	1.5215e-01	1.00	0.4351	1.8532e-01	1.00
	(std.dev)	(0.1686)	(0.0552)		(0.1641)	(0.0499)		(0.1537)	(0.0545)	
	GDE3	1.1905	1.4467e-03	0.43	1.1180	6.1652e-03	0.76	0.9887	1.8859e-02	0.98
	(std.dev)	(0.0378)	(0.0046)		(0.0505)	(0.0139)		(0.0723)	(0.0185)	
	MOEADD	0.4350	1.6434e-01	1.00	0.4305	1.6033e-01	1.00	0.3696	1.8951e-01	1.00
	(std.dev)	(0.1683)	(0.0608)		(0.1654)	(0.0585)		(0.1595)	(0.0607)	
	NSGA-II	0.4437	1.7386e-01	1.00	0.3977	1.8545e-01	1.00	0.3895	1.9099e-01	1.00
	(std.dev)	(0.1513)	(0.0516)		(0.1626)	(0.0521)		(0.1877)	(0.0558)	
	NSGA-III	0.4830	1.5836e-01	1.00	0.4276	1.8031e-01	1.00	0.3794	1.9586e-01	1.00
	(std.dev)	(0.1846)	(0.0562)		(0.1640)	(0.0522)		(0.1381)	(0.0470)	
Eq-DTLZ3	ANSGA-III	1.5875	5.2920e-02	1.00	0.5048	1.3578e-01	1.00	0.4635	1.5289e-01	1.00
	(std.dev)	(1.6046)	(0.0516)		(0.1701)	(0.0539)		(0.1784)	(0.0549)	
	GDE3	879.6119	0.0000e+00	0.41	820.9253	0.0000e+00	0.75	700.6133	0.0000e+00	0.96
	(std.dev)	(23.9536)	(0.0000)		(31.5546)	(0.0000)		(52.9224)	(0.0000)	
	MOEADD	14.5069	9.0199e-02	1.00	9.9210	1.5471e-01	1.00	1.7548	1.7833e-01	0.96
	(std.dev)	(19.7134)	(0.0625)		(18.3423)	(0.0515)		(5.7219)	(0.0594)	
	NSGA-II	0.7561	8.5652e-02	1.00	0.5296	1.2794e-01	1.00	0.5037	1.4055e-01	1.00
	(std.dev)	(0.4374)	(0.0597)		(0.1657)	(0.0384)		(0.1802)	(0.0571)	
	NSGA-III	1.7594	5.4301e-02	1.00	0.5369	1.2404e-01	1.00	0.4899	1.3495e-01	1.00
	(std.dev)	(2.0103)	(0.0579)		(0.1516)	(0.0381)		(0.1800)	(0.0489)	
Eq-DTLZ4	ANSGA-III	0.8927	0.0000e+00	1.00	0.9173	0.0000e+00	1.00	0.9924	0.0000e+00	1.00
	(std.dev)	(0.1908)	(0.0000)		(0.1542)	(0.0000)		(0.1458)	(0.0000)	
	GDE3	1.6249	0.0000e+00	0.44	0.6345	0.0000e+00	1.00	0.6249	0.0000e+00	1.00
	(std.dev)	(0.1573)	(0.0000)		(0.0680)	(0.0000)		(0.0279)	(0.0000)	
	MOEADD	1.0363	0.0000e+00	0.98	1.0361	0.0000e+00	0.90	1.0773	0.0000e+00	0.91
	(std.dev)	(0.1859)	(0.0000)		(0.1787)	(0.0000)		(0.1208)	(0.0000)	
	NSGA-II	0.8521	0.0000e+00	1.00	0.8599	0.0000e+00	1.00	0.8649	0.0000e+00	1.00
	(std.dev)	(0.1794)	(0.0000)		(0.1245)	(0.0000)		(0.1243)	(0.0000)	
	NSGA-III	0.9113	0.0000e+00	1.00	0.9365	0.0000e+00	1.00	0.9921	0.0000e+00	1.00
	(std.dev)	(0.1813)	(0.0000)		(0.1665)	(0.0000)		(0.1568)	(0.0000)	
Eq-IDTLZ1	ANSGA-III	0.2243	4.4889e-03	1.00	0.2243	4.1900e-03	1.00	0.1935	5.4621e-03	1.00
	(std.dev)	(0.0774)	(0.0021)		(0.0732)	(0.0022)		(0.0726)	(0.0023)	
	GDE3	283.1098	0.0000e+00	0.41	251.3672	0.0000e+00	0.71	212.4951	0.0000e+00	0.89
	(std.dev)	(10.4742)	(0.0000)		(13.4323)	(0.0000)		(21.6521)	(0.0000)	
	MOEADD	3.4448	6.3285e-03	1.00	0.7503	7.0050e-03	1.00	1.3854	7.2420e-03	1.00
	(std.dev)	(7.7558)	(0.0023)		(3.1174)	(0.0020)		(4.8200)	(0.0020)	
	NSGA-II	0.2304	4.3745e-03	1.00	0.2070	5.0626e-03	1.00	0.1838	5.7159e-03	1.00
	(std.dev)	(0.0646)	(0.0018)		(0.0641)	(0.0018)		(0.0704)	(0.0023)	
	NSGA-III	0.1953	4.5958e-03	1.00	0.2135	4.8155e-03	1.00	0.2172	5.1530e-03	1.00
	(std.dev)	(0.0626)	(0.0018)		(0.0753)	(0.0021)		(0.0743)	(0.0026)	
Eq-IDTLZ2	ANSGA-III	0.4379	1.0003e-01	1.00	0.4322	1.0958e-01	1.00	0.4400	9.9467e-02	1.00
	(std.dev)	(0.1775)	(0.0494)		(0.1916)	(0.0495)		(0.1927)	(0.0544)	
	GDE3	1.2222	2.7783e-04	0.44	1.1454	3.3265e-03	0.81	0.9874	8.7486e-03	0.99
	(std.dev)	(0.0358)	(0.0015)		(0.0563)	(0.0067)		(0.0817)	(0.0107)	
	MOEADD	0.4722	9.9690e-02	0.99	0.4598	1.0674e-01	1.00	0.4446	1.0927e-01	0.97
	(std.dev)	(0.1782)	(0.0373)		(0.1831)	(0.0363)		(0.2025)	(0.0460)	
	NSGA-II	0.4378	1.0107e-01	1.00	0.4056	1.1500e-01	1.00	0.3807	1.3126e-01	1.00
	(std.dev)	(0.1525)	(0.0424)		(0.1609)	(0.0412)		(0.1852)	(0.0478)	
	NSGA-III	0.4568	9.4090e-02	1.00	0.4307	1.1251e-01	1.00	0.4023	1.1577e-01	1.00
	(std.dev)	(0.1425)	(0.0424)		(0.1737)	(0.0394)		(0.1913)	(0.0462)	

Table 7.5: Results for Eq-DTLZ 1-4 and Eq-IDTLZ 1-2 with $k = 4$ and $p = 1$ for some MOEAs. (# is average of the number of feasible solutions at the end of each run).

		200,000			300,000			500,000		
	Method	Δ_p	HV	#	Δ_p	HV	#	Δ_p	HV	#
Eq-DTLZ1	ANSGA-III	0.1403	2.0917e-03	1.00	0.1263	2.1639e-03	1.00	0.1163	2.4119e-03	1.00
	(std.dev)	(0.0621)	(0.0007)		(0.0456)	(0.0006)		(0.0488)	(0.0006)	
	GDE3	105.4149	0.0000e+00	0.97	43.9482	0.0000e+00	1.00	19.2151	0.0000e+00	1.00
	(std.dev)	(29.1985)	(0.0000)		(26.2691)	(0.0000)		(9.6110)	(0.0000)	
	MOEADD	1.9029	2.0180e-03	1.00	0.2360	2.1493e-03	1.00	0.2203	2.4363e-03	0.99
	(std.dev)	(4.9098)	(0.0007)		(0.5022)	(0.0006)		(0.6569)	(0.0005)	
	NSGA-II	0.1400	2.1356e-03	1.00	0.1315	2.2726e-03	1.00	0.1093	2.6012e-03	1.00
	(std.dev)	(0.0607)	(0.0007)		(0.0623)	(0.0006)		(0.0412)	(0.0005)	
	NSGA-III	0.1388	1.9419e-03	1.00	0.1121	2.2874e-03	1.00	0.1335	2.1171e-03	1.00
	(std.dev)	(0.0461)	(0.0006)		(0.0309)	(0.0005)		(0.0639)	(0.0007)	
Eq-DTLZ2	ANSGA-III	0.3766	1.2962e-01	1.00	0.3881	1.2796e-01	1.00	0.3102	1.5029e-01	1.00
	(std.dev)	(0.1298)	(0.0343)		(0.1519)	(0.0436)		(0.1142)	(0.0351)	
	GDE3	1.0951	1.0227e-03	1.00	0.8815	8.4280e-03	1.00	0.6344	2.6954e-02	1.00
	(std.dev)	(0.0489)	(0.0024)		(0.0566)	(0.0093)		(0.0699)	(0.0174)	
	MOEADD	0.3926	1.0698e-01	1.00	0.3439	1.2296e-01	1.00	0.3559	1.2090e-01	0.98
	(std.dev)	(0.1347)	(0.0363)		(0.1065)	(0.0352)		(0.1309)	(0.0366)	
	NSGA-II	0.3180	1.5444e-01	1.00	0.3188	1.5177e-01	1.00	0.2875	1.6824e-01	1.00
	(std.dev)	(0.1204)	(0.0348)		(0.1238)	(0.0397)		(0.1351)	(0.0416)	
	NSGA-III	0.3784	1.2526e-01	1.00	0.3426	1.3301e-01	1.00	0.2687	1.5845e-01	1.00
	(std.dev)	(0.1304)	(0.0396)		(0.1156)	(0.0368)		(0.0873)	(0.0315)	
Eq-DTLZ3	ANSGA-III	0.4402	9.1970e-02	1.00	0.4020	1.0668e-01	1.00	0.4059	1.1820e-01	1.00
	(std.dev)	(0.1140)	(0.0245)		(0.0995)	(0.0264)		(0.1432)	(0.0371)	
	GDE3	702.8266	0.0000e+00	0.95	489.6774	0.0000e+00	1.00	235.1171	0.0000e+00	1.00
	(std.dev)	(90.5679)	(0.0000)		(130.7910)	(0.0000)		(124.3284)	(0.0000)	
	MOEADD	6.9869	1.0514e-01	1.00	7.5595	1.1124e-01	0.99	9.4657	1.1767e-01	0.96
	(std.dev)	(16.5245)	(0.0346)		(13.6181)	(0.0342)		(18.7491)	(0.0425)	
	NSGA-II	0.3990	1.1246e-01	1.00	0.4103	1.1237e-01	1.00	0.3694	1.3219e-01	1.00
	(std.dev)	(0.1033)	(0.0248)		(0.1127)	(0.0314)		(0.1519)	(0.0459)	
	NSGA-III	0.4513	9.4477e-02	1.00	0.4632	9.2126e-02	1.00	0.4373	1.0062e-01	1.00
	(std.dev)	(0.1432)	(0.0330)		(0.1296)	(0.0350)		(0.1343)	(0.0370)	
Eq-DTLZ4	ANSGA-III	1.1518	0.0000e+00	1.00	1.1655	0.0000e+00	1.00	1.1548	0.0000e+00	1.00
	(std.dev)	(0.0877)	(0.0000)		(0.0690)	(0.0000)		(0.0487)	(0.0000)	
	GDE3	1.7942	0.0000e+00	0.62	1.2749	0.0000e+00	0.97	1.1126	0.0000e+00	1.00
	(std.dev)	(0.1539)	(0.0000)		(0.2958)	(0.0000)		(0.0858)	(0.0000)	
	MOEADD	1.2004	0.0000e+00	0.98	1.1925	0.0000e+00	0.98	1.2064	0.0000e+00	0.98
	(std.dev)	(0.0606)	(0.0000)		(0.0812)	(0.0000)		(0.0494)	(0.0000)	
	NSGA-II	1.1111	0.0000e+00	1.00	1.1278	0.0000e+00	1.00	1.1441	0.0000e+00	1.00
	(std.dev)	(0.0755)	(0.0000)		(0.0513)	(0.0000)		(0.0421)	(0.0000)	
	NSGA-III	1.1492	0.0000e+00	1.00	1.1296	0.0000e+00	1.00	1.1601	0.0000e+00	1.00
	(std.dev)	(0.0762)	(0.0000)		(0.0883)	(0.0000)		(0.0692)	(0.0000)	
Eq-IDTLZ1	ANSGA-III	0.1770	4.5274e-04	1.00	0.1742	5.9833e-04	1.00	0.1579	6.0165e-04	1.00
	(std.dev)	(0.0485)	(0.0002)		(0.0481)	(0.0002)		(0.0569)	(0.0003)	
	GDE3	276.8964	0.0000e+00	0.87	124.6459	0.0000e+00	1.00	36.9768	2.3605e-05	1.00
	(std.dev)	(49.3092)	(0.0000)		(71.3200)	(0.0000)		(28.6105)	(0.0001)	
	MOEADD	2.4053	5.8740e-04	1.00	0.7574	5.2090e-04	1.00	2.6786	5.9474e-04	1.00
	(std.dev)	(8.3746)	(0.0003)		(3.1285)	(0.0003)		(7.3435)	(0.0003)	
	NSGA-II	0.1880	4.4054e-04	1.00	0.1710	5.8898e-04	1.00	0.1351	7.3235e-04	1.00
	(std.dev)	(0.0622)	(0.0003)		(0.0599)	(0.0002)		(0.0487)	(0.0004)	
	NSGA-III	0.1572	5.8704e-04	1.00	0.1704	5.6080e-04	1.00	0.1705	5.1858e-04	1.00
	(std.dev)	(0.0513)	(0.0002)		(0.0464)	(0.0002)		(0.0494)	(0.0003)	
Eq-IDTLZ2	ANSGA-III	0.3949	5.8823e-02	1.00	0.3737	6.4042e-02	1.00	0.3565	7.6060e-02	1.00
	(std.dev)	(0.1064)	(0.0254)		(0.1217)	(0.0282)		(0.1255)	(0.0256)	
	GDE3	1.2680	5.6012e-04	1.00	1.0102	1.6977e-03	1.00	0.7066	7.1966e-03	1.00
	(std.dev)	(0.0886)	(0.0017)		(0.0961)	(0.0027)		(0.0908)	(0.0064)	
	MOEADD	0.4411	4.9904e-02	1.00	0.4559	5.4874e-02	0.95	0.3935	5.9261e-02	0.97
	(std.dev)	(0.1139)	(0.0185)		(0.1489)	(0.0256)		(0.1181)	(0.0245)	
	NSGA-II	0.3088	7.7665e-02	1.00	0.3149	8.5321e-02	1.00	0.2898	9.5782e-02	1.00
	(std.dev)	(0.1316)	(0.0331)		(0.1322)	(0.0320)		(0.1473)	(0.0425)	
	NSGA-III	0.3918	5.8573e-02	1.00	0.3557	6.9857e-02	1.00	0.3938	6.2936e-02	1.00
	(std.dev)	(0.1192)	(0.0251)		(0.1003)	(0.0246)		(0.1315)	(0.0318)	

Table 7.6: Results for Eq-DTLZ 1-4 and Eq-IDTLZ 1-2 with $k = 4$ and $p = 2$ for some MOEAs. (# is average of the number of feasible solutions at the end of each run).

		200,000			300,000			500,000		
	Method	Δ_p	HV	#	Δ_p	HV	#	Δ_p	HV	#
Eq-DTLZ1	ANSGA-III	0.2735	4.7381e-04	1.00	0.2525	5.7108e-04	1.00	0.2653	4.1813e-04	1.00
	(std.dev)	(0.0450)	(0.0003)		(0.0600)	(0.0003)		(0.0457)	(0.0003)	
	GDE3	180.1942	0.0000e+00	0.01	182.5249	0.0000e+00	0.01	189.2532	0.0000e+00	0.01
	(std.dev)	(6.2677)	(0.0000)		(8.0290)	(0.0000)		(8.1336)	(0.0000)	
	MOEADD	0.2910	2.5956e-04	0.90	0.2768	3.4217e-04	0.93	0.5146	3.8035e-04	0.80
	(std.dev)	(0.0564)	(0.0002)		(0.0477)	(0.0003)		(1.4282)	(0.0003)	
	NSGA-II	0.2696	5.5364e-04	1.00	0.2452	6.4946e-04	1.00	0.2535	6.3457e-04	1.00
	(std.dev)	(0.0538)	(0.0003)		(0.0612)	(0.0004)		(0.0580)	(0.0003)	
	NSGA-III	0.2792	4.9979e-04	1.00	0.2777	4.3820e-04	1.00	0.2734	5.6532e-04	1.00
	(std.dev)	(0.0460)	(0.0003)		(0.0497)	(0.0004)		(0.0445)	(0.0003)	
Eq-DTLZ2	ANSGA-III	0.7553	2.1471e-02	1.00	0.7576	2.2727e-02	1.00	0.7131	2.9263e-02	1.00
	(std.dev)	(0.0788)	(0.0143)		(0.0794)	(0.0132)		(0.0861)	(0.0123)	
	GDE3	1.1851	0.0000e+00	0.00	1.1679	0.0000e+00	0.01	1.1407	7.0851e-05	0.02
	(std.dev)	(0.0442)	(0.0000)		(0.0383)	(0.0000)		(0.0356)	(0.0004)	
	MOEADD	0.6356	2.5679e-02	0.94	0.7221	1.7111e-02	0.80	0.6454	2.2654e-02	0.84
	(std.dev)	(0.2003)	(0.0164)		(0.1576)	(0.0175)		(0.2240)	(0.0165)	
	NSGA-II	0.7560	2.1120e-02	1.00	0.6882	3.0116e-02	1.00	0.6808	3.5500e-02	1.00
	(std.dev)	(0.0853)	(0.0154)		(0.0848)	(0.0150)		(0.0889)	(0.0185)	
	NSGA-III	0.7331	2.2257e-02	1.00	0.7033	2.7953e-02	1.00	0.7163	2.7012e-02	1.00
	(std.dev)	(0.0934)	(0.0158)		(0.0783)	(0.0120)		(0.0936)	(0.0141)	
Eq-DTLZ3	ANSGA-III	0.7467	2.0993e-02	1.00	0.7345	2.3463e-02	1.00	0.7340	2.1322e-02	1.00
	(std.dev)	(0.0498)	(0.0121)		(0.0781)	(0.0097)		(0.0934)	(0.0113)	
	GDE3	957.3515	0.0000e+00	0.00	986.8496	0.0000e+00	0.01	995.2061	0.0000e+00	0.02
	(std.dev)	(28.7975)	(0.0000)		(20.0831)	(0.0000)		(30.1458)	(0.0000)	
	MOEADD	2.7655	2.3234e-02	0.86	2.5255	1.5373e-02	0.88	0.6508	2.5170e-02	0.98
	(std.dev)	(8.7815)	(0.0200)		(6.7121)	(0.0140)		(0.2172)	(0.0180)	
	NSGA-II	0.7759	1.9365e-02	1.00	0.7060	2.8315e-02	1.00	0.7329	2.5670e-02	1.00
	(std.dev)	(0.0667)	(0.0130)		(0.1086)	(0.0149)		(0.0657)	(0.0175)	
	NSGA-III	0.7505	2.2988e-02	1.00	0.7567	1.8175e-02	1.00	0.7133	2.6541e-02	1.00
	(std.dev)	(0.0888)	(0.0138)		(0.0755)	(0.0121)		(0.0717)	(0.0110)	
Eq-DTLZ4	ANSGA-III	1.8168	0.0000e+00	0.63	1.3375	0.0000e+00	0.96	1.3693	0.0000e+00	0.92
	(std.dev)	(0.7684)	(0.0000)		(0.3442)	(0.0000)		(0.3941)	(0.0000)	
	GDE3	2.1729	0.0000e+00	0.00	2.1920	0.0000e+00	0.00	2.1578	0.0000e+00	0.00
	(std.dev)	(0.0472)	(0.0000)		(0.0437)	(0.0000)		(0.0523)	(0.0000)	
	MOEADD	1.2623	0.0000e+00	0.85	1.2706	0.0000e+00	0.83	1.2872	0.0000e+00	0.86
	(std.dev)	(0.0567)	(0.0000)		(0.0497)	(0.0000)		(0.0630)	(0.0000)	
	NSGA-II	1.3157	0.0000e+00	0.93	1.3791	0.0000e+00	0.89	1.3020	0.0000e+00	0.96
	(std.dev)	(0.2978)	(0.0000)		(0.3907)	(0.0000)		(0.2152)	(0.0000)	
	NSGA-III	1.4570	0.0000e+00	0.83	1.4508	0.0000e+00	0.89	1.3902	0.0000e+00	0.93
	(std.dev)	(0.5407)	(0.0000)		(0.5779)	(0.0000)		(0.3881)	(0.0000)	
Eq-IDTLZ1	ANSGA-III	0.2842	2.7608e-04	1.00	0.2713	3.0512e-04	1.00	0.2675	3.9467e-04	1.00
	(std.dev)	(0.0420)	(0.0002)		(0.0444)	(0.0002)		(0.0484)	(0.0002)	
	GDE3	394.4817	0.0000e+00	0.01	399.6722	0.0000e+00	0.01	412.7986	0.0000e+00	0.01
	(std.dev)	(11.5844)	(0.0000)		(14.0671)	(0.0000)		(15.1309)	(0.0000)	
	MOEADD	0.8120	2.5498e-04	0.99	1.1708	2.8837e-04	0.86	0.2920	2.8243e-04	0.93
	(std.dev)	(2.9878)	(0.0002)		(4.8514)	(0.0002)		(0.0449)	(0.0001)	
	NSGA-II	0.2739	2.8454e-04	1.00	0.2558	3.4889e-04	1.00	0.2653	3.5920e-04	1.00
	(std.dev)	(0.0420)	(0.0002)		(0.0548)	(0.0002)		(0.0463)	(0.0002)	
	NSGA-III	0.2923	2.7932e-04	1.00	0.2735	3.2622e-04	1.00	0.2673	3.3804e-04	1.00
	(std.dev)	(0.0328)	(0.0001)		(0.0517)	(0.0002)		(0.0431)	(0.0002)	
Eq-IDTLZ2	ANSGA-III	0.7368	1.2937e-02	1.00	0.7469	1.4338e-02	1.00	0.7135	1.8397e-02	1.00
	(std.dev)	(0.0780)	(0.0101)		(0.0822)	(0.0113)		(0.0782)	(0.0152)	
	GDE3	1.6088	0.0000e+00	0.00	1.5881	0.0000e+00	0.01	1.5498	0.0000e+00	0.02
	(std.dev)	(0.0641)	(0.0000)		(0.0549)	(0.0000)		(0.0554)	(0.0000)	
	MOEADD	0.7087	1.7435e-02	0.87	0.6984	2.0164e-02	0.80	0.6996	2.0251e-02	0.81
	(std.dev)	(0.1617)	(0.0134)		(0.1398)	(0.0163)		(0.1685)	(0.0160)	
	NSGA-II	0.7581	1.8413e-02	1.00	0.6833	1.6652e-02	1.00	0.6882	1.9565e-02	1.00
	(std.dev)	(0.0790)	(0.0115)		(0.0800)	(0.0140)		(0.0917)	(0.0154)	
	NSGA-III	0.7530	1.5853e-02	1.00	0.7289	1.4447e-02	1.00	0.7037	2.3990e-02	1.00
	(std.dev)	(0.0694)	(0.0136)		(0.0817)	(0.0141)		(0.1235)	(0.0135)	



(a) PS GDE3

(b) PS NSGA-II

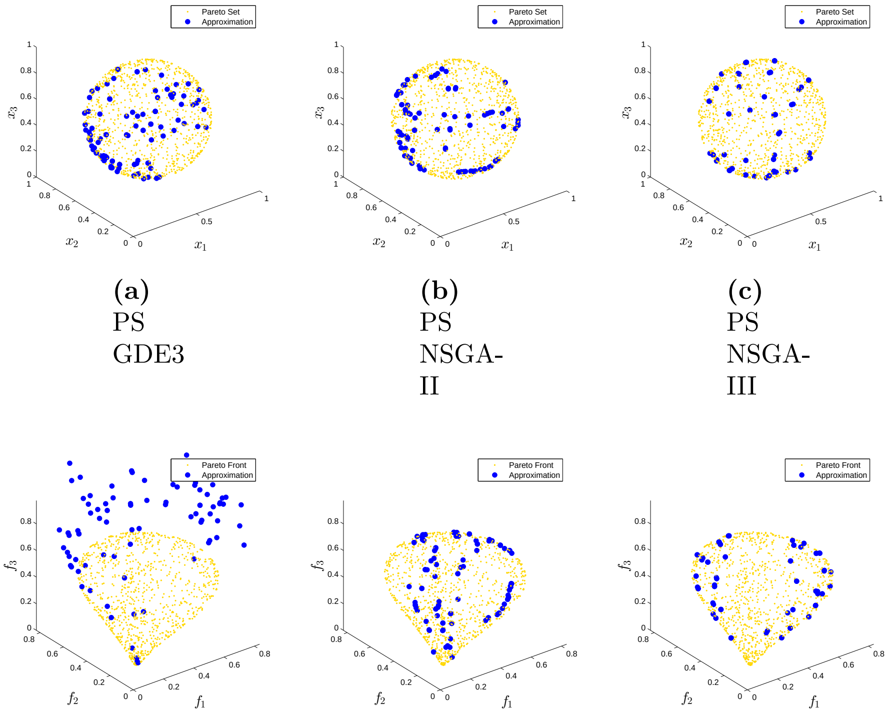
(c) PS NSGA-III

(d) PF GDE3

(e) PF NSGA-II

(f) PF NSGA-III

Figure 7.3: Results of the MOEAS on Eq-DTLZ1 and a budget of $150,000$ function evaluations.



(a) PS GDE3

(b) PS NSGA-II

(c) PS NSGA-III

(d) PF GDE3

(e) PF NSGA-II

(f) PF NSGA-III

Figure 7.4: Results of the MOEAs on Eq-DTLZ2 and a budget of $500,000$ function evaluations.

Chapter 8

Conclusions and Future Work

In this chapter, we first summarize the thesis work in Section 8.1. After that, we discuss our findings and contributions and point out their limitations in Section 8.2. Finally, in Section 8.3, we outline directions for possible future research.

8.1 Obtained Results

In this section, we briefly present the principal contributions of this work.

As described in Chapter 2, PE is a global/local tool for the treatment of MaOPs that consist of two phases. The first one is about obtaining a set of a few candidates –ideally optimal solutions, but a set of good approximations is enough– to perform a steering phase according to the preferences of the DM. Thus, the second phase is the local exploration following the Pareto landscapes (PS and PF) in a given direction. However, the PE framework was only able to deal with continuous MaOPs, and it was focused on the steering phase, mainly, with the steering in objective space, steering in decision variable space, and the steering in weight space.

First, we focused on the extension of the PE for continuous MaOPs. In this sense, we considered the scenario in which the DM is almost satisfied with a particular solution (which can be provided by one of the steering phases of PE or via a different method, e.g., a reference point method), then he/she wants to explore solutions around it. For dealing with this scenario, we proposed the Unbiased Neighborhood Exploration, which can perform the exploration in neighborhoods around a specific solution both in objective and decision spaces (Section 3.1). In this new approach of steering, we obtain a set of N elements evenly distributed around the desired solution. Here the DM can decide in which space to perform the steering and the number N of desired neighbors.

On the other hand, it is known that the DM usually wants to obtain the knee of the PF in many applications. Consequently, we presented a way to find the knee for continuous MaOPs using PE, which is equivalent to the mathematical definition of the knee. We provided the mathematical proof of this fact (Section 3.2).

To finish with the PE for continuous MaOPs, we demonstrated the effectiveness and usefulness of this method with the plastic injection molding process (Section 3.3). The use of applications is essential in the context of interactive methods because making fair comparisons is not always possible, due to the fact that each process requires different pieces of information. Moreover, comparisons of the PE against other continuation methods or MOEAs is unfair, as they try to approximate all the set of optimal solutions.

We extended the PE framework for the treatment of MaOPs with different smoothness assumptions. In particular, we proposed a fine tunning approach using MOEAs to deal with the steering in objective space for discrete problems, which we test with the well-known multi-objective knapsack problem (Section 4.1). We also apply this approach with linear MaOPs, but using the Simplex method as the local search strategy (Section 4.2).

It is also essential to provide the initial optimal solutions for the steering phases; such solutions must be global and representative of the PF, i.e., they have to cover the extension of the real PF, and they must be present at any disconnected component of the real PF. It is also desirable to obtain a reduced set of them with a reasonable computational effort. In this direction, we decided to hybridize PT with a MOEA with the described characteristics, which can also deal with equality constrained MOPs, as PT can handle this type of constraint. Notice that, in principle, we can use a set of solutions provided by any MOEA as initial points for the steering phases of PE. However, for the case of PT (where the goal is to approximate the entire optimal set), we need to provide a way to properly treat such solutions (Chapter 5).

Related with the exploration in decision space, we know that while so far, entirely a few suitable diversity mechanisms exist to obtain a spread in objective space, the consideration of the Pareto set approximations has been mainly neglected. This represents a possible shortcoming. To solve this, we proposed the variation rate, a heuristic to preserve diversity in decision space (Chapter 6). This topic is also important in the context of the exploration tools for the PE because the DM could be interested in the steering in decision space, which would require points in all the disconnected components of the Pareto set.

Finally, as we noticed the absence of an adequate benchmark for equality constrained MOPs, we proposed a new test suite for equality constrained MOPs (Chapter 7).

We will discuss the particular conclusions for each one of the above contributions in the next section.

8.2 Conclusions

Here, we discuss, based on the examples and and the numerical results of Chapters 3-7, the impact of this work.

In Section 3.1, we developed, as an extension to the continuous PE, a way to explore the neighborhood around a specific solution both in objective and decision space. The result of this algorithm is a set of N neighbors evenly distributed around the initial optimal solution that we demonstrate with some examples. Although making comparisons against this approach is very difficult, since no other method can do the same, we present a comparison using the rNSGA-II to show it. Numerical results show that our approach is very efficient; it spends only one function evaluation, with its corresponding Jacobian, to compute each neighbor (which is impossible to obtain with any MOEA). Moreover, all the graphs show the desired evenly distribution. Also, in Section 3.2, we mathematically demonstrate that the PE can find the knee for continuous MaOPs. Thus, we can conclude that our method works very well in continuous MaOPS.

In Section 3.3, we consider the many-objective design of a plastic injection molding process, which consists of seven objectives that all have a potentially significant impact on the decision-making process. We demonstrated on the case study of a particular plastic gear that the PF of related MOPs can be reliably computed via a continuation-like method and using a surrogate model. More precisely, we use PT for subproblems where we consider only 2 and 3 objectives; while, when considering the complete model ($k = 7$), none of the classical methods can be chosen any more due to the “curse of dimensionality”. As an alternative, we utilized the PE. For this, we investigated four different scenarios. It is conjectured from these results that the PE can serve as a powerful tool for the many-objective design of plastic injection molding.

In Section 4.1, we addressed a decision-making tool for discrete MaOPs, where we used the multidimensional multi-objective knapsack problem as a demonstrator. Here we proposed a local fine-tuning method that allows the search process to be steered from a given solution along the PF in a user-specified direction. More precisely, we presented a framework and two possible realizations of it: one by means of a GA for directly solving the dynamic reference point problem, and another one based on MOEA/D that focuses on a region of the Pareto front delimited by the reference point. Given that only a particular segment of the Pareto front is computed, one retrieves a much more accurate search efficiency compared against the classical method (i.e., aiming to compute all Pareto optimal solutions), which we have demonstrated on several benchmark problems. We think that this method can be used as a post-processing step to all existing many-objective optimization solvers and that this will actually help the DM to identify his/her most-preferred solution.

In Section 4.2, we adapted the framework presented in Section 4.1 for the numerical treatment of linear MaOPs. More precisely, we performed two approaches that were capable of following a preference-directed path to explore a large dimensional objective space locally. We conducted experiments for three and five objective benchmark problems. We stress that the method is in principle not restricted to moderate values of k as for each problem a sequence of SOPs is generated, and for each SOP, one optimal solution is obtained using the Simplex method.

In Chapter 5, we have proposed a two-phase hybrid algorithm combining a MOEA, based on the NSGA-II, and the PT. Numerical results and comparisons against four state-of-the-art MOEAs have shown that this new strategy is highly competitive and can lead to satisfying results with a moderate budget of function evaluations.

In Chapter 6, we dealt with the problem of conserving diversity in decision space without losing quality in objective space. To achieve this goal, we have first presented the general framework of the variation rate that combines the usage of the averaged distance in variable space with the selection operator that is given by the multi-objective evolutionary algorithm (MOEAs). We have also illustrated the possible integration of the variation rate into four MOEAs that represent the state-of-the-art. Numerical results have shown that the use of the variation rate improves the performance of the standalone algorithms for Type III problems, while the variation rate algorithms are not significantly worse for the standard benchmark problems, even in some cases variation rate improves the performance of the original algorithm.

Finally, in Chapter 7, we have first proposed a set of eight equality constrained MOPs for the benchmarking of MOEAs. All problems are based on the well-known and widely used DTLZ and IDTLZ test problems and inherit their valuable properties. Moreover, all problems are scalable in the number of decision variables and the number of objectives as well as in the number of equality constraints. The Pareto sets of all problems differ from the corresponding Pareto sets of the unconstrained counterparts and can be expressed analytically, which are two further important aspects for benchmarking. In a next step, we have shown the performance of some selected state-of-the-art MOEAs on the proposed Eq-(I)DTLZ problems. The computations have shown that the results of the different evolutionary algorithms are yet not satisfactory, as they present a poor distribution along the Pareto set/front.

8.3 Future Work

Though the results along this work are strongly promising, there are several lines of research arising from all our proposals that may be pursued in the future.

For instance, we know that the performance of the PE for continuous MaOPs depends, of course, of several factors (e.g., the step size and the initial optimal solution). For this reason, we need to provide more detailed numerical results to validate the proposals for the unbiased neighborhood exploration and the approach for finding the knee of the PF.

On the other hand, more research has to be performed in this direction to obtain a new class of hybrid evolutionary algorithms for the fast and reliable numerical treatment of general MOPs. For this task, it will be necessary to reduce the required derivative information. Further, more comparisons have to be performed to demonstrate the benefit of the novel hybrid. Finally, it is intended to apply this approach to real-world problems. For instance, for the plastic injection molding process, we can also extend the PE implementation to handle movements in directions in decision space and weight space to increase the set of alternatives given the preferences of the DM.

Related with the approaches for different smoothness assumptions, although this work can be considered as a proof-of-principle, there is still much to be done. First of all, the tuning of the method is an issue to guarantee to get better solutions in less time, and it may be interesting to consider other solvers. For instance, we intend to investigate other potential local search strategies according to the problem to be solved.

In the case of the variation rate, it will be mandatory to adapt some of the genetic operators of the evolutionary algorithms to exploit the diversity in decision space, as the variation rate is only a selection mechanism. In order to obtain optimal solutions, specifically in decision space, the exploration will have to be increased. Furthermore, it will be necessary to develop a specific indicator for problems of Type III. In general, performance indicators evaluate an approximation based on the value of the objectives, and this does not provide enough information for the distribution of the variables.

Finally, from the results with our benchmark, it can be said that the design of MOEAs for the fast and reliable treatment of equality constrained problems is undoubtedly a critical path for future research. We think that the Eq-(I)DTLZ benchmark suite can make a significant contribution to achieve this goal.

Appendix A

Appendix 1

In this appendix we provide the properties of the plastic gear used in the Plastic Injection Molding (PIM) problem (see Section 3.3).

A.1 A Plastic Gear

The properties of the material are listed in Table A.1. The process parameters, their ranges and units, utilized as design variables are listed in Table A.2. Tables A.3, A.4 and A.5 lists the rest of process parameters considered during the numerical simulation. Similarly, Table A.6 lists the outcomes of interest. Figure A.1 illustrates an example of the warpage scale in the simulation software.

Table A.1: Material properties of PP.

Density [g/cm^3]	1.35
Eject temperature [°C]	90
Thermal conductivity [erg/(sec cm °C)]	35000
Elastic modulus [dyne/cm^2]	3e+010
Poisson ratio	0.38
Heat capacity [erg/(g °C)]	1.5e+007
Melt temperature range [°C]	200–220
Mold temperature range [°C]	40–80

Table A.2: Design variables.

Process parameter	Design variable	Range
Melt temperature [°C]	x_1	190–230
Packing time [sec]	x_2	3–5
Packing pressure [MPa]	x_3	84–140
Cooling time [sec]	x_4	8–14

Table A.3: Process parameters.

Process parameter	Value
Filling time [sec]	0.10
Mold temperature [°C]	60.0
Maximum pressure machine [MPa]	140.00
Injection volume [cc]	1.89
VP switch by volume filled [%]	98.00
Mold opening time [sec]	5.00
Ejection temperature [°C]	90.0
Air temperature [°C]	25.0

Table A.4: Flow rate profile.

Section	Time [%]	Flow rate [%]
1	20	30
2	40	60
3	80	90
4	100	30

Table A.5: Packing rate profile.

Section	Time [%]	Flow rate [MPa]
1	50	x_3
2	80	70
3	100	35

Table A.6: Objective functions.

Outcome	Function	Optimization
Maximum warpage deformation [mm]	f_1	Minimize
Maximum volumetric shrinkage [%]	f_2	Minimize
Maximum Von Mises stress [MPa]	f_3	Minimize
Sink marks displacement [mm]	f_4	Minimize
Maximum claming force [ton]	f_5	Minimize
Cycle time [sec]	f_6	Minimize
High shear stress [%]	f_7	Minimize

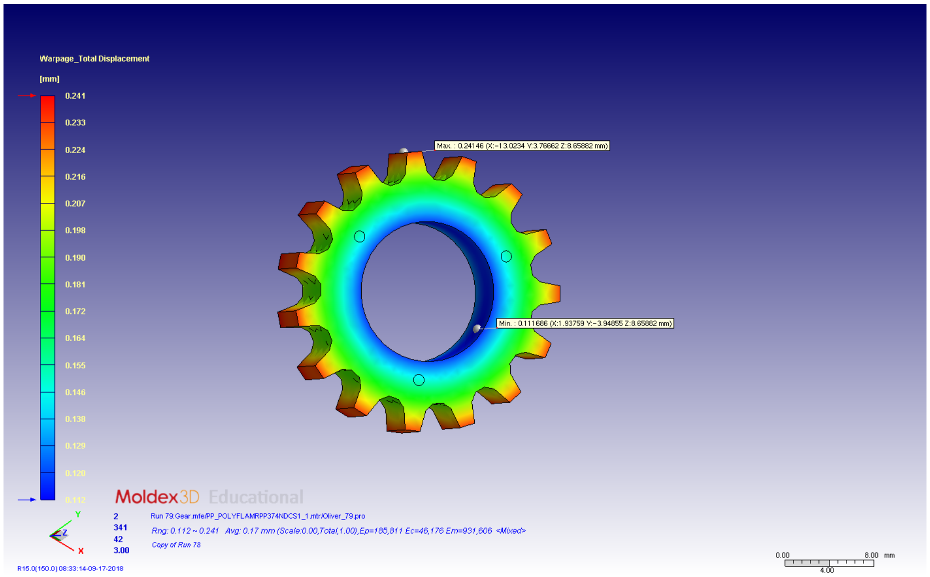

Warpage_Total Displacement
[mm]
0.241
0.233
0.224
0.216
0.207
0.198
0.190
0.181
0.172
0.164
0.155
0.146
0.138
0.129
0.120
0.112
Max. : 0.24146 (X:-13.0234 Y:3.76662 Z:8.65882 mm)
Min. : 0.111686 (X:1.93759 Y:-3.94855 Z:8.65882 mm)
Moldex3D Educational
Y
Z
X
2
341
42
3.00
Run 79:Gear.mfe/PP_POLYFLAMRPP374NDCS1_1.mtr/Oliver_79.pro
Rng: 0.112 ~ 0.241 Avg: 0.17 mm (Scale:0.00,Total,1.00),Ep=185,811 Ec=46,176 Em=931,606 <Mixed>
Copy of Run 78
R15.0(150.0) 08:33:14-09-17-2018
0.00
8.00 mm
4.00

Figure A.1: Warpage in the plastic part.

Bibliography

[Abhishek et al., 2017] Abhishek, K., Rakesh Kumar, V., Datta, S., and Mahapatra, S. S. (2017). Parametric appraisal and optimization in machining of CFRP composites by using TLBO (teaching–learning based optimization algorithm). *Journal of Intelligent Manufacturing*, 28(8):1769–1785.

[Aguirre and Tanaka, 2009] Aguirre, H. and Tanaka, K. (2009). Many-objective optimization by space partitioning and adaptive ε-ranking on mnk-landscapes. In *Evolutionary Multi-Criterion Optimization*, pages 407–422. Springer.

[Alvarado-Iniesta et al., 2019] Alvarado-Iniesta, A., Cuate, O., and Schütze, O. (2019). Multi-objective and many objective design of plastic injection molding process. *The International Journal of Advanced Manufacturing Technology*, 102(9):3165–3180.

[Alvarado-Iniesta et al., 2018] Alvarado-Iniesta, A., Guillen-Anaya, L. G., Rodríguez-Picón, L. A., and Ñeco-Caberta, R. (2018). Multi-objective optimization of an engine mount design by means of memetic genetic programming and a local exploration approach. *Journal of Intelligent Manufacturing.*

[Alves and Almeida, 2007] Alves, M. and Almeida, M. (2007). MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Computers & operations research*, 34(11):3458–3470.

[Bader et al., 2010] Bader, J., Deb, K., and Zitzler, E. (2010). Faster hypervolume-based search using monte carlo sampling. In *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, pages 313–326. Springer.

[Bader and Zitzler, 2011] Bader, J. and Zitzler, E. (2011). Hype: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76.

[Bakhtiari et al., 2016] Bakhtiari, H., Karimi, M., and Rezazadeh, S. (2016). Modeling, analysis and multi-objective optimization of twist extrusion process using predictive models and meta-heuristic approaches, based on finite element results. *Journal of Intelligent Manufacturing*, 27(2):463–473.

[Beume et al., 2007] Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669.

[Björck, 1996] Björck, A. (1996). *Numerical methods for least squares problems.* Siam.

[Bogoya et al., 2018] Bogoya, J. M., Vargas, A., Cuate, O., and Schütze, O. (2018). A (p,q)-averaged Hausdorff distance for arbitrary measurable sets. *Mathematical and Computational Applications*, 23(3).

[Bogoya et al., 2019] Bogoya, J. M., Vargas, A., and Schütze, O. (2019). The averaged Hausdorff distances in multi-objective optimization: A review. *Mathematics*, 7(10).

[Box and Draper, 1971] Box, M. J. and Draper, N. R. (1971). Factorial designs, the $|X'X|$ criterion, and some related matters. *Technometrics*, 13(4):731–742.

[Branke et al., 2008] Branke, J., Deb, K., Miettinen, K., and Slowinski, R. (2008). *Multiobjective optimization: Interactive and evolutionary approaches*, volume 5252. Springer Science & Business Media.

[Brockhoff et al., 2012] Brockhoff, D., Wagner, T., and Trautmann, H. (2012). On the properties of the R2 indicator. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 465–472, New York, NY, USA. ACM.

[Buchanan, 1997] Buchanan, J. T. (1997). A naive approach for solving MCDM problems: The guess method. *Journal of the Operational Research Society*, 48(2):202–206.

[Castillo et al., 2017] Castillo, J. C., Segura, C., Aguirre, A. H., Miranda, G., and León, C. (2017). A multi-objective decomposition-based evolutionary algorithm with enhanced variable space diversity control. In *Proceedings of GECCO 2017*, pages 1565–1571, New York, NY, USA. ACM.

[Coello and Cortés, 2005] Coello, C. A. C. and Cortés, N. C. (2005). Solving multi-objective optimization problems using an artificial immune system. *Genetic Programming and Evolvable Machines*, 6(2):163–190.

[Coello et al., 2007] Coello, C. A. C., Lamont, G. B., Van Veldhuizen, D. A., et al. (2007). *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer.

[Cuate et al., 2019] Cuate, O., Uribe, L., Ponsich, A., Lara, A., Beltran, F., Sánchez, A. R., and Schütze, O. (2019). A new hybrid metaheuristic for equality constrained bi-objective optimization problems. In Deb, K., Goodman, E., Coello Coello, C. A., Klamroth, K., Miettinen, K., Mostaghim, S., and Reed, P., editors, *Evolutionary*

Multi-Criterion Optimization, pages 53–65, Cham. Springer International Publishing.

[D'Addona et al., 2017] D'Addona, D. M., Ullah, A. M. M. S., and Matarazzo, D. (2017). Tool-wear prediction and pattern-recognition using artificial neural network and dna-based computing. *Journal of Intelligent Manufacturing*, 28(6):1285–1301.

[Das, 1999] Das, I. (1999). On characterizing the "knee" of the Pareto curve based on normal-boundary intersection. *Structural optimization*, 18(2):107–115.

[Das and Dennis, 1997] Das, I. and Dennis, J. E. (1997). A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. *Structural optimization*, 14(1):63–69.

[Das and Dennis, 1998] Das, I. and Dennis, J. E. (1998). *Normal-boundary intersection*: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J of Opt*, 8(3):631–657.

[Deb, 2001] Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, volume 16. John Wiley & Sons.

[Deb and Jain, 2014] Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans Evol Comp*, 18(4):577–601.

[Deb et al., 2002] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197.

[Deb et al., 2001] Deb, K., Pratap, A., and Meyarivan, T. (2001). Constrained test problems for multi-objective evolutionary optimization. In *International conference on evolutionary multi-criterion optimization*, pages 284–298. Springer.

[Deb et al., 2006] Deb, K., Sinha, A., and Kukkonen, S. (2006). Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1141–1148. ACM.

[Deb and Sundar, 2006] Deb, K. and Sundar, J. (2006). Reference point based multi-objective optimization using evolutionary algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 635–642.

[Deb et al., 2005] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In *Evolutionary multiobjective optimization*, pages 105–145. Springer.

[Deb and Tiwari, 2008] Deb, K. and Tiwari, S. (2008). Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3):1062 – 1087.

[Dellnitz et al., 2005] Dellnitz, M., Schütze, O., and Hestermeyer, T. (2005). Covering Pareto sets by multilevel subdivision techniques. *Journal of Optimization Theory and Applications*, 124(1):113–155.

[Dennis and Moré, 1977] Dennis, Jr, J. E. and Moré, J. J. (1977). Quasi-Newton methods, motivation and theory. *SIAM review*, 19(1):46–89.

[Di Pierro et al., 2007] Di Pierro, F., Khu, S.-T., Savic, D., et al. (2007). An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *Evolutionary Computation, IEEE Transactions on*, 11(1):17–45.

[Dilettoso et al., 2017] Dilettoso, E., Rizzo, S. A., and Salerno, N. (2017). A weakly Pareto compliant quality indicator. *Mathematical and Computational Applications*, 22(1).

[Edgeworth, 1881] Edgeworth, F. Y. (1881). *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. Number 10. CK Paul.

[Eskelinen et al., 2010] Eskelinen, P., Miettinen, K., Klamroth, K., and Hakanen, J. (2010). Pareto navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, 32(1):211–227.

[Fan et al., 2019a] Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., and Goodman, E. (2019a). Difficulty adjustable and scalable constrained multiobjective test problem toolkit. *Evolutionary Computation*, pages 1–28.

[Fan et al., 2019b] Fan, Z., Li, W., Cai, X., Li, H., Wei, C., Zhang, Q., Deb, K., and Goodman, E. (2019b). Push and pull search for solving constrained multi-objective optimization problems. *Swarm and Evolutionary Computation*, 44:665 – 679.

[Farina and Amato, 2002] Farina, M. and Amato, P. (2002). On the optimal solution definition for many-criteria optimization problems. In *Proceedings of the NAFIPS-FLINT international conference*, pages 233–238.

[Fliege et al., 2009] Fliege, J., Drummond, L. G., and Svaiter, B. F. (2009). Newton's method for multiobjective optimization. *SIAM Journal on Optimization*, 20(2):602–626.

[Gass and Saaty, 1955] Gass, S. and Saaty, T. (1955). The computational algorithm for the parametric objective function. *Naval Research Logistics Quarterly*, 2(1):39–45.

[Gibbons and Chakraborti, 2011] Gibbons, J. D. and Chakraborti, S. (2011). *Nonparametric statistical inference*. Springer.

[Griewank and Corliss, 1992] Griewank, A. and Corliss, G. F. (1992). *Automatic differentiation of algorithms: theory, implementation, and application.* Defense Technical Information Center.

[Hansen and Jaszkiewicz, 1994] Hansen, M. P. and Jaszkiewicz, A. (1994). *Evaluating the quality of approximations to the non-dominated set.* IMM, Department of Mathematical Modelling, Technical Universityof Denmark.

[Hillermeier, 2001] Hillermeier, C. (2001). *Nonlinear multiobjective optimization: A generalized homotopy approach*, volume 135. Springer.

[Huband et al., 2005] Huband, S., Barone, L., While, L., and Hingston, P. (2005). A scalable multi-objective test problem toolkit. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 280–295. Springer.

[Huband et al., 2006] Huband, S., Hingston, P., Barone, L., and While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506.

[Ishibuchi et al., 2009] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. (2009). Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. In *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE.

[Ishibuchi et al., 2017] Ishibuchi, H., Setoguchi, Y., Masuda, H., and Nojima, Y. (2017). Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190.

[Jain and Deb, 2014] Jain, H. and Deb, K. (2014). An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622.

[Jan and Khanum, 2013] Jan, M. A. and Khanum, R. A. (2013). A study of two penalty-parameterless constraint handling techniques in the framework of moea/d. *Applied Soft Computing*, 13(1):128 – 148.

[Jan and Zhang, 2010] Jan, M. A. and Zhang, Q. (2010). MOEA/D for constrained multiobjective optimization: Some preliminary experimental results. In *2010 UK Workshop on Computational Intelligence (UKCI)*, pages 1–6.

[Jaszkiewicz and Słowiński, 1999] Jaszkiewicz, A. and Słowiński, R. (1999). The 'Light Beam Search' approach an overview of methodology applications. *European Journal of Operational Research*, 113(2):300–314.

[Jeffrey et al., 1993] Jeffrey, H., Nafpliotis, N., and Goldberg, D. E. (1993). Multiobjective optimization using the niched Pareto genetic algorithm. *IlliGAL report*, (93005):61801–2296.

[Karush, 1939] Karush, W. (1939). *Minima of functions of several variables with inequalities as side constraints*. PhD thesis, Master's thesis, Dept. of Mathematics, Univ. of Chicago.

[Kitayama et al., 2017] Kitayama, S., Miyakawa, H., Takano, M., and Aiba, S. (2017). Multi-objective optimization of injection molding process parameters for short cycle time and warpage reduction using conformal cooling channel. *The International Journal of Advanced Manufacturing Technology*, 88(5):1735–1744.

[Kitayama and Natsume, 2014] Kitayama, S. and Natsume, S. (2014). Multiobjective optimization of volume shrinkage and clamping force for plastic injection molding via sequential approximate optimization. *Simulation Modelling Practice and Theory*, 48:35 – 44.

[Krantz and Parks, 2002] Krantz, S. G. and Parks, H. R. (2002). *The implicit function theorem: History, theory, and applications*. Springer.

[Kuhn and Tucker, 1951] Kuhn, H. W. and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492, Berkeley and Los Angeles. University of California Press.

[Kukkonen and Lampinen, 2005] Kukkonen, S. and Lampinen, J. (2005). GDE3: The third evolution step of generalized differential evolution. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 443–450. IEEE.

[Lawson and Hanson, 1995] Lawson, C. L. and Hanson, R. J. (1995). *Solving least squares problems*, volume 15. Siam.

[Li and Zhang, 2009] Li, H. and Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.

[Li et al., 2015] Li, K., Deb, K., Zhang, Q., and Kwong, S. (2015). An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans. Evolutionary Computation*, 19(5):694–716.

[Löhne and Weißing,] Löhne, A. and Weißing, B. Bensolve-vlp solver, version 2.0. 1. *URL http://bensolve. org*.

[Martín and Schütze, 2014] Martín, A. and Schütze, O. (2014). A new predictor corrector variant for unconstrained bi-objective optimization problems. In *EVOLVE-A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation V*, pages 165–179. Springer.

[Martín and Schütze, 2018] Martín, A. and Schütze, O. (2018). Pareto tracer: A predictor–corrector method for multi-objective optimization problems. *Engineering Optimization*, 50(3):516–536.

[Martinez and Coello, 2014] Martinez, S. Z. and Coello, C. A. C. (2014). A multi-objective evolutionary algorithm based on decomposition for constrained multi-objective optimization. In *2014 IEEE Congress on Evolutionary Computation (CEC)*, pages 429–436.

[Mckay et al., 2000] Mckay, M. D., Beckman, R. J., and Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 42(1):55–61.

[Mejía et al., 2017] Mejía, J. A. H., Schütze, O., Cuate, O., Lara, A., and Deb, K. (2017). RDS-NSGA-II: a memetic algorithm for reference point based multi-objective optimization. *Engineering Optimization*, 49(5):828–845.

[Miettinen, 1999] Miettinen, K. (1999). *Nonlinear Multiobjective Optimization, volume 12 of International Series in Operations Research and Management Science.* Kluwer Academic Publishers, Dordrecht.

[Miettinen et al., 2010] Miettinen, K., Eskelinen, P., Ruiz, F., and Luque, M. (2010). Nautilus method: An interactive technique in multiobjective optimization based on the nadir point. *European Journal of Operational Research*, 206(2):426–434.

[Miettinen and Mäkelä, 2000] Miettinen, K. and Mäkelä, M. M. (2000). Interactive multiobjective optimization system www-nimbus on the internet. *Computers & Operations Research*, 27(7):709–723.

[Miettinen and Mäkelä, 2002] Miettinen, K. and Mäkelä, M. M. (2002). On scalarizing functions in multiobjective optimization. *OR spectrum*, 24(2):193–213.

[Nie and Ellingwood, 2004] Nie, J. and Ellingwood, B. R. (2004). A new directional simulation method for system reliability. part i: application of deterministic point sets. *Prob Eng Mech*, 19(4):425–436.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006). *Numerical optimization.* Springer Series in Operations Research and Financial Engineering. Springer.

[Pareto, 1896] Pareto, V. (1896). *Cours D'économie politique.* Lausanne, F. Rouge; Paris, Pichon.

[Rachmawati and Srinivasan, 2006] Rachmawati, L. and Srinivasan, D. (2006). A multi-objective evolutionary algorithm with weighted-sum niching for convergence on knee regions. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pages 749–750.

[Rudolph et al., 2007] Rudolph, G., Naujoks, B., and Preuss, M. (2007). Capabilities of EMOA to detect and preserve equivalent Pareto subsets. In *EMO 2017*, pages 36–50, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Rudolph et al., 2016] Rudolph, G., Schütze, O., Grimme, C., Domínguez-Medina, C., and Trautmann, H. (2016). Optimal averaged Hausdorff archives for bi-objective problems: theoretical and numerical results. *Computational Optimization and Applications*, 64(2):589–618.

[Saha and Ray, 2012] Saha, A. and Ray, T. (2012). Equality constrained multi-objective optimization. In *2012 IEEE Congress on Evolutionary Computation, CEC 2012*, pages 1–7.

[Saxena et al., 2013] Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., and Zhang, Q. (2013). Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *Evolutionary Computation, IEEE Transactions on*, 17(1):77–99.

[Schütze et al., 2019] Schütze, O., Cuate, O., Martín, A., Peitz, S., and Dellnitz, M. (2019). Pareto explorer: a global/local exploration tool for many-objective optimization problems. *Engineering Optimization*, 0(0):1–24.

[Schütze et al., 2012] Schütze, O., Esquivel, X., Lara, A., and Coello, C. A. C. (2012). Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522.

[Schütze et al., 2011] Schütze, O., Lara, A., and Coello Coello, C. (2011). The directed search method for unconstrained multi-objective optimization problems. *Proceedings of the EVOLVE–A Bridge Between Probability, Set Oriented Numerics, and Evolutionary Computation.*

[Schütze et al., 2013] Schütze, O., Witting, K., Ober-Blöbaum, S., and Dellnitz, M. (2013). *Set Oriented Methods for the Numerical Treatment of Multiobjective Optimization Problems*, pages 187–219. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Singh and Ray, 2011] Singh, H. I., A. and Ray, T. (2011). A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems. *IEEE Trans on Evol Comp*, 15(4):539–556.

[Srinivas and Deb, 1994] Srinivas, N. and Deb, K. (1994). Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248.

[Steuer and Choo, 1983] Steuer, R. E. and Choo, E.-U. (1983). An interactive weighted tchebycheff procedure for multiple objective programming. *Mathematical programming*, 26(3):326–344.

[Sun et al., 2018] Sun, J.-Q., Xiong, F.-R., Schütze, O., and Hernández, C. (2018). *Cell Mapping Methods*. Springer.

[Takahama and Sakai, 2006] Takahama, T. and Sakai, S. (2006). Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites. In *2006 IEEE International Conference on Evolutionary Computation*, pages 1–8.

[Tian et al., 2017] Tian, Y., Cheng, R., Zhang, X., and Jin, Y. (2017). Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12(4):73–87.

[Van Veldhuizen, 1999] Van Veldhuizen, D. A. (1999). Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, Air Force Institute of Technology.

[von Lücken et al., 2014] von Lücken, C., Barán, B., and Brizuela, C. (2014). A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*, 58(3):707–756.

[Wierzbicki, 1981] Wierzbicki, A. P. (1981). *A mathematical basis for satisficing decision making*. Springer.

[Yang et al., 2013] Yang, S., Li, M., Liu, X., and Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *Evolutionary Computation, IEEE Transactions on*, 17(5):721–736.

[Zhang and Li, 2007] Zhang, Q. and Li, H. (2007). MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans on Evol Comp*, 11(6):712–731.

[Zhang et al., 2008] Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (2008). Multiobjective optimization test instances for the cec 2009 special session and competition. *University of Essex,UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, 264.

[Zitzler et al., 2007] Zitzler, E., Brockhoff, D., and Thiele, L. (2007). The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 862–876. Springer.

[Zitzler et al., 2008] Zitzler, E., Knowles, J., and Thiele, L. (2008). *Quality Assessment of Pareto Set Approximations*, pages 373–404. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Zitzler and Thiele, 1999] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271.

[Zou et al., 2008] Zou, X., Chen, Y., Liu, M., and Kang, L. (2008). A new evolutionary algorithm for solving many-objective optimization problems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 38(5):1402–1412.