Center for Research and Advanced Studies of the
National Polytechnic Institute of Mexico

Zacatenco Campus
Computer Science Department

Techniques to Deal with Many-objective Optimization Problems Using Evolutionary Algorithms

by

Antonio López Jaimes

as the fulfillment of the requirement for the degree of

Ph.D. in Computer Science

Advisor: **Dr. Carlos Artemio Coello Coello**

Mexico City May 2011

Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional

Unidad Zacatenco
Departamento de Computación

Técnicas para resolver problemas de optimización con muchas funciones objetivo usando algoritmos evolutivos

Tesis que presenta

Antonio López Jaimes

para obtener el grado de

Doctor en Ciencias en Computación

Director de tesis: **Dr. Carlos Artemio Coello Coello**

México, D.F. Mayo de 2011

“Chiedi al rio perché gemente
dalla balza ov’ebbe vita
corre al mar, che a sé l’invita
e nel mar sen va a morir...”

Dedico esta tesis a Liz, que hizo realidad el elixir.

ABSTRACT

Many problems in engineering, industry, and in many other fields, involve the simultaneous optimization of several objectives. These types of problems are called Multiobjective Optimization Problems (MOPs). A distinctive characteristic of a MOP is that its objectives have some degree of conflict among them (i.e., one objective cannot be improved without deterioration of at least any other objective). While in single-objective optimization a single (global) optimal solution is aimed for, in multiobjective optimization, a set of alternatives with different trade-offs among the objectives is usually achieved. The method most commonly adopted in multiobjective optimization to compare solutions is the *Pareto dominance relation*. Hence, optimal solutions are called *Pareto optimal solutions*. The evaluation of these solutions using the objective functions is collectively known as *Pareto optimal front*.

One of the most successful approaches for solving MOPs is the use of Multiobjective Evolutionary Algorithms (MOEAs), which are stochastic search and optimization methods that simulate the natural evolution process. In many cases, MOEAs have been applied to MOPs with 2 or 3 objectives. Nonetheless, recent experimental and analytical studies have shown that the effectiveness of Pareto-based MOEAs is deteriorated as the number of objectives increases. A widely accepted explanation for this deterioration is that the proportion of equivalent solutions, in terms of the Pareto dominance relation, quickly increases with the number of objectives. Another difficulty is that the number of points to approximate a Pareto front usually increases exponentially with the number of objectives.

In this thesis we present several techniques to remedy some difficulties to solve MOPs with a high number of objectives (*many-objective problems*). The proposed techniques can be classified in two main classes: *i*) reduction of the number of objectives of the problem during the search or, *a posteriori*, and *ii*) new preference relations to order Pareto-equivalent solutions. First, we proposed two algorithms to reduce the number of objectives. The underlying idea of these objective reduction algorithms is to identify the nonconflicting objectives in order to discard them. Based on experimental evidence, we can say that our techniques outperformed similar objective reduction algorithms. Further, they have a lower time complexity. Later on, we incorporated an objective reduction algorithm into a MOEA in order to approximate the entire Pareto front. From this proposal we can conclude that reducing the number of objectives during the search improves the scalability of MOEA in terms of the number of objectives. Another important finding is that simultaneously

searching in different objective subsets also improves the search ability of a MOEA.

We also develop a new preference relation based on a reference point approach. This relation offers an easy way to integrate decision maker's preferences into a MOEA without modifying its basic structure. Additionally, the proposed reference relation was used to deal with many-objective problems. The experimental results indicate that the proposed relation is less affected by an increase in the number of objectives than Pareto dominance.

RESUMEN

Muchos problemas en ingeniería, en la industria, y en muchas otras áreas, plantean la optimización simultánea de varios objetivos. Problemas de este tipo son conocidos como problemas de optimización multiobjetivo (MOPs). Una característica distintiva de un MOP es que entre sus objetivos existe cierto grado de conflicto (i.e., un objetivo no puede ser mejorado sin deteriorar al menos otro objetivo). Mientras que en optimización mono-objetivo se busca una única solución óptima (global), en la optimización multiobjetivo, usualmente se obtiene un conjunto de soluciones alternativas que representan diferentes niveles de compromiso entre los objetivos. El método que comúnmente se utiliza en optimización multiobjetivo para comparar soluciones es la relación de dominancia de Pareto. De aquí que las soluciones óptimas sean llamadas soluciones óptimas de Pareto. La evaluación de estas soluciones a través de las funciones objetivo son denominadas frente de Pareto.

Uno de los enfoques más exitosos para resolver MOPs es el uso de los algoritmos evolutivos multiobjetivo (MOEAs), los cuales son métodos estocásticos de búsqueda y optimización que simulan el proceso de evolución natural. En muchos casos, los MOEAs han sido aplicados a MOPs con 2 ó 3 objetivos. No obstante, estudios experimentales y analíticos recientes han mostrado que la efectividad de los MOEAs basados en la dominancia de Pareto se deteriora conforme el número de objetivos aumenta. Una explicación comúnmente aceptada para este deterioro es que la proporción de soluciones equivalentes, en términos de la dominancia de Pareto, se incrementa rápidamente con el número de objetivos. Otra dificultad es que el número de puntos para aproximar un frente de Pareto usualmente se incrementa exponencialmente con el número de objetivos.

Nosotros presentamos varias técnicas para contrarrestar algunas dificultades planteadas por los MOPs con muchos objetivos. Las técnicas propuestas se pueden clasificar en dos clases: *i*) reducción del número de objetivos del problema durante la búsqueda o *a posteriori*, y *ii*) utilización de nuevas relaciones de preferencia para jerarquizar soluciones equivalentes en cuanto a la dominancia de Pareto.

Inicialmente propusimos dos algoritmos para reducir el número de objetivos. La idea básica de estos algoritmos es identificar los objetivos no conflictivos para descartarlos. Basados en la evidencia experimental, podemos decir que nuestras técnicas superan a otros algoritmos de reducción similares previamente propuestos. Además, nuestros algoritmos tienen una

complejidad de tiempo menor. Más adelante, incorporamos uno de nuestros algoritmos de reducción en un MOEA con el fin aproximar el frente de Pareto. De esta propuesta podemos concluir que reducir el número de objetivos durante la búsqueda mejora la escalabilidad del MOEA con respecto al número de objetivos. Otro descubrimiento importante es que explorando simultáneamente en diferentes subconjuntos de objetivos también mejora la capacidad de búqueda de un MOEA.

Finalmente, desarrollamos una nueva relación de preferencia basada en un enfoque de punto de referencia. Esta relación ofrece una manera sencilla de integrar preferencias del tomador de decisiones en un MOEA sin modificar su estructura básica. Además, la relación de preferencia propuesta fue usada para tratar con problemas con muchos objetivos. Los resultados experimentales indican que la relación es menos afectada al incrementar el número de objetivos que la dominancia de Pareto.

AGRADECIMIENTOS

Estoy seguro que durante estos últimos cuatro años y medio he aprendido más que en cualquier otro ciclo semejante de mi vida. Es cierto, claro, que he aprendido una profesión, pero también he vivido tantas experiencias que, en balance, me han mejorado personalmente. Por eso me parece justo aprovechar estas líneas para agradecer de manera pública a tantas personas de las que he aprendido a lo largo de estos años. Antes de continuar, debo disculparme con las personas que, por olvido, no lean su nombre más abajo.

En primer lugar, agradezco a mi asesor, el Dr. Carlos A. Coello Coello por enseñarme muy bien la profesión de investigador. Aprecio sobre todo que, al igual que a otros compañeros, me haya ayudado a sentirme parte de la comunidad de nuestro campo de investigación. Cuando vamos a los congresos tengo la confianza que mi trabajo es tan valioso como el de los investigadores de renombre que admiramos.

Antes de que se me olvide, cabe mencionar que el trabajo reportado en esta tesis se derivó del proyecto CONACYT número 103570. Ya está.

De manera muy especial quiero agradecer a mis amigos que conocí en el CINVESTAV: Alfredo Arias, Adriana Lara, María de Lourdes López, Cuauhtemoc Mancillas, Eduardo Vázquez y Saúl Zapotecas[1]. Cuando llegué por primera vez al centro, el edificio del departamento no me gustó, me pareció incluso un poco lúgubre. Ahora no importa, disfruto estar allí por mis amigos, no importa que los muebles sean viejos, que los espacios tengan poca luz, que viaje hora y media para llegar allá, o que haga tanto calor mientras escribo esto. Agradezco a mis amigos la atención que ponían —y sin reírse— cuando tantas veces les traté de explicar en el pizarrón ideas que aún no tenían forma siquiera. Gracias a Lulú por sus consejos.

Durante mi estancia en Japón encontré a muy buenas personas. Agradezco a los profesores Hernán Aguirre y Kiyoshi Tanaka por su hospitalidad cuando estuve en la Universidad de Shinshu en Nagano, Japón. También agradezco a los muchachos del laboratorio, especialmente a Yuta Tamura y Yoshiki Tanaka.

Quiero agradecer también a las secretarías del departamento de Computación, Felipa Rosas, Erica Ríos, y de manera especial a Sofi, por ayudarme más allá de las responsabilidades de su trabajo.

1 El orden es estrictamente alfabético ;)

Agradezco también a los doctores Debrup Chakraborty, Luis Gerardo de la Fraga, Katya Rodríguez Vázquez y Arturo Hernández Aguirre por aceptar revisar esta tesis a pesar de la premura.

Desde luego agradezco a mis padres por darme la libertad y la motivación para llegar hasta este punto de mi carrera.

No quiero terminar sin antes agradecer de todo corazón a mi novia Liz por acompañarme y darme todo el apoyo y cariño que me dieron la fuerza para terminar esta etapa de mi vida. ¡Liz, vamos por otra!

CONTENTS

1 INTRODUCTION 1
- 1.1 Problem Statement 4
- 1.2 General and Specific Goals of the Thesis 5
  - 1.2.1 Main Goal 5
  - 1.2.2 Specific Goals and Contributions of the Thesis 5
- 1.3 Structure of the Thesis 7

2 MULTIOBJECTIVE OPTIMIZATION PROBLEMS 9
- 2.1 Notions of Optimality in a MOP 10
- 2.2 Approaches to Solve a MOP 14

3 OPTIMIZATION METHODS TO SOLVE MOPS 17
- 3.1 Mathematical Programming Methods 17
  - 3.1.1 A Priori Preference Articulation 17
  - 3.1.2 A Posteriori Preference Articulation 18
  - 3.1.3 Interactive Preference Articulation 19
- 3.2 Evolutionary Multiobjective Algorithms 20
  - 3.2.1 Key Elements of a MOEA 22
  - 3.2.2 MOEAs based on Pareto Optimality 26

4 MANY-OBJECTIVE OPTIMIZATION 27
- 4.1 Notions of Conflict Among Objectives 27
- 4.2 Sources of Difficulty When Solving Many-Objective Problems 29
  - 4.2.1 Deterioration of the Search Ability 29
  - 4.2.2 Dimensionality of the Pareto front 31
  - 4.2.3 Visualization of the Pareto front 32
- 4.3 Current Approaches to Deal with Many-Objective Problems 32
  - 4.3.1 Preference Relations to Deal with Many-Objective Problems 33
  - 4.3.2 Objective Reduction Approaches 34
  - 4.3.3 Preference Incorporation Approaches 36

5 OBJECTIVE REDUCTION TO DEAL WITH MANY-OBJECTIVE PROBLEMS 39
- 5.1 Proposed Objective Reduction Algorithms 39
- 5.2 Comparative Study 45
  - 5.2.1 Evaluation of MOSSA 45
  - 5.2.2 Evaluation of KOSSA 46
  - 5.2.3 Final Remarks 54

6 ONLINE OBJECTIVE REDUCTION 57

6.1 Gradual Reduction of the Objectives During the Search 57
6.2 Assessment of the Objective Reduction Schemes Coupled to a MOEA 60
6.2.1 Overall Assessment of the Reduction Schemes 62
6.2.2 Effect of the Reduction Schemes on a MOEA's Search Ability 64
6.2.3 Final Remarks 66
6.3 The Conflict-Based Partitioning Framework 67
6.3.1 General Idea of the Partitioning Framework 68
6.3.2 A New Partition Strategy 69
6.3.3 Partitioning Using Conflict Information 71
6.4 Experimental Results 74
6.4.1 Algorithms and Parameter Settings Employed 74
6.4.2 Test Problems Employed 74
6.4.3 Quality Indicators Employed 75
6.4.4 Problems With Conflict Known A Priori 76
6.4.5 Effect of the Size of the Subspaces 81
6.4.6 Problems With Unknown Conflict 84
6.4.7 Final Remarks 85
7 STUDY OF PREFERENCE RELATIONS IN MANY-OBJECTIVE OPTIMIZATION 91
7.1 Quantitative Analysis of the Preference Relations 92
7.1.1 Quality Indicators and Methods Employed 92
7.1.2 Experimental Settings 94
7.1.3 Analysis of the Expansion Preference Relation 95
7.1.4 Analysis of All the Preference Relations 95
7.2 Final Remarks 104
8 A NEW PREFERENCE RELATION TO DEAL WITH MANY-OBJECTIVE PROBLEMS 107
8.1 The Reference Point Approach and the Achievement Scalarizing Function 108
8.1.1 MOEAs Based On a Scalarizing Function 110
8.2 Chebyshev Relation to Guide the Search 111
8.2.1 User Reference Point Chebyshev Preference Relation 111
8.2.2 Central-guided Chebyshev Preference Relation 115
8.3 An Interactive Method Using the Chebyshev Relation 117
8.4 Evaluation of the Interactive Method 119
8.4.1 Airfoil Shape Problem with 2 Objectives 119
8.4.2 Airfoil Shape Problem with 3 Objectives 123
8.4.3 Airfoil Shape Problem with 6 Objectives 124
8.5 Approximating the Pareto Front with the Chebyshev Relation 126

8.6 Evaluation of the Chebyshev Relation for Approximating the Entire Pareto Front 128
8.6.1 Test Problems Employed 129
8.6.2 Quality Indicators Employed 129
8.6.3 Assessing Convergence 130
8.6.4 Dominance Resistant Solutions in DTLZ Problems 133
8.6.5 Evaluation using WFG6 143
8.7 Final Remarks 145
9 CONCLUSIONS AND FUTURE WORK 147
9.1 Conclusions 147
9.1.1 Objective Reduction During the Search or After It 147
9.1.2 New preference relations 149
9.2 Future Work 150

BIBLIOGRAPHY 151

LIST OF FIGURES

Figure 1 Set of alternative solutions for the optimization example of a chopper motorcycle frame. 1

Figure 2 Search spaces in multiobjective optimization problems. 10

Figure 3 Illustration of the concept of Pareto dominance relation. 11

Figure 4 Illustration of the Pareto optimal set and its image, the Pareto front. 12

Figure 5 Schemes to implement elitism. 24

Figure 6 Hypergrid to maintain diversity in the archive. In this figure we can see how the hypergrid changes during the search. 25

Figure 7 Clustering technique to maintain diversity in the archive. 25

Figure 8 Two objective functions can be in conflict in some subsets of the feasible space, and can be supportive in other subsets. 28

Figure 9 Nondominated regions with respect to a given solution z. 30

Figure 10 Illustration of some Dominance Resistant Solutions in problem DTLZ2. 30

Figure 11 Number of points required to represent a Pareto front with a resolution r, i.e., the number of hypercubes per dimension. 31

Figure 12 Basic strategy of the objective reduction method employed. 41

Figure 13 Pseudocode of the proposed objective reduction algorithm MOSSA. 42

Figure 14 Pseudocode of the objective reduction algorithm KOSSA. 43

Figure 15 Running times of the three objective reduction algorithms. The upper-left plot shows the running times of the three algorithms using $k = k/2$, and $m = 240$, and, for the DS algorithm, 50 generations for Nondominated Sorting Genetic Algorithm II (NSGA-II). The other plots show the running times for the worst ($k = k-1, 2$), median ($k = k/2$) and best ($k = 2, k-1$) cases for the BZ algorithm and KOSSA, respectively. 45

Figure 16 D distribution on the problem DTLZ2_{BZ} for different number of objectives. $D = 1$ corresponds to the true Pareto front. 66

Figure 17 Non-dominated sorting and trucation based on different subspaces of a partition $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_S}\}$. 69

Figure 18 Alternation between the partition and the entire objective space. 71

Figure 19 Basic strategy to create subspaces using conflict information. 73

Figure 20 Subspaces generated using the conflict and random partition strategies on problem $\text{DTLZ5}(I = 4, M = 8)$. Objectives 6-8 and any of the other objectives are the conflicting objectives. 77

Figure 21 Generational distance results in problem $\text{DTLZ5}(I = 4, M)$. From 4-9 objectives it was generated a partition with 2 subspaces, while for 10-15 objectives it was generated one with 3 subspaces. 78

Figure 22 $\text{DTLZ5}(I = 4, M = 8)$: Parallel coordinate plot of the Pareto front approximations obtained with the random and the conflict partition strategies. 78

Figure 23 Inverted Generational distance results in problem $\text{DTLZ5}(I = 4, M)$. From 4-9 objectives a partition with 2 subspaces was generated, while for 10-15 objectives it was generated one with 3 subspaces was generated. 79

Figure 24 $\text{DTLZ5}(I = 4, M)$: ϵ-indicator results. The vertical axis of each subplot denotes the corresponding ϵ value, and the horizontal axis the boxplot for each number of objectives considered. 80

Figure 25 Normalized Hypervolume results for $\text{DTLZ5}(I = 4, M = 8)$. 81

Figure 26 $\text{DTLZ5}(I = 12, M = 24)$: Subspaces generated using 2 subspaces with 12 objectives, and 6 subspaces with 4 objectives. 82

Figure 27 DTLZ5($I = 12, M = 24$): Online Generational Distance using a partition with 2 subspaces and another one with 6 subspaces. 82
Figure 28 DTLZ5($I = 12, M = 24$): Parallel coordinate plot of the Pareto front approximations obtained by the conflict-based strategy with 6 subspaces, and the random strategy with 2 subspaces. 83
Figure 29 Generational Distance and inverted generation distance in problem $DTLZ2_{BZ}$. 85
Figure 30 ϵ-indicator results on $DTLZ2_{BZ}$. 86
Figure 31 Hypervolume on $DTLZ2_{BZ}$. 86
Figure 32 Generated subspaces by the conflict-based partition strategy on the Knapsack problem. 87
Figure 33 Conflict contribution of each of the three subspaces generated using the conflict partition strategy. 87
Figure 34 Generational Distance and inverted generation distance in the Knapsack problem. 88
Figure 35 Normalized Hypervolume on the Knapsack problem. The hypervolumen values were normalized with respect to the hypervolume achieved by the NSGA-II. 88

Figure 36 Pareto front approximations obtained by the *expansion* and the *average ranking* relations. 93
Figure 37 Online GD achieved with the expansion relation using different values for **S** in DTLZ2 with 5 objectives. 96
Figure 38 Online GD achieved with the expansion relation using different values for **S** in DTLZ2 with 10 objectives. 96
Figure 39 Online GD achieved with the expansion relation using different values for **S** in DTLZ2 with 15 objectives. 96
Figure 40 Distribution of the Chebyshev distance obtained with the expansion relation and using different values of **S** in DTLZ2 with 5 objectives. 97
Figure 41 Distribution of the Chebyshev distance obtained with the expansion relation and using different values of **S** in DTLZ2 with 15 objectives. 97
Figure 42 Online GD achieved by the preference relations in DTLZ2 with 3 objectives. 99
Figure 43 Online GD achieved by the preference relations in DTLZ2 with 5 objectives. 99

Figure 44 Online generational distance achieved by the preference relations in the problem DTLZ2 with 10 objectives. 99
Figure 45 Online generational distance achieved by the preference relations in the problem DTLZ2 with 15 objectives. 100
Figure 46 IGD achieved using the preference relations on DTLZ2. 100
Figure 47 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 3 objectives. 100
Figure 48 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 5 objectives. 101
Figure 49 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 10 objectives. 101
Figure 50 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 15 objectives. 101
Figure 51 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 3 objectives. 102
Figure 52 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 5 objectives. 102
Figure 53 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 10 objectives. 103
Figure 54 Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 15 objectives. 103
Figure 55 IGD achieved using the preference relations on DTLZ7. 104

Figure 56 Illustration of how the objective space is divided, and how the vectors in each subspace are compared. 112
Figure 57 Nondominated solutions with respect to the Chebyshev relation. 112

Figure 58 Illustration of the Chebyshev preference relation incorporated into NSGA-II and Strength Pareto Evolutionary Algorithm 2 (SPEA2), using feasible and infeasible reference points. $\mathbf{z}^{\star}_{\infty}$ is the optimum of the achievement function with respect to the current population of the MOEA. In all the examples, we used a threshold $\delta = 0.2$. 114

Figure 59 Illustration of the central-guided Chebyshev preference relation incorporated into NSGA-II. In these plots $\mathbf{z}^{\text{ref}}$ is the approximation of the ideal point, and $\mathbf{z}^{\star}_{\infty}$ is the the vector that we consider the central point of the Pareto front. 116

Figure 60 Plots of the complexity of NSGA-II and Multiobjective Genetic Algorithm (MOGA)'s ranking procedures using the Pareto dominance relation ($O(km^2)$), and the Chebyshev relation ($O(km + m^2)$). 117

Figure 61 PARSEC airfoil parametrization. 120

Figure 62 Simulation of the interactive method. 121

Figure 63 Airfoil of the most preferred solution from the simulation of the interactive method. 122

Figure 64 Airfoil with the best achievement value and the reference airfoil for the problem with 3 objectives. 124

Figure 65 Airfoil with the best achievement value and the reference airfoil for the problem with 6 objectives. 125

Figure 66 Ranks generated using the central-guided Chebyshev preference relation on a population of Pareto-nondominated solutions. Best ranked solutions are located in the central region of the Pareto front. 126

Figure 67 Growth of the normalized threshold, τ, with respect to the current generation. 128

Figure 68 Hypervolume results for problem DTLZ5(I, M) with different numbers of objectives. The results are averaged over 30 runs. 131

Figure 69 Generational distance for problem DTLZ5(I, M) with different numbers of objectives. The results are averaged over 30 runs. 131

Figure 70 Results of the ϵ-indicator for problem DTLZ5(I, M). Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively. 132

Figure 71 Online generational distance for problem DTLZ1 using 3 objectives. The results are averaged over 30 runs. 133

Figure 72 Online generational distance for problem DTLZ1 using 6 objectives. The results are averaged over 30 runs. 133
Figure 73 Generational distance for problem DTLZ1 with different numbers of objectives. The results are averaged over 30 runs. 134
Figure 74 Generational distance for problem DTLZ2 with different numbers of objectives. The results are averaged over 30 runs. 134
Figure 75 Online generational distance for problem DTLZ2 using 3 objectives. The results are averaged over 30 runs. 134
Figure 76 Online generational distance for problem DTLZ2 using 6 objectives. The results are averaged over 30 runs. 134
Figure 77 Hypervolume results for problem DTLZ1 with different numbers of objectives. The results are averaged over 30 runs. 135
Figure 78 Hypervolume results for problem DTLZ2 with different numbers of objectives. The results are averaged over 30 runs. 135
Figure 79 $g(\mathbf{x})$ values for problem DTLZ7 with different numbers of objectives. The results are averaged over 30 runs. 135
Figure 80 Results of the ϵ-indicator for problem DTLZ1. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively. 136
Figure 81 Results of the ϵ-indicator for problem DTLZ2. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively. 137
Figure 82 Results of the ϵ-indicator for problem DTLZ7. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively. 138
Figure 83 Results of the ϵ-indicator for problem WFG6. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively. 139
Figure 84 Illustration of some dominance resistant solutions (Dominance Resistant Solutionss (DRSs)) in problem DTLZ2. 140

Figure 85 Feasible objective function space of DTLZ2 (left) and the extended DTLZ2 (right) and 20 000 solutions generated at random. Both problems have the same Pareto optimal front. However, the extended version avoids dominance resistant solutions. 141
Figure 86 Online generational distance for NSGA-II and SPEA2 using the extended version of **DTLZ1**. The extended version removes dominance resistant solutions. 142
Figure 87 Online generational distance for NSGA-II and SPEA2 using the extended version of **DTLZ2**. The extended version removes dominance resistant solutions. 142
Figure 88 Hypervolume results for problem WFG6 with different numbers of objectives. The results are averaged over 30 runs. 144
Figure 89 Approximation set obtained using the Pareto dominance and Chebyshev dominance relation (threshold of zero) with both 40 000 and 400 000 evaluations. 145

LIST OF TABLES

Table 1 Bound for the number of points required to represent a Pareto front with resolution $r = 25$. 32

Table 2 Complexity of the objective reduction algorithms considered in this study (m is the size of the nondominated set, k the number of objectives and g the number of generations for each run of NSGA-II). 44
Table 3 Essential objectives identified by the proposed MOSSA, Deb and Saxena's algorithm and Brockhoff and Zitzler's algorithm. 46
Table 4 Comparison with respect to the δ-error using the 0/1 knapsack problem with 10 objectives. 48
Table 5 Comparison with respect to the δ-error using the 0/1 knapsack problem with 20 objectives. 49
Table 6 Comparison of the three algorithms with respect to the δ-error, using the DTLZ2_{BZ} problem with 10 objectives. 50

Table 7 Comparison of the three algorithms with respect to the δ-error, using the DTLZ2$_{BZ}$ problem with 20 objectives. 51
Table 8 Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the 0/1 knapsack problem with 10 objectives. 52
Table 9 Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the 0/1 knapsack problem with 20 objectives. 53
Table 10 Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the DTLZ2$_{BZ}$ problem with a total of 10 objectives. 54
Table 11 Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the DTLZ2$_{BZ}$ problem with a total of 20 objectives. 55

Table 12 Results of the reduction schemes with respect to the ϵ-Indicator in the DTLZ2$_{BZ}$ problem using a fixed-time stopping criterion. 63
Table 13 Results of the reduction schemes with respect to the ϵ-indicator in the 0/1 Knapsack problem using a fixed-time stopping criterion. 65
Table 14 Results of the reduction schemes with respect to the value D in the DTLZ2$_{BZ}$ problem using a fixed-generations stopping criterion. 67
Table 15 IGD values for DTLZ5$(I = 12, M = 24)$ using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions. 83
Table 16 $I_{\epsilon+}$ values for DTLZ5$(I = 12, M = 24)$ using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions. 84

Table 17 Parameter ranges for the PARSEC airfoil representation for problems A720 (2 and 3 objs.) and NLF0416 (6 objs.). 120
Table 18 Parameter values at each interaction point. 123
Table 19 Statistics of the achievement function values obtained with preferences and without them in the 3-objective problem. 124
Table 20 Objectives of the airfoil design problem with 6 objectives. 125

Table 21 Statistics of the achievement function values obtained with preferences and without them in the 6-objective problem. 125

ACRONYMS

AR	Average Ranking
CFD	Computational Fluid Dynamics
DM	decision maker
DRS	Dominance Resistant Solutions
EA	Evolutionary Algorithm
ER	Expansion preference relation
FR	Favour relation
IBEA	Indicator-Based Evolutionary Algorithm
KOSSA	k-size subset of non-redundant objectives algorithm
MCDM	Multi-criteria Decision Making
MADA	Multi-attribute Decision Analysis
MOEA	Multiobjective Evolutionary Algorithm
MOGA	Multiobjective Genetic Algorithm
MOP	Multiobjective Optimization Problem
MOSSA	minimum subset of non-redundant objectives with the minimum error possible
MR	Maximum Ranking
NSGA	Nondominated Sorting Genetic Algorithm
NSGA-II	Nondominated Sorting Genetic Algorithm II
PARSEC	PARSEC

POR	Preference Order Relation
ROI	Region of Interest
SPEA	Strength Pareto Evolutionary Algorithm
SPEA2	Strength Pareto Evolutionary Algorithm 2
VEGA	Vector Evaluated Genetic Algorithm

1 INTRODUCTION

Contents

1.1 Problem Statement 4
1.2 General and Specific Goals of the Thesis 5
1.3 Structure of the Thesis 7

In many industrial and research fields such as Engineering, Science, Chemistry, and Biology, just to name a few, there is a need for finding the values that minimize a certain cost required or maximize some benefit. As an example, let us consider the design of a chopper motorcycle frame carried out by Rodríguez et al. [95, 96]. In that work, the design was focused to optimize two objectives: (*i*) minimize the frame mass, and (*ii*) minimize the maximum structural stress. The first objective is related to motorcycle's weight and savings in manufacturing materials. If the mass is minimized, of course, the motorcycle's weight is reduced, however the cost of the materials increases since lighter types of steel are more expensive. In turn, the second objective, has to do with the safety factor of the motorcycle. A high structural stress increases the chances of an abrupt breaking of the frame.

A first example of a multiobjective optimization problem

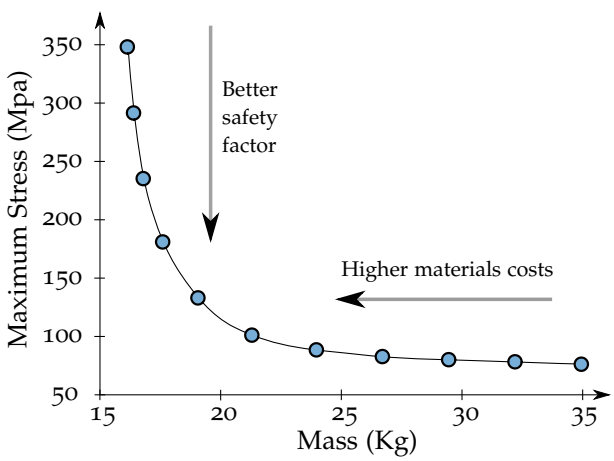


Figure 1: Set of alternative solutions for the optimization example of a chopper motorcycle frame.

Problems like this, in which several objectives are considered simultaneously, are called *Multiobjective Optimization Problems (MOPs)* or *multi-criteria optimization problems*. A distinctive feature of MOPs is that the objectives are defined in incomparable units, and they present some degree of conflict among them (i.e., one objective cannot be improved without deterioration of at least another objective). In the example of the design of a motorcycle frame, the objectives are in conflict, since if the mass decreases it is expected that the maximum structural stress increases. In other words, as the motorcycle's weight is decreased, the chances for an abrupt breaking of the frame increase. Figure 1 shows several alternative solutions for the motorcycle design. The plot in the figure shows the performance of each solution in each of the objectives. As we can see, the mass is given in kilograms (Kg), while the structural stress is given in mega pascals (Mpa). That space, in which each axis corresponds to one objective, is known as *objective function space*. It is also interesting to note that those solutions in the objective space represent different trade-offs between the objectives. For instance, in the lower-right region of the plot, we can find the design with the best structural stress, but the heaviest frame. In contrast, in the upper-left region of the plot, there is the solution with the worst structural stress, but the lighter frame. Between those solutions, we can see the remainder trade-off solutions, i.e., one objective is improved, while the other is worsened. The interested reader is referred to [20] for a comprehensive collection of real-world multiobjective optimization applications.

Notion of "optimality" in multiobjective optimization

One important element both in single- and multi-objective optimization is the preference relation to compare solutions in the objective search space in order to guide the search, and eventually, converge to optimal solutions. In single objective optimization the "less or equal than" relation (assuming minimization) is usually used. Therefore, the result of the search is a single optimum, i.e., a solution less or equal than any other solution in the scalar objective function space. In contrast, in multiobjective optimization, the solutions are compared in an objective function space composed of vectors (one component per objective). The common preference relation used in this case, is the *Pareto dominance relation*. By using this relation there is usually not a single optimal solution, but a set of alternatives representing different trade-offs among the objectives (e.g., those solutions in the example of the design of a motorcycle frame). Those alternative solutions (represented by the values of the variables of the problem) are called *Pareto optimal solutions* and their image in objective function space is known as *Pareto optimal front* (see Figure 1).

In summary, a MOP has a set of several optimal solutions. Nevertheless, in practice, only one solution should be selected from the set of Pareto optimal solutions. Therefore, multiobjective optimization also differs from

single-objective optimization in that the former is composed of two different tasks to solve the problem: a searching task whose goal is to find Pareto optimal solutions, and a decision making task in which a most preferred solution is chosen from the set of Pareto optimal solutions. For instance, in our example, although there are several trade-off designs for the motorcycle frame, only one of the designs should be built. The particular chosen design depends on the particular purposes of the motorcycle. If the prime goal is to produce a light frame, a higher priority must be given to minimize the frame's mass, although that means an increase in material cost, and, further, the safety would be compromised. On the other hand, if a reduction in the cost of materials and a high safety factor are more important, then the search must emphasize the minimization of structural stress.

There are plenty of types of real-world MOPs depending on the characteristics of their components. There are, for example, constrained and unconstrained problems, or problems with continuous or discrete search spaces. This situation gives rise to different optimization methods intended to solve an specific type of MOP. Despite the variety of optimization techniques proposed, in real-world situations there are some types of MOPs in which traditional mathematical programming techniques (see, for example Section 2.2) present a poor performance, or they are not even applicable to those problems. Some examples of these types of problems are the following. Problems in which the objective functions are not given in a closed form, but they are defined by a simulation model. What is more, in some problems the hardware device to be optimized is directly used to evaluate a candidate solution. An additional common characteristic in real-life problems is the presence of noise in objective functions, or dynamic objective functions.

These complexities call for alternative approaches to deal with these types of MOPs. Among these alternative approaches we can find Evolutionary Algorithms (EAs), which are stochastic search and optimization methods that simulate the natural evolution process. In 1984, David Schaffer [100] proposed the first actual implementation of what it is now called Multiobjective Evolutionary Algorithm (MOEA). From that moment on, many researchers [21, 107, 128, 58, 48, 70] have proposed their own MOEAs. Since EAs work with sets of solutions (or population), they are able to find several Pareto optimal solutions in a single run. One important feature of most of these MOEAs is that they use the Pareto dominance relation to compare solutions in objective function space. Specifically, the solutions in the population are ranked using the Pareto dominance relation and the best solutions are combined and varied to generate solutions closer to the Pareto optimal set.

Evolutionary algorithms to solve multiobjective problems

Although MOEAs have been successfully applied to solve many real-world problems, most of these applications consider a small number of objectives.

As we will see in the next section, Pareto-based MOEAs struggle to solve problems with a large number of objectives.

1.1 Problem Statement

Nowadays, MOEAs have shown a remarkable performance in many real-life problems with 2 or 3 objectives. However, recent experimental [61, 117, 92] and analytical [69, 113] studies have shown that MOEAs based on the Pareto dominance relation scale poorly in MOPs with a high number of objectives (4 or more). Although this limitation seems to affect only to Pareto-based MOEAs, optimization problems with a large number of objectives (also known as *many-objective problems*) introduce some difficulties common to any other multi-objective optimizer. In summary, three of the most serious difficulties due to high dimensionality are the following:

1. *Deterioration of the search ability*. A commonly accepted reason for this problem is that the proportion of nondominated solutions (i.e., solutions not worst than any other according to the Pareto dominance relation) in a population increases rapidly with the number of objectives [42]. According to Bentley et al. [6] the number of nondominated k-dimensional vectors on a set of size n is $O(\ln^{k-1} n)$. As a consequence, in a many-objective problem, the selection of solutions is carried out almost at random or guided by diversity criteria.

2. *Dimensionality of the Pareto front*. The number of points required to represent accurately a Pareto front increases exponentially with the number of objectives. Specifically, the number of points necessary to represent a Pareto front with k objectives and resolution r is given by kr^{k-1} (e.g., see [103]). This poses a challenge both to the data structures to efficiently manage that number of points and to the density estimators to achieve an even distribution of the solutions along the Pareto front.

3. *Visualization of the Pareto front*. Clearly, with more than three objectives is not possible to plot the Pareto front as usual. Several visualization techniques have been proposed for high-dimensional Pareto fronts (see e.g., [119, 90]), but none of them is widely used and practically all of them are non-intuitive. Visualization is important because it plays a key role in decision making.

1.2 General and Specific Goals of the Thesis

This thesis deals with the design of new methods to improve Pareto-based MOEAs in order to deal with many-objective problems. Most of the methods proposed are specifically designed to cope with the deterioration of the search ability of Pareto-based MOEAs. Nonetheless, they can also be considered as a remedy to deal with a high dimensionality in the Pareto front, and its corresponding visualization.

1.2.1 *Main Goal*

The main goal of this research is to advance the state-of-the-art with respect to the effectiveness of Pareto optimality in problems with many objectives. Since Pareto optimality is also used in classical optimization techniques the contributions of this study may also useful to the Operations Research community.

1.2.2 *Specific Goals and Contributions of the Thesis*

In order to accomplish its main goal, This thesis is divided in four specific goals. The contributions of this thesis are a direct result of the accomplishment of the specific goals. It is worth noting that the specific goals are described next in the order in which they are presented in this document, and not by their relevance.

1. **Development of new techniques to deal with many-objective problems**. The first contribution proposes two objective reduction techniques intended to be used *a posteriori*, i.e., once an approximation of the set of Pareto optimal solutions has been found. Both reduction techniques are based on a feature selection technique whose main goal is to identify the redundant objectives to remove them in order to obtain a smaller objective subset. In a decision making scenario these techniques can allow the decision maker to work with a moderate number of objectives in such a way that it is still possible, for instance, to plot the Pareto front approximation. According to the results, the reduction algorithms are effective and efficient compared with two reduction algorithms that have been recently proposed. This contribution has been published in [78].

   In another contribution, we proposed two different approaches to integrate one of the proposed reduction techniques into a MOEA. In particular, the conflict information was used during the search in order to improve the search ability of the MOEA. The first approach gradually

reduces the number of objective during the search and at the end of the search it incorporates all the objectives to obtain a final approximation of the entire Pareto front. The resulting MOEA outperformed the original MOEA in all the test problems studied. Therefore, the objective reduction during the search is an excellent choice to deal with many-objective problems. We have published this contribution in [79]. In the second approach, instead of removing the non-conflicting objectives, the conflict information is used to group objectives in terms of their degree of conflict. This way all objectives are used during all the search. Both schemes have shown a better scalability with respect to the number of objectives than a regular MOEA. This contribution has been published in [81]. Later on, an improvement of this method was published in [82].

2. **Development of an alternative preference relation to Pareto optimality**. Regarding alternative preference relations, we present a comparative study of some preference relations proposed to deal with many-objective problems. The main goal of this study is to identify the advantages and disadvantages of the current relations in order to propose a new preference relation. One important discovery of the study was the fact that the preference relations converge to different regions of the Pareto front. Thus, to determine the effectiveness of a Pareto relation we have to take into account both the speed of converge and the location of the optimal subset achieved by the relation. This study was first published in [80]. Afterwards, in [77] we have published additional results.

   After this study, we propose a preference relation designed to solve many-objective problems. Since a limitation of some preference relations is the fact that the emphasized region was fixed, our proposed preference relation is based on a reference point approach. This way, the decision maker can decide the emphasized region during the search. Another advantage of this new preference relation is that it is able to rank nondominated solutions (i.e., equivalent solution according to the Pareto dominance relation). Thus, a MOEA can further select solutions based on the closeness to the Pareto front. This and the next contribution have been published in [76].

3. **Development of a new scheme to deal with many objective problems that takes into account the decision maker's preferences.** Based on the proposed preference relation we designed an interactive scheme to incorporate preferences into a MOEA. The scheme allows the decision maker to gradually provide his preferences in order to gradually reduce the region of interest on the Pareto front. The experimental

evaluation shows that the interactive scheme was able to improve the search ability of the reference MOEA in many-objective problems. Besides, an interactive scheme reduces the number of function evaluations since only solutions in the region of interest are evaluated.

4. **Gain knowledge about the sources that cause problems for MOEAs based of Pareto optimality.** Along the contribution presented in the following chapters we have found some interesting results about scalability in many-objective problems. The most important is that not all MOPs become harder as the number of objectives is increased (see Chapter 8). The difficulty of some problems is practically the same regardless of the number of objectives. Another finding is that the conflict among the objectives plays an important role in the difficulty of many-objective problems. Problems in which the differences of conflict among the objective is small are harder.

1.3 Structure of the Thesis

This thesis is organized in 9 chapters. The first 4 chapters describe background concepts required to understand the rest of this thesis. The last 5 chapters present the thesis contributions and their corresponding results and conclusions.

Chapter 2 presents a brief introduction to multiobjective optimization problems. This chapter also introduces the traditional Pareto dominance relation which is commonly used in multiobjective optimizers.

In Chapter 3, we describe the two most important groups of optimization methods to solve MOPs, namely mathematical programming methods, and multiobjective evolutionary algorithms. We also describe the main elements required to design a multiobjective evolutionary algorithm.

Chapter 4 introduces the challenges posed by multiobjective optimization problems with a large number of objectives. We also present a brief review of the current proposals found in the specialized literature to deal with this type of problems.

The first contribution of this thesis work is presented Chapter 5. In this chapter we describe the design and results of the two objective reduction algorithms. Besides of the fact that these algorithms can be directly used during the decision making process, they also constitute the basis of the two schemes proposed in Chapter 6. These schemes use one of the objective reduction methods that adopt conflict information to improve the search ability in many-objective problems.

In Chapter 7 we present a comparative study of some preference relation techniques. Later on, in Chapter 8 we present our proposal of a new

preference relation to deal with many-objective problems. That relation was incorporated in an interactive method so that the decision maker can gradually provide his\her preferences to reduce the region of interest. The proposed relation was also used in a scheme in order to find the entire Pareto front.

Finally, in Chapter 9 we present the conclusions obtained regarding our contributions. Also, we describe future paths of research.

2 MULTIOBJECTIVE OPTIMIZATION PROBLEMS

Contents

2.1 Notions of Optimality in a MOP 10
2.2 Approaches to Solve a MOP 14

Many problems in engineering, industry, and in many other fields, involve the simultaneous optimization of many objectives. In many cases, the objectives are defined in incomparable units, and they present some degree of conflict among them (i.e., one objective cannot be improved without deterioration of at least any other objective). These problems are called *multiobjective* or *multicriteria* problems. Let us consider, for example, a shipping company which is interested in minimizing the total duration of its routes to improve customer service. Additionally, the company also wants to minimize the number of trucks used in order to reduce operating costs. Clearly, these objectives are in conflict since adding more trucks reduces the duration of the routes, but increases operation costs. In addition, the objectives of this problem are expressed in different measurement units.

In single-objective optimization, it is possible to determine between any given pair of solutions if one is better than the other. As a result, we usually obtain a single optimal solution (i.e., the global optimum). However, in multiobjective optimization there does not exist a straightforward method to determine if a solution is better than other. The method most commonly adopted in multiobjective optimization to compare solutions is called *Pareto dominance relation* [91] which, instead of a single optimal solution, leads to a set of alternatives with different trade-offs among the objectives. These solutions are called *Pareto optimal solutions* or *non-dominated solutions*.

Next, we present some general concepts and notations used in the remainder of this document. These concepts are repeatedly used in several of the further chapters, whereas concepts only used in particular chapters, are defined when they are used in order to facilitate the reading of this thesis.

Definition 1 (Multiobjective Optimization Problem)**.** *Formally, a Multiobjective Optimization Problem (MOP) is defined as:*

$$\begin{aligned} \text{“}\textit{Minimize}\text{”} \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})]^T \\ \textit{subject to} \quad & \mathbf{x} \in \mathcal{X}. \end{aligned} \tag{2.1}$$

The vector $\mathbf{x} \in \mathbb{R}^n$ is formed by n *decision variables* representing the quantities for which values are to be chosen in the optimization problem. The *feasible set* $\mathcal{X} \subseteq \mathbb{R}^n$ is implicitly determined by a set of equality and inequality constraints of the form $g_i(\mathbf{x}) \leqslant 0$, and $h_i(\mathbf{x}) = 0$, respectively. The vector function $\mathbf{f} : \mathcal{X} \to \mathbb{R}^k$ is composed by $k \geqslant 2$ scalar *objective functions* $f_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \ldots, k)$. In multiobjective optimization, the sets $\mathbb{R}^n$ and $\mathbb{R}^k$ are known as *decision variable space* and *objective function space*, respectively. The image of $\mathcal{X}$ under the function $\mathbf{f}$ is a subset of the objective function space denoted by $\mathcal{Z} = \mathbf{f}(\mathcal{X})$ and referred to as the *feasible set in the objective function space*. Figure 2 presents an example with 2 decision variables and 2 objective functions. In that figure we can appreciate how the vector-valued function $\mathbf{f}$ maps solutions in the feasible set, $\mathcal{X}$, into the feasible set, $\mathcal{Z}$, in the objective function space. The objective function space attracts a lot of attention in multiobjective optimization since the performance of each solution is evaluated in that space.

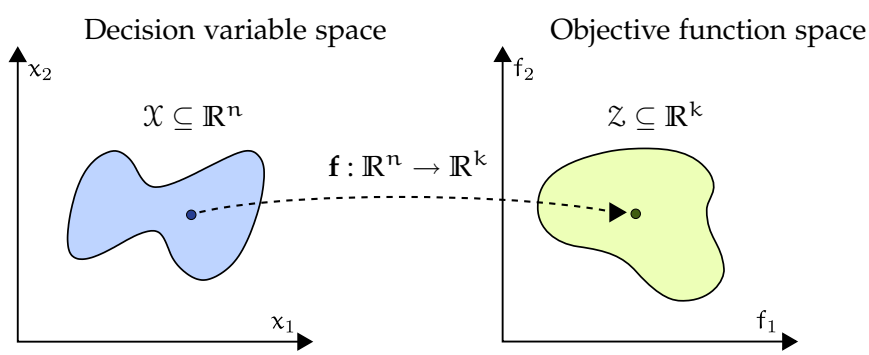


Figure 2: Search spaces in multiobjective optimization problems.

2.1 Notions of Optimality in a MOP

In order to define more precisely the multiobjective optimization problem stated in definition 1 we have to establish the meaning of minimizing in $\mathbb{R}^k$. That is to say, it is required to define how vectors $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^k$ have to be compared for different solutions $\mathbf{x} \in \mathbb{R}^n$. The relation “less than or equal” ($\leqslant$) is used in single-objective optimization to compare the scalar objective

values. By using this relation there may be many different optimal solutions $\mathbf{x} \in \mathcal{X}$, but only one optimal value $f_{\min} = \min\{f(\mathbf{x}) \mid x \in \mathcal{X}\}$ since the relation $\leqslant$ induces a total order in $\mathbb{R}$ (i.e., every pair of solutions is comparable so we can sort solutions from the best to the worst one). In contrast, in multiobjective optimization problems, there is no canonical order on $\mathbb{R}^k$, and thus, we need weaker definitions of order to compare vectors in $\mathbb{R}^k$.

In single-optimization we can impose a total order on the solutions

There is no canonical order on $\mathbb{R}^k$ in multiobjective optimization

In multiobjective optimization, the *Pareto dominance relation* originally proposed by Edgeworth in 1881 [39], and later generalized by the french-italian economist Vilfredo Pareto in 1896 [91] is usually adopted.

Definition 2 (Pareto Dominance relation)**.** *It is said that a vector $\mathbf{z}^1$ dominates vector $\mathbf{z}^2$, denoted by $\mathbf{z}^1 \prec_{pareto} \mathbf{z}^2$, if and only if:*

$$\forall i \in \{1, \ldots, k\} : z_i^1 \leqslant z_i^2 \qquad \textit{and} \qquad \exists i \in \{1, \ldots, k\} : z_i^1 < z_i^2. \tag{2.2}$$

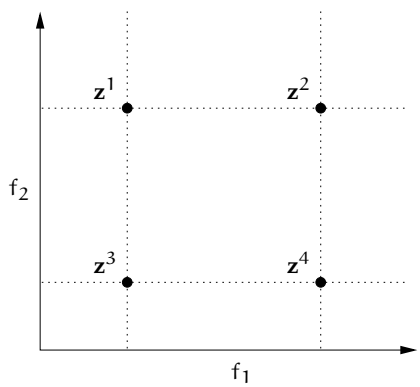


Figure 3: Illustration of the concept of Pareto dominance relation.

Figure 3 illustrates the Pareto dominance relation with an example with four 2-dimensional vectors. Vector $\mathbf{z}^3$ is strictly less than $\mathbf{z}^2$ in both objectives, therefore $\mathbf{z}^3 \prec_{\text{pareto}} \mathbf{z}^2$. Vector $\mathbf{z}^3$ also Pareto-dominates $\mathbf{z}^1$ since with respect to f_1 those vectors are equal, but in f_2, $\mathbf{z}^3$ is strictly less than $\mathbf{z}^1$. Since $\prec_{\text{pareto}}$ is not a total order, some elements can be incomparable like is the case with $\mathbf{z}^1$ and $\mathbf{z}^4$, i.e., $\mathbf{z}^1 \not\prec_{\text{pareto}} \mathbf{z}^4$ and $\mathbf{z}^4 \not\prec_{\text{pareto}} \mathbf{z}^1$. The remainder comparisons are the following: $\mathbf{z}^3 \prec_{\text{pareto}} \mathbf{z}^4$, $\mathbf{z}^1 \prec_{\text{pareto}} \mathbf{z}^2$, and $\mathbf{z}^4 \prec_{\text{pareto}} \mathbf{z}^2$.

Thus, to find the optimal solutions of a MOP we have to find those solutions $\mathbf{x} \in \mathcal{X}$ whose images, $\mathbf{z} = \mathbf{f}(\mathbf{x})$, are not Pareto-dominated by any other vector in the feasible space. In the example of Figure 3, no vector dominates $\mathbf{z}^3$, and, therefore, we say that $\mathbf{z}^3$ is *nondominated*. If for two solutions, $\mathbf{z}^1$ and $\mathbf{z}^2$, happens that $\mathbf{z}^1 \not\prec_{\text{pareto}} \mathbf{z}^2$ and $\mathbf{z}^2 \not\prec_{\text{pareto}} \mathbf{z}^1$, then it is said that those solution are *mutually nondominated*.

Definition 3 (Pareto Optimality). *A solution $\mathbf{x}^* \in \mathcal{X}$ is said to be Pareto optimal if there does not exist another solution $\mathbf{x} \in \mathcal{X}$ such that $\mathbf{f}(\mathbf{x}) \prec_{pareto} \mathbf{f}(\mathbf{x}^*)$.*

The set of Pareto optimal solutions and its image in objective space is defined as follows.

Definition 4 (Pareto optimal set). *The Pareto optimal set, P_{opt}, is defined as:*

$$P_{opt} = \{\mathbf{x} \in \mathcal{X} \mid \nexists\, \mathbf{y} \in \mathcal{X} : \mathbf{f}(\mathbf{y}) \prec_{pareto} \mathbf{f}(\mathbf{x})\}. \tag{2.3}$$

Definition 5 (Pareto front). *For a Pareto optimal set, P_{opt}, the Pareto front, PF_{opt}, is defined as:*

$$PF_{opt} = \{\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})) \mid \mathbf{x} \in P_{opt}\}. \tag{2.4}$$

Figure 4 illustrates the concept of Pareto optimal set and its image in the objective space, the Pareto front. Darker points denote Pareto optimal vectors. In variable space, these vectors are referred to as Pareto optimal decision vectors, while in objective space, they are called Pareto optimal objective vectors. As we can see in the figure, the Pareto front is only composed by nondominated vectors.

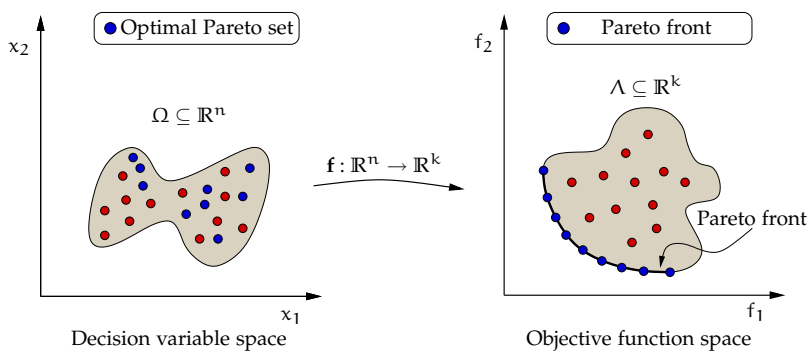


Figure 4: Illustration of the Pareto optimal set and its image, the Pareto front.

In multiobjective optimization there is another commonly employed variant of the Pareto dominance relation. This variant relaxes the relation so that the resulting optimal set contains additional solutions compared with the Pareto optimal set. Next, we define that alternative preference relation, and its corresponding optimal set.

Definition 6 (Strict Pareto dominance). *A vector $\mathbf{z}^1$ strictly dominates a vector $\mathbf{z}^2$, denoted by $\mathbf{z}^1 \prec\prec_{pareto} \mathbf{z}^2$, if and only if:*

$$\forall i \in \{1, \ldots, k\} : \; z_i^1 < z_i^2. \tag{2.5}$$

In the example shown in Figure 3, we saw that solution $\mathbf{z}^3$ Pareto dominates 3 solutions, namely, $\mathbf{z}^2$, $\mathbf{z}^1$ and $\mathbf{z}^4$. However, solution $\mathbf{z}^3$ is not strictly better than solutions $\mathbf{z}^1$ and $\mathbf{z}^4$ in all the objectives. Consequently, solution $\mathbf{z}^3$ only strictly Pareto dominates solution $\mathbf{z}^2$, i.e., $\mathbf{z}^3 \prec\prec_{\text{pareto}} \mathbf{z}^2$. In general, if we look for a solution $\mathbf{x}^* \in \mathcal{X}$ such that there is no $\mathbf{x} \in \mathcal{X}$ that strictly dominates it, then we are requiring less for $\mathbf{x}^*$ than in the case of Pareto optimality. Therefore, the optimality in terms of the relation $\prec\prec$ is known as weak Pareto optimality.

Definition 7 (Weak Pareto Optimality)**.** *A solution $\mathbf{x}^* \in \mathcal{X}$ is said to be weak Pareto optimal if there does not exist another solution $\mathbf{x} \in \mathcal{X}$ such that $\mathbf{f}(\mathbf{x}) \prec\prec_{pareto} \mathbf{f}(\mathbf{x}^*)$.*

Definition 8 (Weak Pareto optimal set)**.** *The weak Pareto optimal set, P_{weak}, is defined as:*

$$P_{weak} = \{\mathbf{x} \in \mathcal{X} \mid \nexists\, \mathbf{y} \in \mathcal{X} : \mathbf{f}(\mathbf{y}) \prec\prec \mathbf{f}(\mathbf{x})\}. \tag{2.6}$$

If $\mathbf{x}^* \in P_{\text{opt}}$, that means that there is no solution $\mathbf{x} \in \mathcal{X}$, $\mathbf{f}(\mathbf{x}) \neq \mathbf{f}(\mathbf{x}^*)$, such that $\mathbf{f}(\mathbf{x}) \leqslant \mathbf{f}(\mathbf{x}^*)$. Clearly, it also holds that there is no solution $\mathbf{x} \in \mathcal{X}$ such that $\mathbf{f}(\mathbf{x}) < \mathbf{f}(\mathbf{x}^*)$. Therefore, from definitions 3 and 7 we can see that $\mathbf{x}^*$ is also a weak Pareto optimal solution. In other words, P_{opt} is a subset of P_{weak}.

In [44] it is provided a general definition of optimality, given a MOP defined by a binary relation $\mathcal{R}$, and the feasible sets $\mathcal{X}$, $\mathcal{Z}$.

Definition 9 (General Optimality)**.** *A solution $\mathbf{x}^* \in \mathcal{X}$ is said to be an optimal solution of a MOP if there does not exist another solution $\mathbf{x} \in \mathcal{X}$, $\mathbf{x} \neq \mathbf{x}^*$, such that $\mathbf{f}(\mathbf{x})\, \mathcal{R}\, \mathbf{f}(\mathbf{x}^*)$.*

Definition 10 (General Optimal Set)**.** *The optimal set of a MOP is defined as:*

$$\text{Opt} = \{\mathbf{x} \in \mathcal{X} \mid \nexists\, \mathbf{y} \in \mathcal{X} : \mathbf{f}(\mathbf{y})\, \mathcal{R}\, \mathbf{f}(\mathbf{x})\}. \tag{2.7}$$

As we mentioned before, Pareto dominance is the most common preference relation used in multiobjective optimization. However, it is only one of a set of useful preference relations available. For instance, in [44] the lexicographic relation and its complement, the lexicographic optimality, are presented.

Definition 11 (Lexicographic relation)**.** *It is said that a vector $\mathbf{z}^1$ is lexicographically preferred to vector $\mathbf{z}^2$, denoted by $\mathbf{z}^1 \preceq_{lex} \mathbf{z}^2$, if and only if:*

$$z^1_{k^*} < z^2_{k^*} \quad \textit{or} \quad \mathbf{z}^1 = \mathbf{z}^2, \tag{2.8}$$

where $k^* = \min\{k : z^1_k \neq z^2_k\}$.

This preference relation is useful when there exists a natural order of importance among the objectives. For example, let us consider the MOP to select a car we want to buy where the objectives are price, gasoline consumption, and power. In this example, the price might be more important than gasoline consumption, and this, in turn, can be more important than power. Thus, if two cars have the same price, then they are compared w.r.t. gasoline consumption, and so on. Note that the lexicographic relation is a total order since it is reflexive ($\mathbf{z}^1 \preceq_{\text{lex}} \mathbf{z}^1$), antisymmetric ($\mathbf{z}^1 \preceq_{\text{lex}} \mathbf{z}^2 \wedge \mathbf{z}^2 \preceq_{\text{lex}} \mathbf{z}^1 \Rightarrow \mathbf{z}^1 = \mathbf{z}^1$) and connected ($\mathbf{z}^1 \preceq_{\text{lex}} \mathbf{z}^2 \vee \mathbf{z}^2 \preceq_{\text{lex}} \mathbf{z}^1$).

As in the case of the Pareto dominance relation, for the lexicographic relation, a notion of optimality can be defined.

Definition 12 (Lexicographic optimality)**.** *A solution* $\mathbf{x}^* \in \mathcal{X}$ *is lexicographically optimal if* $\mathbf{f}(\mathbf{x}^*) \preceq_{lex} \mathbf{f}(\mathbf{x})$ *for all* $\mathbf{x} \in \mathcal{X}$.

In chapter 7 we present other preference relations proposed or used in the context of MOPs with a large number of objectives.

2.2 Approaches to Solve a MOP

In most MOPs, the Pareto optimal set is composed of a large or even an infinite number of solutions. Nevertheless, in practice, only one solution should be selected from the Pareto optimal set. The person who has the task of selecting the most preferred solution from the set of Pareto optimal alternatives is known as decision maker (DM). The preferences of the DM are incorporated in order to induce a total order among the solutions of PF_{opt}.

There are several approaches in which DM's preferences can be incorporated (or articulated). For instance, the DM can rank the set of objectives according to their importance. Another possibility is to obtain a sample of the Pareto front, and then, select a solution from this sample. A common classification of the techniques to solve MOPs takes into account the moment at which the DM is required to provide preference information [87, 22]. The classification is the following:

1. The preference information is incorporated after a representative sample of the entire Pareto front is obtained (*a posteriori* approaches).

2. The preference information is incorporated before the search process (*a priori* approaches). Thus, only a subset of the Pareto front, defined by the preferences, is attained.

3. The DM is iteratively asked to proportionate his/her preferences during the search (*interactive* approaches). In this way, the search is focused on a subset of the Pareto front that is gradually reduced.

Since the size of the Pareto optimal set might be infinite, in practice, the goal of an *a posteriori* approach is finding an approximation set of the Pareto optimal front with the "best" quality.

Definition 13 (Pareto front approximation)**.** *A Pareto front approximation, denoted by* PF_{approx}*, is a subset of the objective space* $\mathcal{Z}$ *composed of mutually nondominated vectors. I.e., for any two vectors* $\mathbf{z}^1, \mathbf{z}^2 \in \mathsf{PF}_{approx}$*,* $\mathbf{z}^1 \nprec \mathbf{z}^2 \wedge \mathbf{z}^2 \nprec \mathbf{z}^1$.

Currently, it is well accepted that the quality of an approximation set is determined by *i*) the closeness to the Pareto optimal front, and *ii*) the spread over the entire Pareto optimal front [130, 22].

In interactive optimization methods it is useful to know the lower and upper bounds of the Pareto front. The *ideal point* represents the lower bounds and is defined by $z_i^\star = \min_{z \in \mathcal{Z}}(z_i)$, $\forall i = 1, \ldots, k$. In turn, the upper bounds are defined by the *nadir point*, which is given by $z_i^{nad} = \max_{z \in \mathsf{PF}_{opt}}(z_i)$, $\forall i = 1, \ldots, k$. In order to avoid some problems when the ideal and nadir points are equal or very close, a point strictly better than the ideal point is usually defined. This point is called the *utopian point* and is defined by $z_i^{\star\star} = z_i^\star - \epsilon$, $\forall i = 1, \ldots, k$, where $\epsilon > 0$ is a small scalar.

In Section 6.2, a common methodology to assess the quality of a Pareto front approximation is presented.

3 OPTIMIZATION METHODS TO SOLVE MOPS

Contents

3.1 Mathematical Programming Methods 17
3.2 Evolutionary Multiobjective Algorithms 20

In this chapter we present two families of optimization methods to solve Multiobjective Optimization Problems (MOPs). The first group of methods is collectively known as mathematical programming methods and were proposed by the Multi-criteria Decision Making (MCDM) community. The other family of methods are the Multiobjective Evolutionary Algorithms (MOEAs), which are stochastic optimizers, proposed in the Evolutionary Algorithm (EA) field.

3.1 Mathematical Programming Methods

In this section we present a brief description of some of the most popular MCDM techniques. These methods are organized in *a priori* methods, *a posteriori* methods, and interactive methods. A wider selection of methods, and a detailed description of them can be found in [87, 40].

3.1.1 *A Priori Preference Articulation*

3.1.1.1 *Goal Programming*

In this method, developed by Charnes and Cooper [17], the decision maker (DM) has to assign targets or goals that wishes to achieve for each objective. These values are incorporated into the problem as additional constraints. The objective function then tries to minimize the absolute deviations from the targets to the objectives.

In addition, goal programming provides the flexibility to deal with cases that have multiple conflicting goals. Essentially, the goals may be ranked in order of importance to the problem solver. Note that this technique yields a nondominated solution if the goal point is chosen in the feasible space.

3.1.1.2 *Lexicographic Method*

In this method, the objectives are ranked in order of importance by the decision maker (from best to worst). The optimal value $f_i^\star$ $(i = 1, \ldots, k)$ is then obtained by minimizing the objective functions sequentially, starting with the most important one and then proceeding according to the order of importance of the objectives. Additionally, the optimal value found for each objective is added as a constraint for subsequent optimizations. This way, the optimal value of the most important objectives is preserved. Only in the case of several optimal solutions in the single optimization of the current objective, the rest of the objectives are considered. Therefore, in the worst case, we have to carry out k single objective optimizations. In [87, 40] it is proved that the optimal solution obtained by the lexicographic problem is Pareto optimal.

3.1.2 *A Posteriori Preference Articulation*

3.1.2.1 *Linear Combination of Weights*

In this method, the general idea is to associate each objective function with a weighting coefficient and minimize the weighted sum of the objectives. In this way, the multiobjective problem is transformed into a single objective problem. Some important properties of this method are the following [87]. The solution to the weighting problem is a weakly Pareto optimal solution. However, it is possible to achieve a Pareto optimal solution if the weighting coefficients are positive for all the objectives, or if the problem has a unique solution. Additionally, the method of linear combination of weights can generate any Pareto optimal solution of a convex MOP (i.e., if all the objective functions and the feasible region are convex).

3.1.2.2 *ε-Constraint Method*

The ε-constraint method is one of best known scalarization techniques to solve multiobjective problems. In this approach one of the objectives is minimized while the others are used as constraints bound by some allowable levels ε_i. In order to find several Pareto optimal solutions, we need to solve the ε-constraint problem using multiple different values for ε_i. The general scheme is an iterative optimization process in which the user needs to provide the range of the reference objective. In addition, it must be provided the increment for the constraints imposed by ε. This increment determines the number of Pareto optimal solutions generated. In [40] and [87] it is proved that the optimal solution of the ε-constraint problem is weakly Pareto optimal.

3.1.3 *Interactive Preference Articulation*

3.1.3.1 *Chebyshev Method*

The Chebyshev method proposed in [109], is an interactive method based on the minimization of a function value. The metric to be used for measuring the distances to an utopian objective vector is the weighted Chebyshev metric. Thus, the multiobjective optimization problem is transformed into a single-objective optimization problem. Every Pareto optimal solution of any multiobjective optimization problem can be found by solving the new single-objective problem. However, with this approach, some of the solutions may be weakly Pareto optimal solutions. This negative aspect is solved by formulating the distance minimization problem as a lexicographic problem. In each iteration, the Chebyshev method provides different subsets of nondominated solutions. These solutions consist of $P(\approx n)$ representative points, generated by using the lexicographic Chebyshev problem, from which the DM is required to select one as his most preferred.

3.1.3.2 *Reference Point Methods*

The reference point approach, proposed by Wierzbicki [121, 122], is an interactive multiobjective optimization technique based on the definition of an achievement scalarization function.

Definition 14 (Achievement scalarizing function)**.** *An achievement scalarizing function is a parameterized function* $s_{\mathbf{z}^{ref}}(\mathbf{z}) : \mathbb{R}^k \to \mathbb{R}$, *where* $\mathbf{z}^{ref} \in \mathbb{R}^k$ *is a reference point representing the decision maker's aspiration levels. Thus, the multiobjective problem is transformed into the following scalar problem:*

$$\begin{aligned} &\textit{Minimize} && s_{\mathbf{z}^{ref}}(\mathbf{z}) \\ &\textit{subject to} && \mathbf{z} \in \mathcal{Z}. \end{aligned} \tag{3.1}$$

The basic idea of this technique is the following. First, the DM is asked to give a reference point. Then, the solutions that better satisfy the aspiration levels are computed using the achievement scalarization function. If the DM is satisfied with the current solution, the interactive process ends. Otherwise, the DM must provide another reference point.

3.1.3.3 *Light Beam Search*

The light beam search method proposed by Jaszkiewicz and Słowiński [65], is an iterative method which combines the reference point idea and tools of Multi-attribute Decision Analysis (MADA). In each iteration, a finite sample of nondominated points is generated. The sample is composed of a current

point called *middle point*, which is obtained in the previous iteration, and J nondominated points from its neighborhood. A local preference model in the form of an *outranking relation* S is used to define the neighborhood of the middle point. It is said that a outranks b (aSb), if a is considered to be at least as good as b. The outranking relations are defined by the DM, which specify three preference thresholds for each objective. They are *indifference threshold*, *preference threshold* and *veto threshold*. The DM has the possibility to scan the inner area of the neighborhood along the objective function trajectories between any two characteristic neighbors or between a characteristic neighbor and the middle point.

3.2 Evolutionary Multiobjective Algorithms

MOPs can be classified in different groups depending on the characteristics of their components, i.e., whether the objective functions are convex or not, or whether the decision space is continuous or discrete. The following list presents a possible classification for MOPs based on the nature of some key elements. The following classification does not pretend to be exhaustive, but to simply provide a good idea of the common components that guide the design of an optimization method.

- **Constraints on the variables**. An optimization problem can have no constraints on the decision variables, or it can have equality or inequality constraints.
- **Nature of the decision space**. A problem is called an *integer multiobjective optimization problem* if some of the variables can only take integer values. On the other hand, if all the variables can take real values, the problem is known as a *continuous multiobjective optimization problem*.
- **Geometry of the objective functions and feasible space**. A MOP is convex if all the objective functions and the feasible variable space are convex.
- **Nature of the objective functions and constraints**. When all the objective functions and constraints are linear, then the MOP is called linear. If at least one objective or constraint is nonlinear the problem is called nonlinear MOP.

In many cases, the development of a new optimization technique is the result of the need to solve some kind of real-life MOP. Therefore, the design of that new technique is oriented to take advantage of the particular characteristics of the given problem. For instance, there are many techniques

specialized for solving linear multiobjective optimization problems (see e.g., [125]), or techniques devoted to solve convex MOPs (see e.g., [46]).

In real-world situations there are some types of MOPs in which traditional mathematical programming techniques (e.g., those presented in Section 2.2) present a poor performance, or they are not even applicable. Some examples are the following. Problems in which the objective functions are not given in a closed form, but they are defined by a simulation model (e.g., [96, 111]). What is more, in some problems the hardware device to be optimized is directly used to evaluate a candidate solution (e.g., [123, 11, 110, 72]). An additional common characteristic in real-life problems is the presence of noise in the objective functions (e.g., [66, 60, 4]), or dynamic objective functions (e.g., [5, 115]).

These complexities call for alternative approaches to deal with these types of MOPs. Among these alternative approaches we can find EAs, which are stochastic search and optimization methods that simulate the natural evolution process. At the end of the 1960s, Rosenberg [97] proposed the use of genetic algorithms to solve MOPs. However, it was until 1984, when David Schaffer [100] proposed the first actual implementation of what it is now called MOEA. From that moment on, many researchers [21, 107, 128, 58, 48, 70] have developed their own MOEAs.

As other stochastic search strategies (e.g., simulated annealing, ant colony optimization, or particle swarm optimization), EAs do not guarantee to find the Pareto optimal set but try to find a nondominated set whose vectors are as close as possible to the Pareto optimal front. On the other hand, EAs are particularly well-suited to solve MOPs because they work in parallel with a set of potential solutions (i.e., the population). This feature makes them to find several solutions of the Pareto optimal set (or a good approximation) in a single run. Furthermore, EAs are less susceptible to the shape or continuity of the Pareto front, than traditional mathematical programming techniques.

Some researchers [19, 28, 47, 86] have identified some limitations of traditional mathematical programming algorithms to solve MOPs. Some of them are the following:

1. We need to run many times those algorithms to find several elements of the Pareto optimal set.

2. Many of them need domain knowledge about the problem to be solved.

3. Some of those algorithms are sensitive to the shape or continuity of the Pareto front.

Single objective EAs and MOEAs share a similar structure. The major difference is the fitness assignment mechanism since a MOEA deals with fitness vectors of dimension k ($k \geqslant 2$). As pointed out by different authors [128, 22],

finding an approximation to the Pareto front is by itself a bi-objective problem whose objectives are:

- Minimize the distance of the generated vectors to the Pareto optimal front, and
- Maximize the diversity of the achieved Pareto front approximation.

Therefore, the fitness assignment must consider these two objectives. Algorithm 1 describes the basic structure of a multiobjective evolutionary algorithm.

Algorithm 1 Pseudocode of a Multiobjective Evolutionary Algorithm.

1: $t \leftarrow 0$
2: Generate an initial population $P(t)$
3: **while** the stopping criterion is not fulfilled **do**
4: Evaluate the objective vector $\mathbf{f}$ for each individual in $P(t)$
5: Assign a fitness for each individual in $P(t)$
6: Select from $P(t)$ a group of parents $P'(t)$ preferring the fitter ones
7: Recombine individuals of $P'(t)$ to obtain an offspring population $P''(t)$
8: Mutate individuals in $P''(t)$
9: Combine $P(t)$ and $P''(t)$ and select the best individuals to get $P(t+1)$
10: $t \leftarrow t+1$

Usually, the initial population is generated in a random manner. However, if we have some knowledge about the characteristics of a good solution, it is wise to use this information to create the initial population. The fitness assignment requires ranking the individuals according to a preference relation and then, assigning a scalar fitness value to each individual using such rank. The selection for reproduction (line 6) is carried out as in the single objective case, for instance, using tournament selection. In contrast, the selection for survival (line 9), intended to maintain the best solutions so far (i.e., elitism), uses a preference relation to remove some solutions and maintain the population size constant. To ensure diversity of the approximation set, the selection mechanism is also based on a density estimator of the objective function space.

3.2.1 *Key Elements of a MOEA*

In this section we will describe in detail the most important elements of a MOEA that should to be taken into account for its design.

3.2.1.1 *Fitness Assignment*

In a MOEA we need an additional process to transform a fitness vector into a scalar value. Mainly, there are three schemes to carry out this process, namely: criterion-based, aggregation-based, and preference-based.

CRITERION-BASED. This approach alternately chooses each of the objective functions during the selection stage. That is, to select an individual or group of individuals only one objective is considered. For instance, the Vector Evaluated Genetic Algorithm (VEGA) [100] divides the population into k equally-sized subpopulations and a different objective is used to assign fitness within each subpopulation.

AGGREGATION-BASED. In this method, the objective functions are aggregated or combined into a single scalar value. During the optimization process, the parameters are systematically varied to generate different elements of the Pareto optimal set. Note that, although an aggregation-based approach can be formulated as a preference relation, the solutions are not compared in objective function space. That is to say, vectors are mapped from $\mathbb{R}^k$ to $\mathbb{R}$ before the comparison.

PREFERENCE-BASED. In this scheme a preference relation is used to induce a partial order of the population in objective function space. Then, a scalar score (rank) is assigned to each solution based on how the solution compares with respect to the other solutions. For example, dominance-rank schemes count the number of individuals by which a given individual is dominated. In the dominance-count schemes the fitness of an individual corresponds to the number of individuals that it dominates. As we noticed in a previous section, Pareto dominance is the preference relation most commonly adopted in MOEAs.

3.2.1.2 *Elitism*

Elitism is the mechanism intended to prevent the loss of the best solutions found during the search due to stochastic effects. This concept plays a major role in modern MOEAs since, along with mutation, guarantees global convergence. In multiobjective optimization, the implementation of elitism is more complex than in single objective optimization. Since we count with limited memory resources, if more nondominated solutions arise than those that can be stored, then some good solutions have to be discarded. Therefore, the adopted elitist strategy determines if the MOEA is globally convergent or not. Currently, we can mainly distinguish two approaches to implement elitism. One of them is to combine the old and the new populations, and then use a deterministic selection to preserve the best solutions in the

next generation [32]. The other approach is to maintain an external set of individuals called "archive" that stores the nondominated solutions found during the search process. Figure 5 illustrates these two approaches.

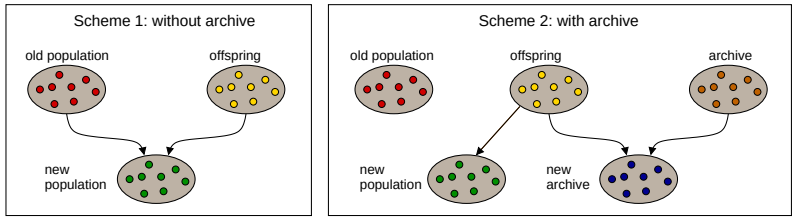


Figure 5: Schemes to implement elitism.

3.2.1.3 *Density Estimators*

One of the goals in MOEAs is to obtain a set of nondominated solutions which are well distributed along the Pareto front. In the following, we describe some techniques to maintain diversity in the population.

FITNESS SHARING. The goal of fitness sharing [56] is to form and maintain subpopulations (niches) distributed over objective function space. The idea is to consider fitness as a resource that needs to be shared among individuals in the same niche. Thus, the larger the number of individuals in the niche, the smaller the fitness assigned to each individual. Formally, the shared fitness f_{s_i} of individual i is defined by:

$$f_{s_i} = \frac{f_i}{\sum_{j=1}^{N} \phi(d_{ij})}, \tag{3.2}$$

where f_i is the fitness of individual i, and $\phi(d_{ij})$ is the sharing function, defined by:

$$\phi(d_{ij}) = \begin{cases} 1 - \left(\frac{d_{ij}}{\sigma_{sh}}\right) & , d_{ij} < \sigma_{share} \\ 0 & , \text{otherwise} \end{cases} \tag{3.3}$$

where σ_{share} is the niche radius and d_{ij} is the distance between individuals i and j.

HYPERGRIDS. A hypergrid divides objective function space in regions called hypercubes. Each nondominated solution occupies a hypercube as it is shown in Figure 6. The idea is only to accept nondominated solutions belonging to underpopulated hypercubes. Although the number of divisions in the hypergrid in each dimension is constant, the position

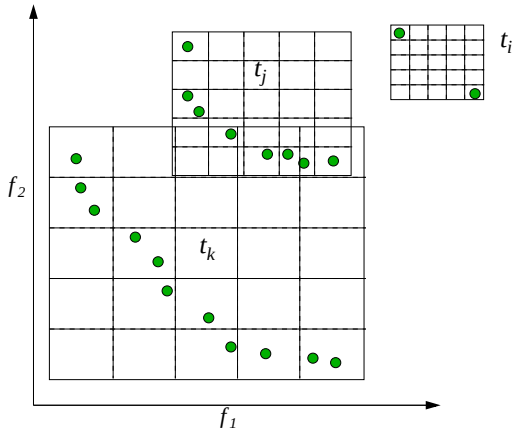


Figure 6: Hypergrid to maintain diversity in the archive. In this figure we can see how the hypergrid changes during the search.

and extension of the grid can be adapted during the search process (see Figure 6).

CLUSTERING The objective of a clustering algorithm is to partition a set of points in such a way that: (a) each group contains points very similar to each other; and (b) the points of one group are very different from the points of other groups. In a MOEA we use clustering to preserve diversity in the archive and reduce its size. This process consists of three steps (Figure 7):

1. Partition the archive using a clustering algorithm.
2. Select a representative individual of each group, i.e., the centroid.
3. Remove all the other individuals in the cluster.

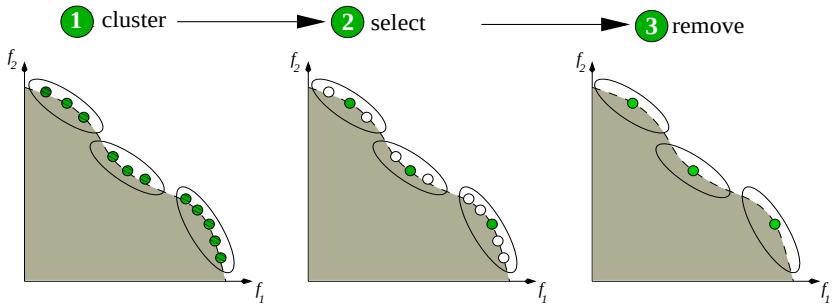


Figure 7: Clustering technique to maintain diversity in the archive.

3.2.2 *MOEAs based on Pareto Optimality*

MULTI-OBJECTIVE GENETIC ALGORITHM (MOGA). Fonseca and Fleming [48] proposed the Multiobjective Genetic Algorithm (MOGA), which is based on the scheme proposed by Goldberg [55]. This algorithm ranks the population based on nondominance. Thus, the rank of an individual $\mathbf{x}^i$ at generation t is equal to the number of solutions, $p(\mathbf{x}^i, t)$, by which it is dominated, namely $\text{rank}(\mathbf{x}^i, t) = 1 + p(\mathbf{x}^i, t)$.

NON-DOMINATED SORTING GENETIC ALGORITHM (NSGA). Srinivas and Deb [107] implemented Goldberg's idea in a more straightforward way. The Nondominated Sorting Genetic Algorithm (NSGA) ranks the population in different nondominated layers or fronts with respect to nondominance. The first front (the best ranked) is composed by the nondominated individuals of the current population. The second front is the set composed of the nondominated individuals excluding individuals in the first rank. In general, each front is computed only using the unranked individuals in the population. Deb et al. [32] later proposed a new version of the algorithm called NSGA-II. This algorithm improves the efficiency of the original NSGA by reducing the number of times that the population needs to be ranked, and incorporates an elitist selection scheme, as well as a crowded-comparison operator.

NICHED PARETO GENETIC ALGORITHM (NPGA). Horn and Nafpliotis [58] combined tournament selection and Pareto dominance. In this method two individuals are randomly selected in order to compare them against a subset of the population. The individual that results nondominated is selected as a parent. On the other hand, if both individuals are dominated or nondominated, then the winner is decided by means of a fitness sharing function.

STRENGTH PARETO EVOLUTIONARY ALGORITHM (SPEA). The Strength Pareto Evolutionary Algorithm (SPEA) was developed by Zitzler and Thiele [128] as a way to combine the most successful techniques of different MOEAs. SPEA uses the individuals stored in the archive to rank the individuals in the current population. For each individual in the archive a value is computed (referred to as *strength*) that is equal to the number of population members that are dominated by the corresponding archive member. Finally, the fitness of each individual $\mathbf{x}$ is computed by adding up the strength of all archive members that dominate $\mathbf{x}$. This scheme tries to guide the search towards the Pareto front and, at the same time, preserves the diversity of the nondominated solutions.

4 MANY-OBJECTIVE OPTIMIZATION

Contents

4.1 Notions of Conflict Among Objectives 27
4.2 Sources of Difficulty When Solving Many-Objective Problems 29
4.3 Current Approaches to Deal with Many-Objective Problems 32

Since the first implementation of a Multiobjective Evolutionary Algorithm (MOEA) in the mid 1980s [101], a wide variety of new MOEAs have been proposed, gradually improving in both their effectiveness and efficiency to solve Multiobjective Optimization Problems (MOPs) [22]. However, most of these algorithms have been evaluated and applied to problems with only two or three objectives, in spite of the fact that many real-world problems have more than three objectives [51, 62].

Recent experimental [61, 117, 92] and analytical [113, 69] studies have shown that MOEAs based on Pareto optimality [91] scale poorly in MOPs with a high number of objectives (4 or more). Although this limitation seems to affect only to Pareto-based MOEAs, optimization problems with a large number of objectives (also known as many-objective problems) introduce some difficulties common to any other multi-objective optimizer.

The goal of this chapter is presenting a general view of the difficulties posed by many-objective problems for Pareto-based MOEAs. Specifically, we present a review of the potential sources of difficulty currently found in the specialized literature. Likewise, we present a brief review of the current proposals to deal with these sources of difficulty. Since some of these proposals are based on conflict information among the objectives, first some definitions of conflict are provided.

4.1 Notions of Conflict Among Objectives

One important condition of a multiobjective problem is the conflict among their objectives. If the objectives have no conflict among them, then we could solve the problem optimizing each objective function independently. Nonetheless, it has been found that in some problems, although a conflict ex-

ists elsewhere, some objectives behave in a non-conflicting manner. Although different authors have proposed definitions for conflict among objectives (see, e.g. [16, 93, 112, 12]), in this thesis we only present the employed or relevant to this document.

Definition 15. *Let be $S_{\mathcal{X}}$ a subset of $\mathcal{X}$, then, according to Carlsson and Fullér, two objectives can be related in the following ways (assuming minimization):*

1. f_i *is in conflict with* f_j *on* $S_{\mathcal{X}}$ *if* $f_i(\mathbf{x}^1) \leqslant f_i(\mathbf{x}^2)$ *implies* $f_j(\mathbf{x}^1) \geqslant f_j(\mathbf{x}^2)$ *for all* $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.

2. f_i *supports* f_j *on* $S_{\mathcal{X}}$ *if* $f_i(\mathbf{x}^1) \geqslant f_i(\mathbf{x}^2)$ *implies* $f_j(\mathbf{x}^1) \geqslant f_j(\mathbf{x}^2)$ *for all* $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.

3. f_i *and* f_j *are independent on* $S_{\mathcal{X}}$*, otherwise.*

In the cases 2 and 3, those objectives are also called nonconflicting objectives. When $S_{\mathcal{X}} = \mathcal{X}$, it is said that f_i is in conflict with (or supports) f_j globally. However, in many MOPs the relation among the objectives changes when comparing different subsets of $\mathcal{X}$. Figure 8 shows an example in which two functions are in conflict in some subsets of $\mathcal{X}$, while in others, they support each other.

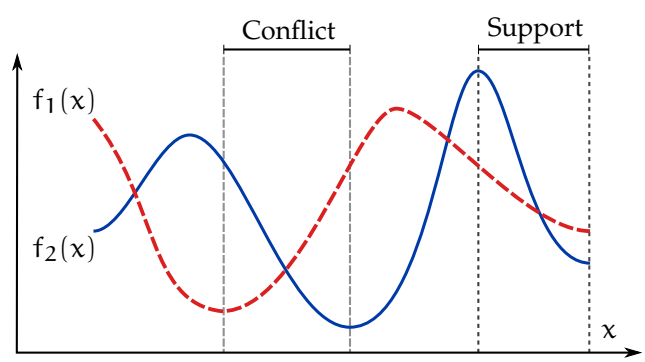


Figure 8: Two objective functions can be in conflict in some subsets of the feasible space, and can be supportive in other subsets.

Nonconflicting objectives are also known as nonessential or redundant objectives because, as pointed out by Gal and Hanne [52], when a nonconflicting objective is removed from the original set of objectives, the resulting Pareto front does not change. Based on the notion of nonessential objectives, Brockhoff and Zitzler [12] proposed a conflict definition that verifies whether the Pareto dominance relation changes when some objectives are removed, or not. The Pareto dominance relation induced by a given set F of objectives is defined as $\preceq_F = \{(\mathbf{x}, \mathbf{y}) \mid \forall f_i \in F : f_i(\mathbf{x}) \leqslant f_i(\mathbf{y}) \wedge \exists i : f_i(\mathbf{x}) < f_i(\mathbf{y})\}$.

Definition 16. *Let* $F_1, F_2 \subseteq \Phi$ *be two subsets of objectives, where* Φ *is the entire set of objectives* $\Phi = \{f_1, f_2, \ldots, f_k\}$*. Then, we call* F_1 *nonconflicting with* F_2 *iff* $(\preceq_{F_1} \subseteq \preceq_{F_2}) \wedge (\preceq_{F_2} \subseteq \preceq_{F_1})$*.*

In other words, F_1 and F_2 are called nonconflicting if and only if the corresponding relations $\preceq_{F_1}$ and $\preceq_{F_2}$ are identical, but not necessarily $F_1 = F_2$. The nonconflicting definition is useful since if F and $F' \subset F$ are nonconflicting, then we can replace F with F' and obtain the same Pareto optimal front. The objectives in F' are then called essential objectives, whereas the objectives in $F \setminus F'$ are known as nonessential or redundant objectives.

In practice, however, one is often interested in a further objective reduction at the cost of slight changes in the dominance relation. Brockhoff and Zitzler proposed a measure, based on the additive ϵ-dominance relation, to compute the change between two dominance relations. The ϵ-dominance relation induced by a set F is defined by $\preceq_F^{\epsilon} = \{(\mathbf{x}, \mathbf{y}) \mid \forall f_i \in F : f_i(\mathbf{x}) - \epsilon \leqslant f_i(\mathbf{y}) \wedge \exists i : f_i(\mathbf{x}) - \epsilon < f_i(\mathbf{y})\}$.

Definition 17. *Let* $F_1, F_2 \subseteq F$ *be two subsets of objectives, where* F *is the entire set of objectives. Then, we call* F_1 *δ-nonconflicting with* F_2 *iff* $(\preceq_{F_1} \subseteq \preceq_{F_2}^{\epsilon}) \wedge (\preceq_{F_2} \subseteq \preceq_{F_1}^{\epsilon})$*.*

In this case, if an objective subset $F' \subset F$ is δ-nonconflicting with F, then we can omit all objectives in $F \setminus F'$ without causing a larger error than δ in the omitted objectives.

4.2 Sources of Difficulty When Solving Many-Objective Problems

4.2.1 *Deterioration of the Search Ability*

One of the reasons for this problem is that the proportion of nondominated solutions (i.e., equally good solutions) in a population increases rapidly with the number of objectives [42]. Figure 9 shows the nondominated regions with respect to a given solution z. As we can see, for 2 objectives 1/2 of the search space is composed of nondominated regions, whereas for 3 objectives 3/4 of the search space is composed of nondominated regions.

In general, the expression to denote the proportion, e, of nondominated regions and the whole search space [42] is given by $e = (2^k - 2)/2^k$, where k is the number of objectives. This proportion goes to infinity when the number of objectives approaches infinity. According to Bentley et al. [6] the number of nondominated k-dimensional vectors on a set of size n is $O(\ln^{k-1} n)$. As a consequence, in many-objective problems, the selection of solutions is carried out almost at random or guided by diversity criteria. In fact, Mostaghim and Schmeck [89] have shown that a random search

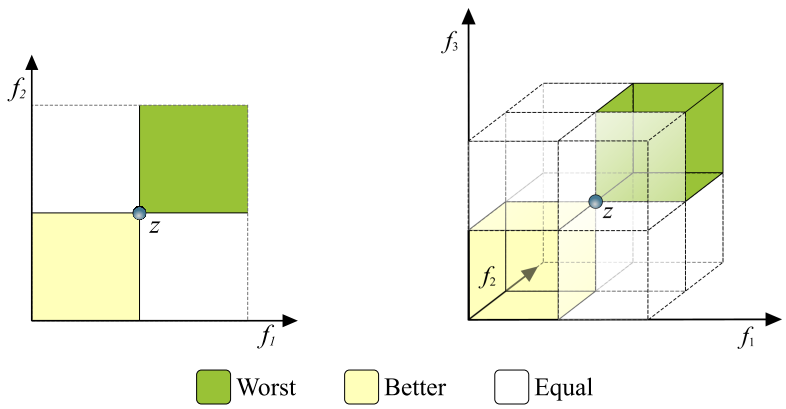


Figure 9: Nondominated regions with respect to a given solution z.

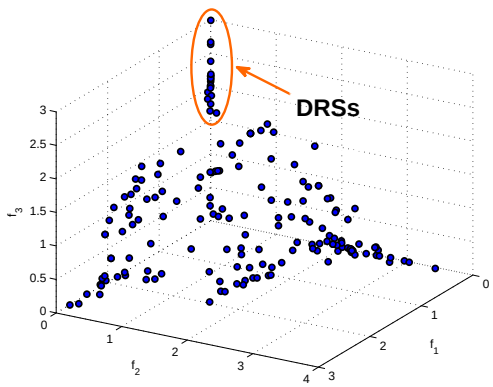


Figure 10: Illustration of some Dominance Resistant Solutions in problem DTLZ2.

optimizer achieves better results than the NSGA-II [32] in a problem with 10 objectives. Another possible source of the poor scalability of MOEAs is the increment of the number of Dominance Resistant Solutionss (DRSs) as the number of objectives is increased [71, 57, 34, 59]. Dominance resistant solutions are those with a poor value in at least one of the objectives, but with near optimal values in the others. In other words, those are nondominated solutions, but far from the Pareto optimal front. Figure 10 shows an example of DRSs in the well-known test problem DTLZ2. Although solutions marked as DRSs seem to be dominated by some solution in the lower part of the circled solutions, they achieve marginal improvements in objectives f_1 or f_2, and therefore, they are nondominated solutions, but having poor values in objective f_3, though.

4.2.2 *Dimensionality of the Pareto front*

Due to the 'curse of dimensionality', the number of points required to represent accurately a Pareto front increases exponentially with the number of objectives. Formally, the number of points necessary to represent a Pareto front with k objectives and resolution r is bounded by $O(kr^{k-1})$ (e.g., see [103]). This expression is derived assuming that each solution is contained inside a hypercube to preserve an even distribution. As can be seen in Figure 11, the number of hypercubes determines the resolution of the Pareto front, i.e., r is the number of hypercubes per dimension. An example of the shortest connected and non-degenerated 2-objective Pareto front (the straight line) is shown on the left side of Figure 11. The figure also shows a bound for the largest Pareto front for 2 and 3 objectives. In general, the bounding Pareto front is formed by k hyperplanes containing r^{k-1} hypercubes each (see, for example, the 3-objective case shown on the right side of Figure 11). This way, the maximum number of points of a 2-objective Pareto front with resolution $r = 6$ is $2 \cdot 6^{2-1} = 12$, whereas for 3 objectives and $r = 5$ is $3 \cdot 5^{3-1} = 75$. Table 1 shows the maximum number of points required to represent a Pareto front for different number of objectives using a resolution of $r = 25$, which is a conservative number considering that a resolution of $r = 50$ is usually used in several studies to obtain 100 solutions in 2-objective problems. Notwithstanding, for 5 objectives, we would require approximately 2 million points to represent a Pareto front with resolution $r = 25$. There are other formulations leading to a similar exponential expression with respect to k. For example, using the concept of ϵ-dominance, Laumanns et al. [74] give a similar exponential bound for the size of an approximation of a Pareto front.

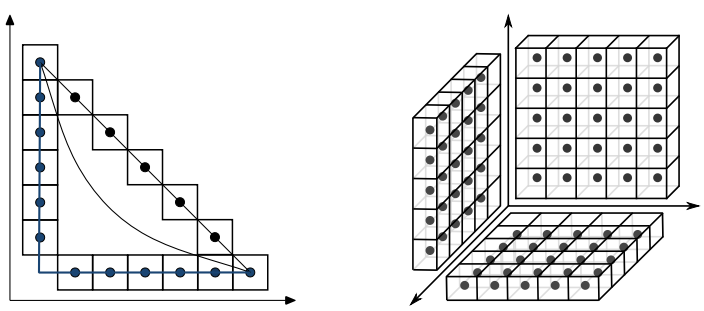


Figure 11: Number of points required to represent a Pareto front with a resolution r, i.e., the number of hypercubes per dimension.

This poses some difficulties to solve many-objective problems. The most important one is the number of function evaluations required to deal with a large number of solutions. This is a serious issue since some real-world

problems (e.g., [18]), due to time constraint reasons, have a small budget of function evaluations. In fact, there is an important research effort towards designing MOEAs that generate good approximations of the Pareto front using less than 1000 function evaluations (e.g., [41, 68, 54, 124]). Other challenges are related to the design of both data structures to efficiently manage that number of points, and density estimators to achieve an even distribution of the solutions along the Pareto front. Unfortunately, even if we could efficiently obtain an accurate approximation of the Pareto front, the selection of one solution among such a huge number of solutions would be a very difficult task for the decision maker (DM).

4.2.3 *Visualization of the Pareto front*

Clearly, with more than three objectives it is not possible to plot the Pareto front as usual. This is a serious problem since visualization plays a key role for a proper decision making process. Parallel coordinates [119] and self-organizing maps [90] are some of the methods proposed to ease decision making in high dimensional problems. However, more research in this field is still required.

4.3 Current Approaches to Deal with Many-Objective Problems

Besides studies about the scalability of Pareto-based MOEAs, in the current literature we can find several proposals to overcome those scalability issues. The most common approaches can be categorized as follows:

1. Adopt or propose a preference relation that yields a finer solution ordering than the one yielded by Pareto optimality. In other words, these relations are able to further rank nondominated solutions. In addition, most of these preference relations share the property that their optimal set of solutions is a subset of the Pareto optimal set.

k	**Points**
2	50
4	62 500
5	1 953 125
7	1 708 984 375

Table 1: Bound for the number of points required to represent a Pareto front with resolution $r = 25$.

Therefore, these techniques can also be used as a remedy to cope with the dimensionality of Pareto fronts in many-objective problems.

2. Reduce the number of objectives of the problem during the search process or, a posteriori, once an approximation of the Pareto front has been found [31, 12, 78]. The main goal of this kind of reduction techniques is to identify the nonconflicting objectives (at least to a certain extent) in order to discard them.

3. Incorporation of preference information interactively during the search. By incorporating preferences we can cope with many-objective problems in two aspects. First, the search can be focused on the decision maker's region of interest, avoiding this way, the evaluation of a huge number of solutions. Second, the preference relations usually used in interactive methods help to deal with a large number of objectives since they are able to rank incomparable nondominated solutions.

In the remainder of this section some of the most relevant approaches to deal with many-objective problems are presented.

4.3.1 *Preference Relations to Deal with Many-Objective Problems*

Bentley and Wakefield [7] proposed the Average Ranking (AR) and the Maximum Ranking (MR) preference relations. The AR relation computes, for each solution, a different rank considering each objective independently. The final rank is obtained by summing up the ranks on each objective. In turn, the MR relation takes the best rank as the global rank. Clearly, this method favors extreme solutions, i.e., solutions with high performance in some of the objectives, although with poor overall performance. Although it is less evident, the average ranking relation also favors extreme solutions.

Average and Maximum ranking.

In the *favour relation*, proposed by Drechsler et al. [37], a vector $\mathbf{z}^1$ is preferred to vector $\mathbf{z}^2$ with respect to the favour relation ($\mathbf{z}^1 \prec_{\text{favour}} \mathbf{z}^2$), if and only if:

Favour preference relation.

$$\sharp\{i : z_i^1 < z_i^2, 1 \leqslant i \leqslant k\} > \sharp\{j : z_j^1 > z_j^2, 1 \leqslant j \leqslant k\}.$$

In other words, the favoured vector is that which outperforms the other one in more objectives. Unfortunately, this relation emphasizes extreme solutions.

The Preference Order Relation (POR), developed by di Pierro [36], is based on the concept of *efficiency of order* proposed by Das [27], which states that: A vector $\mathbf{z}^*$ is efficient of order q if it is not dominated by any other vector in all the $\binom{k}{q}$ objective subsets of size q.

Preference order relation.

Based on that definition, it is said that vector $\mathbf{z}^1$ is preferred to vector $\mathbf{z}^2$ ($\mathbf{z}^1 \prec_{\text{POR}} \mathbf{z}^2$), if and only if, for some integer q and $\forall I \subseteq \{1, 2, \ldots, k\}$ such that $|I| = q$:

$$z_i^1 \leqslant z_i^2 \ \ \forall i \in I, \text{ and } \exists i \in I: \ z_i^1 < z_i^2.$$

In other words, if $\mathbf{z}^1$ and $\mathbf{z}^2$ do not dominate each other, then the solutions are compared in a lower-dimensional space in order to break the tie.

Sato, Aguirre and Tanaka [98] proposed a preference relation to control the dominance area of solutions. This method controls the degree of expansion or contraction of the dominance area by modifying each objective vector $\mathbf{z}$ with the expression:

Expansion preference relation.

$$z_i' = \frac{r \cdot \sin(\omega_i + s_i \cdot \pi)}{\sin(s_i \cdot \pi)} \ \ \forall i = 1, 2, \ldots, k,$$

where $\mathbf{s} \in \mathbb{R}^k$ is a user-defined vector, $r = \|\mathbf{z}\|$, and ω_i is the declination angle between $\mathbf{z}$ and the axis of f_i.

If the user adopts values $s_i < 0.5$ ($\forall i = 1, 2, \ldots, k$), the dominance area is expanded and produces a more fine-grained ranking of solutions which would strengthen the selection process. Thus, we can say that vector $\mathbf{z}$ is preferred to vector $\mathbf{y}$ with respect to the *expansion relation* ($\mathbf{z} \prec_{\text{expansion}} \mathbf{y}$), if and only if $\mathbf{z}' \prec_{\text{pareto}} \mathbf{y}'$.

Farina and Amato [43] proposed an alternative relation which takes into account the number of improved objectives between two solutions. This relation employs three quantities, $n_b(\mathbf{x}_1, \mathbf{x}_2), n_e(\mathbf{x}_1, \mathbf{x}_2)$ and $n_w(\mathbf{x}_1, \mathbf{x}_2)$, which denote the objectives where $\mathbf{x}_1$ is better, equal or worse than $\mathbf{x}_2$, respectively. Using these quantities the concepts of $(1-k)$-$\mathtt{Dominance}$ and k-$\mathtt{Optimality}$ are defined. A solution $\mathbf{x}_1$ $(1-k)$-dominates $\mathbf{x}_2$ if and only if

k-optimaltiy relation.

$$\begin{cases} n_e(\mathbf{x}_1, \mathbf{x}_2) < M \\ n_b(\mathbf{x}_1, \mathbf{x}_2) \geqslant \frac{M - n_e}{k+1} \end{cases}$$

In a similar way to Pareto optimality, a solution $\mathbf{x}^*$ is k-$\mathtt{optimum}$ if and only if there is no $\mathbf{x}$ in the decision variable space such that $\mathbf{x}$ k-dominates $\mathbf{x}^*$.

4.3.2 *Objective Reduction Approaches*

Deb and Saxena [31] proposed a method for reducing the number of objectives based on principal component analysis. The main assumption is that if two objectives are negatively correlated (taking the generated Pareto front as the data set), then these objectives are in conflict with each other.

To determine the most conflicting objectives (i.e., the most essential), the authors analyze in turn the eigenvectors (i.e., the principal components) of the correlation matrix. That is, by picking the most-negative and the most-positive elements from the first eigenvector, we can identify the two most important conflicting objectives. To aggregate more objectives to the set of essential objectives the remainder of the eigenvectors are analyzed in a similar way until the cumulative contribution of the eigenvalues exceeds a threshold cut (TC). This method is incorporated into an iterative scheme which uses a multiobjective optimizer (the actual implementation uses Non-dominated Sorting Genetic Algorithm II (NSGA-II) [32]) to obtain a reduced objective set containing only the non-redundant objectives according to the analysis of the eigenvectors. In this scheme, the evolutionary multiobjective optimizer is first run and then, the correlation analysis is carried out to obtain a reduced set of objectives. This process is repeated using the new reduced set of objectives. The process stops when the current subset is equal to the subset generated in the previous iteration.

Brockhoff and Zitzler [12] defined two kinds of objective reduction problems and two corresponding algorithms to solve them. The problems proposed are the following:

1. **The δ-MOSS problem**. Given a multiobjective optimization problem, the δ-minimum objective subset problem is defined as follows.
   - **Input**: A Pareto front approximation of the MOP and a $\delta \in \mathbb{R}$.
   - **Task**: Compute the minimum objective subset $F' \subseteq F$ such that F' is δ-nonconflicting with F.
2. **The k-EMOSS problem**. Given a multiobjective optimization problem, the problem of finding the minimum objective subset of size k with minimum error is defined as follows.
   - **Input**: A Pareto front approximation of the MOP and a $k \in \mathbb{R}$.
   - **Task**: Compute an objective subset $F' \subseteq F$ with size $|F'| \leqslant k$, such that F' is δ-nonconflicting with F with the minimum possible δ.

Since both problems are $\mathcal{NP}$-hard, the authors proposed both an exact and a greedy algorithm for each of them. The exact algorithms for both problems have time complexity $O(m^2 k \cdot 2^s)$, where m is the size of the given nondominated set and k is the number of objectives. On the other hand, the greedy algorithm for the δ-MOSS problem has time complexity $O(\min\{m^2k^3, m^4k^2\})$, while the greedy algorithm for the k-EMOSS problem has time complexity $O(m^2k^3)$.

4.3.3 *Preference Incorporation Approaches*

Among the earliest attempts to incorporate preferences in a MOEA, we can find Fonseca and Fleming's proposal [48, 50]. This proposal consisted of extending the ranking mechanism of Multiobjective Genetic Algorithm (MOGA) [49] using a new relation called preferability relation. This relation accommodates goal information (equivalent to a reference point in other methods) and priorities in a single preference relation. The DM should define goal values and group objectives according to its priority. Using the preferability relation two solutions are first compared in terms of the group of objectives with the highest priority. If the objectives of both solutions meet all their goal values or, contrarily, violate some or all of their goal values in a similar way, the next priority objective group is considered. This process continues until reaching the lowest priority group, where solutions are compared using the Pareto dominance relation. By setting particular goals and priorities the authors derived the following special cases: the usual Pareto relation, lexicographic relation, constrained optimization, and goal programming. One disadvantage of this relation is that is affected by the feasibility of the goal provided by the decision maker. If the given goal is far away from the feasible region, then the solutions will be mainly compared in terms of the objective priorities, reducing the relation to the lexicographic relation. In addition, if two solutions either do or do not meet their goals, the relation does not take into account the degree of under- or over-attainment.

Deb [29] proposed a technique to transform goal programming problems into multiobjective optimization problems which are then solved using a MOEA. In goal programming the DM has to assign goals that wishes to achieve for each objective, and these values are incorporated into the problem as additional constraints. The objective function then attempts to minimize the absolute deviations from the goals to the objectives. Unfortunately, as the previous method, this approach is sensitive to the feasibility of the goal values. If the goal is contained in the feasible space, it could prevent the generation of a better solution. On the other hand, if the goal is located far away from the feasible space, the effect of the method is practically nonexistent.

Branke et al. [8] proposed an approach called Guided MOEA which models DM's preferences using the trade-off between pairs of objectives. That is, for each pair of objectives, the DM has to indicate how many units of objective f_i he/she is willing to trade-off in exchange of one unit of objective f_j, and vice versa. The authors determine a new preference relation that uses these pairs of trade-offs. By setting appropriate trade-offs it is possible to focus the search to any subregion of the Pareto front. The main drawback of this approach is the difficulty to determine the trade-offs as the number

of objectives increases, since the DM has to provide $k(k-1)$ trade-offs. Furthermore, this method is only applicable in problems with a convex Pareto front.

Cvetković and Parmee [25, 24] proposed the use of binary preference relations that can be expressed qualitatively (i.e., using words such as "less important" or "don't care"). These preferences are translated to quantitative terms (i.e., weights) to guide the search towards certain region of interest of the Pareto front. The weights generated can be used with a simple aggregating approach (i.e., a sum of weights) or with Pareto ranking. In the second case, the weights are used to modify the definition of nondominance used by the ranking scheme of the MOEA. There may be some practical issues to take into account if this approach is used interactively, since the DM is asked a considerably high number of questions to make it possible to translate qualitative preferences into quantitative values. This could become too expensive (computationally speaking) if done repeatedly along the evolutionary process.

More recently, Deb and Sundar [31] incorporated a reference point approach into the NSGA-II [33]. They introduced a modification in the crowding distance operator in order to select from the last nondominated front the solutions that would take part of the new population. They used the Euclidean distance to sort and rank the population accordingly (the solution closest to the reference point receives the best rank). This method was designed to take into account a set of reference points. The drawback of this scheme is that it only guarantees weak Pareto optimality. That is to say, besides Pareto optimal solutions, the method might generate some weakly Pareto optimal solutions, particularly in MOPs with disconnected Pareto fronts. A similar approach was also proposed by Deb and Kumar [30], in which the light beam search procedure [65] was incorporated into the NSGA-II. Similar to the previous approach, they modified the crowding operator to incorporate DM's preferences. They used a weighted achievement function to assign a crowding distance to each solution in each front. Thus, the solution with the least distance will have the best crowding rank. Like in the previous approach, this algorithm finds a subset of solutions around the optimum of the achievement function adopting the usual outranking relation. A vector $\mathbf{z}^1$ outranks vector $\mathbf{z}^2$ if $\mathbf{z}^1$ is considered to be at least as good as $\mathbf{z}^2$. In [65] three kinds of thresholds are defined to determine if one solution outranks another one, namely, indifference, preference, and veto threshold. However, in [30] the veto threshold is the only one used. This relation depends on the crowding comparison operator. In contrast, the new preference relation presented in this work does not depend on external methods, and, therefore, it can be used in every Pareto-based MOEA.

Recently, Thiele et al. [114] proposed a variant of the Indicator-Based Evolutionary Algorithm (IBEA) [127], in which preference information is incorporated by means of an achievement scalarization function. The basic idea is to divide the original indicator value (which is to be maximized) by the achievement value (which is to be minimized). Thus, solutions with a smaller achievement value will be preferred since the modified indicator value is larger. The approach of [114] is similar to the one proposed in Chapter 8. However, their approach was designed for IBEAs [127]. In contrast, our approach can be used both in IBEAs and Pareto-based MOEAs. Later, the new IBEA of Thiele et al. was used in [45] in order to approximate the entire Pareto front by defining several reference points.

5

OBJECTIVE REDUCTION TO DEAL WITH MANY-OBJECTIVE PROBLEMS

Contents

5.1 Proposed Objective Reduction Algorithms 39
5.2 Comparative Study 45

In some problems, it is possible that although a conflict exists between some objectives, others behave in a non-conflicting manner. In this case, we can discard these objectives to obtain a lower-dimensional problem. An objective reduction technique can be helpful both in the decision making process and during the search. That is, the decision maker would have to analyze fewer objectives and a lower number of nondominated solutions. On the other hand, Pareto-based optimizers can be improved if the number of objectives is reduced during the search.

In this section we present an algorithm to reduce the number of objectives of a given problem by removing the non-conflicting objectives (also called, non-essential or redundant objectives). The algorithm is based on an unsupervised feature selection technique proposed by Mitra et al. [88] where the goal is to identify the subset of essential objectives of a problem. We developed two variants of the algorithm, namely: *i*) an algorithm that finds the minimum subset of objectives with the minimum error possible, and *ii*) an algorithm that finds a subset of objectives of a given size and that yields the minimum error possible.

5.1 Proposed Objective Reduction Algorithms

We propose an unsupervised feature selection technique to identify the most conflicting objectives in order to reduce the number of objectives of an optimization problem. The technique employed was originally proposed by Mitra et al. [88] to preprocess data prior to classification by selecting a subset of the original features.

As in Deb and Saxena's approach [31] (see section 4.3.2), this technique also uses a correlation matrix to measure the conflict between each pair of objectives. This matrix is computed using an approximation set of the

Pareto front generated by some optimizer, for instance, a multiobjective evolutionary algorithm.

The original algorithm as proposed by Mitra et al., used $1 - |\rho(x, y)|$ (where $\rho(x, y)$ is the correlation coefficient between random variables x and y) as the similarity measure between features, which only determines the degree of correlation (positive or negative) between features (objectives in our context) x and y. However, in the case of objective selection we are interested in measuring only the negative correlation between objectives in an approximation of the Pareto Front. For this purpose, we used $1 - \rho(x, y) \in [0, 2]$ instead. Thus, a result of zero indicates that objectives x and y are completely positively correlated and a value of 2 indicates that x and y are completely negatively correlated.

A negative correlation between a pair of objectives means that one objective increases while the other decreases and vice versa. On the other hand, if the correlation is positive, then both objectives increase or decrease at the same time. This way, we could interpret that the more negative the correlation between two objectives, the more the conflict between them.

We propose the following algorithms to identify the essential objectives in a multiobjective problem:

1. An algorithm that finds a minimum subset of non-redundant objectives with the minimum error possible. We denote this algorithm as MOSSA.

2. An algorithm that finds a k-size subset of non-redundant objectives, yielding the minimum error possible. We refer to this algorithm as KOSSA.

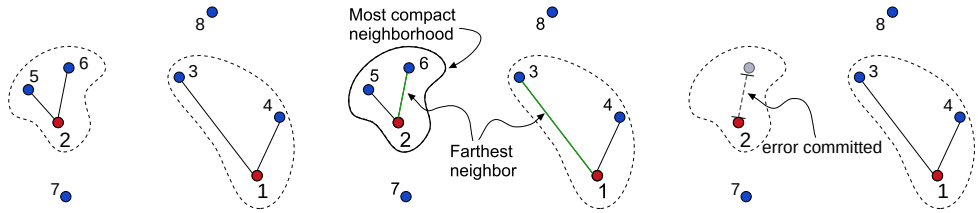
The central part of the two algorithms can be divided in three steps:

1. Divide the objective set into homogeneous neighborhoods of size q around each objective. The conflict between objectives takes the role of the distance. That is, the more the conflict between two objectives, the more distant they are in the "conflict" space. Figure 12(a) shows only two neighborhoods of a hypothetical situation with eight objectives and $q = 2$.

2. Select the most compact neighborhood. That is, the neighborhood with the minimum distance to its q-th neighbor (i.e., the farthest one in the neighborhood). Figure 12(b) shows the farthest neighbor for each of the two neighborhoods. As it can be seen in the example, the neighborhood on the left is the most compact one.

3. Retain the center of that neighborhood and discard its q neighbors (the objectives with least conflict in the current set). In this process, the

distance to the q-th neighbor can be thought of as the error caused by removing the q objectives (see Figure 12(c)).



(a) Divide the objective set into neighborhoods around each objective.

(b) Select the most compact neighborhood.

(c) Retain the center and remove the neighbors.

Figure 12: Basic strategy of the objective reduction method employed.

The process described above is repeated until some stopping criterion is met. It is important to mention that the size of the neighborhoods q is a parameter that is reduced during the search.

Algorithm MOSSA is described in Figure 13, where $r_{i,q}$ is an entry of the correlation matrix denoting the conflict between objective f_i and its q-th nearest-neighbor. Since the correlation matrix is computed at the beginning of algorithm MOSSA, we do not have to compute the $r_{i,q}$ value each time we need it, but just take it from the corresponding entry of the correlation matrix. It is worth noting that in Step 6 the size, q, of the neighborhood is reduced iteratively in order to find a new error smaller than the previous one.

Finally, we have to make some changes to the correlation matrix in order to use it to select the most conflicting objectives properly. If we use this matrix, it is possible that some important conflicting objectives are discarded. For instance, if objective f_2 is in conflict with f_3 but not with f_1, then f_2 will be very close to f_1 and, thus f_2 can be removed even if it is one of the most conflicting objectives. To overcome this problem we carried out the following process to the correlation matrix:

- Find the maximum conflict value c_i^{max} of each row i in the matrix (i.e., the maximum negative correlation value for each objective).

- Add the value c_i^{max} to the column i. This means that we are assuming that if objective f_i is in conflict with some objectives, then it is in conflict with all the objectives.

Algorithm KOSSA mainly differs from algorithm MOSSA in the stopping criterion and in the input parameters. Figure 14 shows the modifications required to obtain algorithm KOSSA.

The main advantage of both algorithms is their low-computational complexity since unlike other algorithms (e.g., Brockhoff and Zitzler's algorithm presented in Sec. 4.3.2), the search for the best subset is not involved. Regarding the number of objectives, the proposed algorithms have complexity $O(k^2)$, where k is the number of objectives of the given nondominated set. The computation of the correlation for each pair of objectives has complexity $O(m)$, where m is the size of the nondominated set. Thus, the total complexity of both algorithms is $O(mk^2)$. For comparison purposes, Table 2 summarizes the complexities of our two objective reduction approaches, and those of the algorithms proposed by Brockhoff and Zitzler [12], and Deb and Saxena [31] (see Sec. 4.3.2). The time complexity of Deb and Saxena's algorithm corresponds to only one iteration, since the number of iterations depends on the threshold cut parameter. Usually, when this parameter is

Input: Set of nondominated solutions, A
Initial objective set $F = \{f_i, i = 1, ..., k\}$.
Number of neighbors $q \leqslant |F| - 1$.

Step 0: Compute the correlation matrix using A

Step 1: $F' \leftarrow F$.

Step 2: Find objective f_i^{min} which corresponds to
$r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.

Step 3: Retain f_i^{min} and discard its q neighbors.
Let $error \leftarrow r_{i,q}^{min}$.

Step 4: **If** $q > |F'| - 1$ **then** $q \leftarrow |F'| - 1$.

Step 5: **If** $q = 1$ **then** go to **Step 8** to stop.
Compute again $r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$

Step 6: **While** $r_{i,q}^{min} > error$ **do**:
$q \leftarrow q - 1$.
$r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.
If $q = 1$ **then** go to **Step 8**.

Step 7: Go to **Step 2**.

Step 8: Return set F' as the reduced objective set.

Figure 13: Pseudocode of the proposed objective reduction algorithm MOSSA.

near to one, more iterations are required to converge. Also, we are considering that the NSGA-II is used as the optimizer of the overall algorithm. The second term of the complexity corresponds to the NSGA-II as was published in [32], where g is the number of generations.

For practical purposes, Figure 15 shows the running times of Deb and Saxena's algorithm (DS algorithm), Brockhoff and Zitzler's algorithm for k-EMOSS (BZ algorithm), and KOSSA. The three algorithms were implemented in MATLAB[1]. In a first experiment we set the size of the input nondominated set to 240 solutions and, for the DS algorithm, we used 50 generations for each run of NSGA-II. As we can see in the upper-left plot of Figure 15, the running time of the DS algorithm is many times larger than those of the other two algorithms. For this reason, in a second experiment we only analyze in

1 Only in the running time comparison we used the MATLAB implementation of NSGA-II available at `http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=10429`. For the remainder experiments we used the C code available at `http://www.iitk.ac.in/kangal/codes.shtml`.

Input: Nondominated set A.
Initial objective set $F = \{f_i, i = 1, ..., k\}$.
Number of neighbors $q \leqslant |F| - k$.
Size of the desired objective subset, k.

Step 0: Compute the correlation matrix using A.

Step 1: $F' \leftarrow F$.

Step 2: Find objective f_i^{min} which corresponds to
$r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.

Step 3: Retain f_i^{min} and discard its q neighbors from F'.
Let $error \leftarrow r_{i,q}^{min}$.

Step 4: **If** $q > |F'| - k$ **then** $q \leftarrow |F'| - k$.

Step 5: **If** $|F'| = k$ **then** go to **Step 8** to stop.
Compute again $r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.

Step 6: **While** $r_{i,q}^{min} > error$ and $q > 1$ **do**:
$q \leftarrow q - 1$.
$r_{i,q}^{min} \leftarrow \min_{f_i \in F'} \{r_{i,q}\}$.

Step 7: Go to **Step 2**.

Step 8: Return set F' as the reduced objective set.

Figure 14: Pseudocode of the objective reduction algorithm KOSSA.

Algorithm	Complexity
Brockhoff & Zitzler (δ-MOSS)	$O(\min\{m^2k^3, m^4k^2\})$
Brockhoff & Zitzler (k-EMOSS)	$O(m^2k^3)$
Deb & Saxena†	$O(mk^2 + k^3) + O(gm^2k)$
MOSSA & KOSSA	$O(mk^2)$

†Complexity of each iteration of the algorithm.

Table 2: Complexity of the objective reduction algorithms considered in this study (m is the size of the nondominated set, k the number of objectives and g the number of generations for each run of NSGA-II).

detail the BZ algorithm and KOSSA, setting $m = 300$ (upper-right and bottom plots, respectively). Each of these plots shows the running time for the worst, "median" and best cases, depending on the size of the final objective subset size (k). Note that the worst and best cases for these algorithms occur with opposite values of k. Leaving trivial cases aside, for KOSSA, the worst case is presented when $k = 2$, while for the BZ algorithm the worst case occurs when $k = k - 1$. For both algorithms, the median case occurs when $k = k/2$ for even values of k.

We want to end this section with two important remarks. First, it should be noted that although KOSSA is intended to solve the k-EMOSS problem, MOSSA, as shown, does not solve the δ-MOSS problem, but a slightly different problem. Besides the difference between the semantic of the error involved, the δ-MOSS problem asks for the minimum subset with a given δ error, while MOSSA finds the minimum subset with the minimum error that the algorithm can achieve. Second, we have to note that the two algorithms proposed follow a top-down approach instead of a bottom-up approach like in the BZ algorithm[2] or in the DS algorithm. In other words, our algorithms start with the whole set of objectives and iteratively remove some objectives until the minimum non-redundant objective set is obtained. In contrast, the other algorithms start with an empty set to which some objectives are added at each iteration.

2 There exists a top-down version of the BZ algorithm. However, in this study we adopted the bottom-up version.

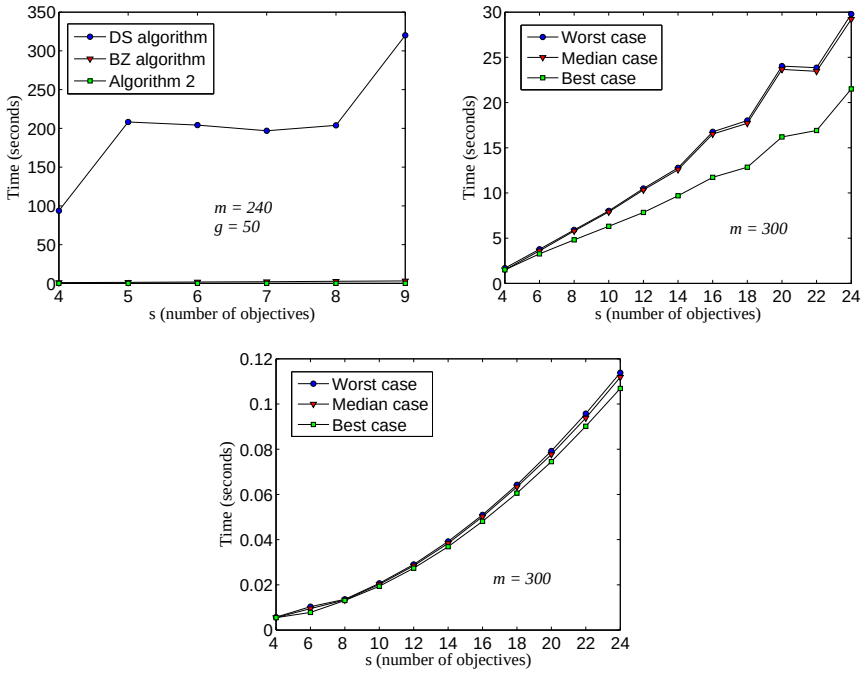


Figure 15: Running times of the three objective reduction algorithms. The upper-left plot shows the running times of the three algorithms using $k = k/2$, and $m = 240$, and, for the DS algorithm, 50 generations for NSGA-II. The other plots show the running times for the worst ($k = k-1, 2$), median ($k = k/2$) and best ($k = 2, k-1$) cases for the BZ algorithm and KOSSA, respectively.

5.2 Comparative Study

5.2.1 *Evaluation of MOSSA*

To evaluate the effectiveness of MOSSA, we compare its results against those obtained by the approach proposed by Deb and Saxena (DS algorithm) and the greedy algorithm proposed by Brockhoff and Zitzler to solve the δ-MOSS (with $\delta = 0$). In this experiment, we employed a variation of the well-known DTLZ5 problem defined in [31]. This variation, denoted by DTLZ5(I,M), allows to fix *a priori* the number of essential objectives, I, from the total number of objectives, M.

We apply the three algorithms to four instances of the DTLZ5(I,M) problem, namely: DTLZ5(2,3), DTLZ5(2,5), DTLZ5(2,10), DTLZ5(3,10). For the proposed approach and for Brockhoff and Zitzler's algorithm we use as input data a nondominated set of 500 solutions generated by the NSGA-II. We use an

Problem	Reduced set of objectives
DTLZ5(2,3)	f_1, f_3
DTLZ5(2,5)	f_1, f_5
DTLZ5(2,10)	f_1, f_{10}
DTLZ5(3,10)	f_1, f_9, f_{10}

Table 3: Essential objectives identified by the proposed MOSSA, Deb and Saxena's algorithm and Brockhoff and Zitzler's algorithm.

implementation of Deb and Saxena's algorithm following the instructions in [31] and for each run of NSGA-II we use a population size of 500, and 300 generations (i.e., 150 000 evaluations).

In all problems, the three algorithms were able to identify the essential objectives. Table 3 shows the essential objectives identified in each case.

5.2.2 *Evaluation of KOSSA*

The validation of KOSSA, intended to solve problem k-EMOSS (see Section 4.3.2), was carried out comparing its results with respect to the greedy algorithm proposed by Brockhoff and Zitzler (BZ algorithm) to solve the same problem. We also used Deb and Saxena's algorithm in this problem. However, since it does not address the k-EMOSS problem directly, we tried different values of the threshold cut (TC) to obtain different sizes of the final reduced objective set. In this comparison we considered two problems. First, we adopted two instances with 10 and 20 objectives of the 0/1 knapsack problem with 100 items. Second, we used two instances with 10 and 20 objectives of a variation, proposed in [13], of the well-known problem DTLZ2, which is denoted here by DTLZ2_{BZ}.

For the proposed approach and the BZ algorithm we used as input a nondominated set with 500 solutions generated by the NSGA-II, while for Deb and Saxena's approach we employed a population of 500 individuals and 400 generations (i.e., a total of 200 000 evaluations) for each run of the NSGA-II. We had to use this higher number of evaluations in order to obtain a good approximation of the optimal Pareto front and therefore achieve a good effectiveness of the algorithm.

In this evaluation, we adopted two different measures. First, we used the δ-error defined by Brockhoff and Zitzler [12]. This error measures the degree of change between the dominance relation induced by a subset of objectives F' and the whole set of objectives, F, with respect to a given solution set.

Thus, a value of 0 in this metric means that the subset F' contains only essential objectives since the dominance relation does not change.

The δ-error is ideal to assess the effectiveness of an objective reduction algorithm in a decision making problem. However to evaluate an algorithm of this kind with respect to the overall search process, we think that a well-known convergence quality indicator can be useful. Ultimately, when we remove some objectives from a problem, we would want that the obtained Pareto front by using the objective subset is as near as possible to the Pareto front of the original problem (i.e., that obtained considering all the objectives). Thus, we can evaluate the quality of two reduced subsets using some quality indicators usually adopted to measure the convergence of a generated Pareto front (PF_{approx}) to the optimal Pareto front (PF_{opt}). The detailed process to make such a comparison is the following. First, we have to obtain PF_{approx} using the two objective subsets. Then, we evaluate the two corresponding sets of Pareto optimal solutions using all the objectives of the original problem to obtain a new PF_{approx}. The closeness of these PF_{approx} sets to PF_{opt} can be used as a measure of quality of the reduced objective subsets. Besides the closeness to the optimal Pareto front we are interested in how well it is covered by the PF_{approx} sets obtained by each subset of objectives. These two objectives are taken into account by the inverted generational distance, so this is a natural candidate to evaluate the quality of two objective subsets.

The *inverted generational distance* (IGD), which is a variation of a quality indicator proposed in [116], is defined by: $IGD = \left(\sqrt{\sum_{i=1}^{n} d_i^2}\right)/n$, where $n = |PF_{opt}|$ and d_i is the Euclidian distance between each vector of PF_{opt} and the nearest member of PF_{approx}. In addition, this quality indicator measures the spread of PF_{approx} onto PF_{opt}. That is, a nondominated set whose vectors are located on a reduced area of PF_{opt} set, will be penalized in the value of this metric even though its vectors belong (or are near) to PF_{opt}. Lower values are preferred for this quality indicator.

5.2.2.1 *Evaluation Using the Change in the Dominance Relation*

The results corresponding to the 0/1 knapsack problem with a total of 10 objectives and using the δ-error measure, are shown in Table 4. The first column indicates the size of the obtained reduced objective set. Additionally, the best results of each row are shown in boldface.

The results show that the proposed algorithm obtained its best results when the number of objectives removed was small (from 1 to 5 objectives). In this problem, the BZ algorithm achieved the best result in 5 of the 8 cases, our approach obtained the best result in 3 cases, and the DS algorithm obtained the best result in only 1 case. Thus, with respect to the δ-error,

	δ-error		
# obj	**BZ**	**DS**	**KOSSA**
2	**1458**	1466	1595
3	**1412**	1513	1596
4	**1331**	1533	1466
5	1158	1362	**1148**
6	954	**859**	988
7	881	1148	**822**
8	**822**	1148	**822**
9	**569**	859	614

Table 4: Comparison with respect to the δ-error using the 0/1 knapsack problem with 10 objectives.

we can consider that the BZ algorithm had the best performance on this particular instance, while the DS algorithm had the worst performance.

In the instance with 20 objectives (see Table 5), KOSSA obtained the best result in 12 of the 18 cases, while BZ algorithm achieved the best result in 7 cases. On the other hand, the DS algorithm obtained the worst result in 14 cases. We conclude that, in this instance, KOSSA had the best performance and the DS algorithm the worst one.

Table 6 shows the results for the problem DTLZ2_{BZ} using a total of 10 objectives. In this instance, the proposed algorithm achieved the best results in 5 of the 8 cases, the BZ algorithm obtained the best results in 3 cases, and the DS algorithm obtained the best result only once. From these results, we can say that the proposed algorithm had the best performance on this problem. However, it is not clear which algorithm had the worst performance. On the one hand, the DS algorithm obtained the worst result in 2 cases and, on the other, although the BZ algorithm achieved the best result in 3 cases, it showed the worst result in 4 cases. Additionally, it is interesting to note that for a subset of 9 objectives, unlike the two reference algorithms, KOSSA was unable to find a subset with a δ-error equal to zero. That is to say, a subset that does not change at all the dominance relation.

With respect to the instance DTLZ2_{BZ} with a total of 20 objectives (see Table 7), the BZ algorithm obtained the best performance achieving the best result in 12 cases, while the proposed algorithm achieved the best result in 10 of the 18 cases. One more time, the DS algorithm was the worst algorithm since it obtained the worst results in 10 cases. In this problem we can observe that 4 objectives are completely redundant, since the dominance

	δ-error		
# obj	**BZ**	**DS**	**KOSSA**
2	**1467**	1599	1589
3	1434	1580	**1304**
4	1388	1433	**1304**
5	**1286**	1406	1304
6	**1243**	1405	1304
7	1164	1233	**1100**
8	1158	1225	**1003**
9	1149	1164	**993**
10	1075	1075	**918**
11	1056	1055	**804**
12	993	1153	**804**
13	937	918	**804**
14	871	1036	**804**
15	793	843	**785**
16	**785**	805	**785**
17	**699**	793	785
18	**359**	586	785
19	**41**	104	785

Table 5: Comparison with respect to the δ-error using the 0/1 knapsack problem with 20 objectives.

	δ-error		
# obj	**BZ**	**DS**	**KOSSA**
2	**0.6959**	0.7353	0.7353
3	**0.6688**	0.7051	0.7051
4	0.6610	0.6239	**0.5455**
5	0.6051	0.5890	**0.4908**
6	0.5916	0.5799	**0.4682**
7	0.5707	0.5799	**0.3787**
8	0.5151	0.5130	**0.3054**
9	**0.0000**	**0.0000**	0.3054

Table 6: Comparison of the three algorithms with respect to the δ-error, using the DTLZ2BZ problem with 10 objectives.

relation does not change absolutely from 16 to 20 objectives. Brockhoff and Zitzler's approach was the only algorithm that reflected this fact, while the DS algorithm and KOSSA found a subset with zero δ-error up to sizes of 19 and 18, respectively.

5.2.2.2 *Evaluation Using the Inverted Generational Distance*

In order to use the IGD metric to evaluate the quality of the objective subsets obtained by the reduction algorithms we need an optimizer to obtain the corresponding PF_{approx} sets using each objective subset. The objective subsets employed in this section are the same used in the previous section to compute the δ-error. Likewise, in the present study the PF_{approx} sets were obtained using the NSGA-II. For each objective subset the following parameters were used: 500 individuals and 1000 generations. The large number of generations was intended to produce Pareto fronts with a small standard deviation of IGD for each subset. For the two knapsack problems the standard deviation was at least 1×10^{-1}, whereas for the DTLZ2BZ problems it was at least 1×10^{-5}. Each IGD value shown in this section is the average of 20 runs for each objective subset. Moreover, instead of using the optimal Pareto front, we used the nondominated set resulting from the union of the PF_{approx} sets generated using the objective subsets of the three algorithms, as well as the PF_{approx} set used as input in the objective reduction algorithms in the previous section.

Table 8 shows the results for the 0/1 knapsack problem with 10 objectives with respect to the IGD metric. As we can see, in 4 of the 8 cases, the

	δ-error		
# obj	**BZ**	**DS**	**KOSSA**
2	**0.5931**	0.7442	0.7326
3	0.5925	0.7326	**0.5761**
4	**0.5570**	0.6275	0.5761
5	**0.4818**	0.5059	**0.4818**
6	0.4441	0.5171	**0.3706**
7	0.3257	0.4799	**0.2877**
8	0.3056	0.3257	**0.2877**
9	**0.2877**	0.4532	**0.2877**
10	**0.2461**	0.3036	0.2877
11	0.0569	0.3706	**0.0279**
12	0.0532	0.4125	**0.0216**
13	**0.0068**	0.4125	0.0216
14	**0.0038**	0.3056	0.0216
15	**0.0001**	0.1014	0.0216
16	**0.0000**	0.3056	0.0216
17	**0.0000**	0.0038	0.0068
18	**0.0000**	0.0038	**0.0000**
19	**0.0000**	**0.0000**	**0.0000**

Table 7: Comparison of the three algorithms with respect to the δ-error, using the DTLZ2BZ problem with 20 objectives.

	IGD		
# obj	**BZ**	**DS**	**KOSSA**
2	5.7834	7.5759	**5.7452**
3	**7.5212**	8.9904	7.5425
4	5.8694	**5.4537**	5.8445
5	3.7525	5.2474	**3.7406**
6	3.4633	3.6170	**3.4318**
7	3.2607	3.6615	**3.2473**
8	**3.1893**	3.7855	3.2001
9	**3.1521**	3.3678	3.1541

Table 8: Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the 0/1 knapsack problem with 10 objectives.

proposed algorithm achieved the best results, while the BZ and DS algorithms obtained the best result in 3 and 1 cases, respectively. With respect to the IGD metric the proposed approach had the best performance on this problem. It is worthwhile to remember that regarding the δ-error, in this instance the BZ algorithm obtained the best performance. This shows a clear inconsistency between the results based on the performance measure adopted. In particular, the DS algorithm obtained the worst result for the subset of 4 objectives using the δ-error, while it achieved the best result with respect to the IGD performance measure.

Regarding the 0/1 knapsack instance with 20 objectives (see Table 9), we can clearly see that the proposed algorithm had the best performance, which agrees with the conclusion obtained using the δ-error. However, the DS algorithm had the second best performance in this instance, while it was the worst algorithm with respect to the δ-error.

With respect to the DTLZ2$_{BZ}$ with 10 objectives, both KOSSA and the DS algorithm obtained the best IGD values in 5 of the 9 cases (see Table 10). Algorithm BZ obtained the best result in 3 cases, although it presented the worst result in 5 cases.

Nonetheless, we can observe a discrepancy between some results obtained by the two performance measures. On the one hand, using the δ-error, the DS algorithm showed the worst performance, but on the other hand, it is the best algorithm, along with KOSSA, regarding the IGD metric.

In the DTLZ2$_{BZ}$ instance using 20 objectives (see Table 11), the BZ algorithm showed the best performance. It obtained the best results in 13 cases. The proposed algorithm obtained the second best performance achieving the

	IGD		
# obj	**BZ**	**DS**	**KOSSA**
2	**7.7657**	8.0942	10.0344
3	6.7232	**6.4659**	6.9511
4	5.8777	**5.1232**	5.4373
5	4.8191	4.4417	**4.1027**
6	5.0627	4.8973	**4.7977**
7	5.1815	4.9218	**4.4888**
8	4.8017	5.0338	**4.7557**
9	4.8552	4.6459	**4.6217**
10	5.0189	5.0132	**4.9704**
11	4.7852	4.6779	**4.6470**
12	4.8673	4.8582	**4.7358**
13	4.9812	**4.7996**	4.8880
14	5.2370	5.0018	**4.9121**
15	5.1767	5.1215	**4.9753**
16	5.1166	5.0791	**4.9389**
17	**5.0209**	5.0534	5.0534
18	5.2423	**5.1553**	5.1639
19	5.1714	**5.1661**	5.2538

Table 9: Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the 0/1 knapsack problem with 20 objectives.

	IGD		
# obj	**BZ**	**DS**	**KOSSA**
2	0.00226	**0.00089**	**0.00089**
3	0.00243	**0.00071**	0.00080
4	**0.00082**	0.00089	**0.00082**
5	0.00115	0.00092	**0.00090**
6	0.00115	**0.00099**	0.00102
7	**0.00093**	0.00102	0.00114
8	0.00093	**0.00082**	**0.00082**
9	**0.00077**	**0.00077**	**0.00077**

Table 10: Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the DTLZ2BZ problem with a total of 10 objectives.

best results in 5 cases. Finally, the DS algorithm obtained the worst results in 13 cases. Therefore, it was the worst algorithm in this instance considering the IGD metric.

5.2.3 *Final Remarks*

Since KOSSA follows a top-down approach one would expect that it would achieve its best results when the size of the objective subset is close to the total number of objectives, specially when $k = k - 1$. However for this case, in 3 of the problems considered (both knapsack instances and DTLZ2BZ with 10 objectives), KOSSA was unable to obtain the least δ-error.

Although in general, IGD tends to decrease as the size of the objective subset increases, in some particular cases the IGD values increase when an objective is added. One possible reason for this behavior is that the number of objectives affects the optimizer degrading its performance as the number of objectives is increased. The results showed that the proposed algorithms are very competitive with respect to two other similar algorithms recently proposed [31, 12]. One advantage of the proposed algorithms over the two reference algorithms is their low computational time. MOSSA was able to identify all the essential objectives of the problems considered. With regard to the δ-error, KOSSA achieved the best performance in 2 of the 4 problem instances included (0/1 knapsack with 20 objectives and DTLZ2BZ with 10 objectives), whereas with respect to the IGD metric, it showed the best performance in 2 of the 4 instances (0/1 knapsack with 10 and 20 objectives)

	IGD		
# obj	**BZ**	**DS**	**KOSSA**
2	0.00235	**0.00115**	**0.00115**
3	0.00144	0.00253	**0.00078**
4	0.00104	0.00233	**0.00094**
5	**0.00094**	0.00110	0.00103
6	**0.00115**	0.00124	0.00148
7	0.00126	**0.00115**	0.00128
8	0.00115	**0.00104**	0.00110
9	**0.00103**	0.00123	0.00120
10	**0.00100**	0.00104	0.00102
11	**0.00097**	0.00104	0.00099
12	**0.00094**	0.00108	0.00097
13	**0.00094**	0.00104	0.00095
14	**0.00092**	0.00107	0.00097
15	**0.00091**	0.00095	0.00092
16	**0.00094**	0.00101	0.00098
17	**0.00095**	0.00096	**0.00095**
18	**0.00092**	0.00093	0.00093
19	**0.00092**	0.00093	**0.00092**

Table 11: Comparison of the three algorithms with respect to the inverted generational distance (IGD), using the DTLZ2BZ problem with a total of 20 objectives.

and, along with the DS algorithm, it obtained the best performance in the DTLZ2BZ problem with 10 objectives. Although, in general, the performance measures yielded consistent results, in some specific cases they yielded contradictory results. A possible explanation for this behavior is that a good performance of an objective reduction algorithm with respect to decision making does not imply a good performance with respect to the search process in general. However, further experimentation is required to either validate or refute this statement.

6 ONLINE OBJECTIVE REDUCTION

Contents

6.1 Gradual Reduction of the Objectives During the Search 57
6.2 Assessment of the Objective Reduction Schemes Coupled to a MOEA 60
6.3 The Conflict-Based Partitioning Framework 67
6.4 Experimental Results 74

In the previous chapter, two objective reduction methods were presented. Those methods can be directly used in the decision making process, i.e., once an approximation of the Pareto front has been obtained. In contrast, in this chapter we present and analyze two different approaches for incorporating objective reduction methods into a Pareto-based MOEA in order to cope with many-objective problems during the search, i.e., in an online fashion.

In the first approach the number of objectives is gradually reduced during the search until the required objective subset size has been reached. In each reduction phase, one of the objective reduction methods presented in the previous section is applied on the current Pareto front approximation. Towards the end of the search, the original objective set is used again to approximate the entire Pareto front. The second approach consists in partitioning the objective space into several subspaces so that a different portion of the population focuses the search on a different subspace. The partitioning of the set of objectives is based on the analysis of the conflict information obtained from the current Pareto front approximation.

6.1 Gradual Reduction of the Objectives During the Search

By selecting a computationally efficient objective reduction method we can expect that the resulting Multiobjective Evolutionary Algorithm (MOEA) improves its efficiency, since a smaller number of objective functions are evaluated. While this may be true, the omission of some objective implies some loss of information that could be important to converge to the real Pareto front. On the other hand, this omission can be useful to cope with the deterioration of the search ability of Pareto-based MOEAs in many-objective problems. With this in mind we propose two schemes to integrate an efficient

reduction method into a MOEA in such a way that the resulting MOEA can be useful even in problems with inexpensive objective functions. Additionally, one of the goals of this chapter is to investigate if an objective reduction method represents a benefit or a damage to the search ability of a MOEA.

Since López Jaimes *et al.*'s algorithms have a lower time complexity, they are suitable to be integrated into a MOEA because of the chances that their computational time savings overcome their overhead are larger than those of the other methods described here. However, in this study we have only chosen the algorithm that finds a subset of objectives of a given size.

When some objectives are discarded from the original problem some information is being lost. The magnitude of this loss depends on the degree of redundancy among the objectives.

In any case, we have to balance the benefit of discarding some objectives along with the computational cost of the reduction algorithm. Two benefits are clear from removing some objectives, namely: *i*) the avoidance of the computation of some (possibly) computational expensive objective functions, and *ii*) the execution time speedup of the MOEA, specially if its time complexity largely depends on the number of objectives.

Next, we describe two schemes to incorporate the KOSSA method into a MOEA. First, we propose a simple scheme where the objective set is successively reduced during most of the search. Only towards the end of the search all the objectives are integrated. This scheme is divided in three stages:

1. In the first stage the MOEA is executed for a number of generations using all the objectives. The MOEA obtains an initial approximation of the Pareto front which will be the first input of the objective reduction method, KOSSA.

2. The second stage is the main stage of the scheme where the objective set is gradually reduced through several generations. In this stage, every certain number of generations KOSSA is executed to reduce the objective set and then the execution of the MOEA is resumed. This process is repeated until the desired objective set size has been reached.

3. In the last stage all the objectives are taken up again to obtain the final approximation of the Pareto front.

The detailed scheme with successive reductions is described in Algorithm 2, where P denotes the best population obtained so far by the MOEA.

In the current implementation of this scheme we decided to schedule the reduction phases equally distributed during the reduction stage. However, other schedules are possible. For instance the number of generations for the next reduction can be shortened each time, since the population converges faster after each reduction. A similar decision can be made with regard to

Algorithm 2 Pseudocode of the successive reduction scheme.

Input:

R: Number of reductions during the search.
k: Size of the minimum objective set allowed.
G_{max}: Total number of generations.
G_{pre}: Generations before the reduction stage.
G_{post}: Generations after the reduction stage.

```
G ← G_pre;   F' ← F
k' ← ⌈(|F| − k)/R⌉   ▷ Number of objectives discarded per reduction.
for r ← 1 until R + 2 do
    for g ← 1 until G do
        MOEA(P, F')
    if r ≠ R + 2 then
        ▷ Reduce the current objective set F'.
        if r ⩽ R then
            F' ← KOSSA(P, F', |F'| − k')
            G ← (G_max − G_pre − G_post)/R
        else
            ▷ Integrate all the objectives at the end of the search.
            F' ← F
            G ← G_post
```

the number of objectives discarded on each reduction. Currently, the same number of objectives is removed at each reduction as it can be seen in the third statement of Algorithm 2.

Although this scheme has the advantage of omitting the evaluation of many objectives during most of the search, it is possible that the loss of information diminishes the MOEA's convergence ability. Therefore, we also proposed a less aggressive scheme which integrates the entire objective set periodically during the search to counterbalance the loss of information. As in the scheme previously described, this mixed scheme starts the search using the whole objective set for some generations. However, it alternates the reduction process with the integration of the original objectives during the remainder of the search. Algorithm 3 presents the details of the mixed scheme.

6.2 Assessment of the Objective Reduction Schemes Coupled to a MOEA

In order to evaluate the performance of the schemes presented in the previous section we chose the Nondominated Sorting Genetic Algorithm II (NSGA-II) as a testbed. As we mention in previous sections, the worth of using an objective reduction method depends on its computational cost, the time complexity of the MOEA (specially if it depends on the number of objectives), the computational cost of the objective functions, and on the effect caused by the removal of objectives.

In order to investigate the effect of these factors, we carried out two types of experiments. The first group of experiments attempts to provide an overall assessment of all those factors in order to determine if the reduction method is advantageous. To do so, instead of using the number of evaluations as a stopping criterion, we use the real computational time instead. By doing so, we can decide if the overall benefits of the reduction method are greater than its possible damages. In the second group of experiments we want to investigate if a reduction method increases or decreases the number of generations required to reach a certain quality of the approximation set produced.

In both types of experiments we compare the NSGA-II equipped with the reduction method (REDGA) against the original NSGA-II. The following problems were adopted in all the experiments: the 0/1 multi-objective knapsack problem with 200 items, and a variation, proposed in [13], of the well-known problem DTLZ2 (denoted here by DTLZ2_{BZ}) with 30 variables. All the runs were executed in a single-core computer with a 2.13 GHZ CPU.

Algorithm 3 Pseudocode of the mixed reduction scheme.

Input:
R: Number of reductions during the search.
k: Size of the minimum objective set allowed.
G_{max}: Total number of generations.
G_{pre}: Generations before the reduction stage.
p_{red}: Percentage of generations using the reduced objective set.

$p_{int} \leftarrow 1 - p_{red}$.
$G_{red} \leftarrow p_{red} \times (G_{max} - G_{pre})/R$
$G_{int} \leftarrow p_{int} \times (G_{max} - G_{pre})/R$
$G \leftarrow G_{pre}$
$F' \leftarrow F$
$k' \leftarrow \lceil (|F| - k)/R \rceil$ ▷ Number of objectives discarded per reduction.

for $r \leftarrow 1$ until $2R + 1$ **do**
  for $g \leftarrow 1$ until G **do**
    MOEA(P, F')

  if $r \neq 2R + 1$ **then**
    ▷ Reduce the current objective set F'.
    if $r \bmod 2 = 1$ **then**
      $F' \leftarrow$ KOSSA$(P, F', |F'| - k')$
      $G \leftarrow G_{red}$
    else
      ▷ Integrate all the objectives for the next generations.
      $F' \leftarrow F$
      $G \leftarrow G_{int}$

In the first group of experiments the results were evaluated using the additive ϵ-Indicator [130], which is defined as

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{\forall \mathbf{z}^2 \in B \ \exists \mathbf{z}^1 \in A : \mathbf{z}^1 \succeq_{\epsilon+} \mathbf{z}^2\}$$

for two nondominated sets A and B, where $\mathbf{z}^1 \succeq_{\epsilon+} \mathbf{z}^2$ iff $\forall i : z_i^1 \leqslant \epsilon + z_i^2$, for a given ϵ. In other words, $I_{\epsilon+}(A, B)$ is the minimum value such that aggregated to any objective vector in B, then $A \succeq B$. In general, $I_{\epsilon+}(A, B) \neq I_{\epsilon+}(B, A)$ so we have to compute both values. The smaller $I_{\epsilon+}(A, B)$ and the larger $I_{\epsilon+}(B, A)$, the better A over B.

6.2.1 *Overall Assessment of the Reduction Schemes*

In these experiments we used four instances for each of the two test problems employed with 4, 6, 8 and 10 objectives. For each number of objectives we fixed the following time windows: 2, 4, 6 and 10 seconds. For all the 30 runs and problems we used a population of 300 individuals. For NSGA-II we employed a crossover probability of 0.9 and a mutation probability of $1/N$ (N is the number of variables). In the knapsack problem we used a binary representation with a mutation probability of $1/n$ (n is the length of the chromosome).

In order to study the successive reduction scheme we reduced in all cases the objective set until a size of $k = 3$ and the percentage of generations before and after the reduction stage was fixed to 20% and 5%, respectively. Here, we studied two scenarios: one that reduces all the required objectives in one reduction (REDGA-S-1), while the other one uses, among all possible number of reductions, an intermediate number of reductions considering a final set of size $k = 3$ (REDGA-S-m). That is, for 6, 8 and 10 objectives were used 2, 3, and 4 reductions, respectively. In the mixed reduction scheme we only used an intermediate number of reductions for every number of objectives (REDGA-X-m), and the other parameters were $k = 3$, $p_{red} = 0.85$ and 20% of the total generations were accomplished before the reduction stage. The results of the ϵ-Indicator for these scenarios on problem DTLZ2_{BZ} are presented in Table 12. Since for four objectives REDGA-S-m and REDGA-X-m are equivalent to the REDGA-S-1 scheme, we only show the results of this scheme against NSGA-II.

As we can clearly see in Table 12, all the reduction schemes perform better than NSGA-II for every number of objectives. Besides, the advantage of the reduction schemes over the NSGA-II increases with the number of objectives. On the other hand, except for 8 objectives, the scheme REDGA-S-m achieved better results than the REDGA-X-m which is the second best in this comparison. This means that the strategy of integrating all the objectives

DTLZ2$_{BZ}$ with 4 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	-	-	0.04450	**0.04450**
REDGA-S-m	-	-	-	-	
REDGA-X-m	-	-	-	-	
NSGA-II	0.06469	-	-	-	0.06469
Average	**0.06469**			0.04450	

DTLZ2$_{BZ}$ with 6 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	0.05961	0.05723	0.06019	0.05901
REDGA-S-m	0.05259	-	0.05085	0.05849	**0.05398**
REDGA-X-m	0.05850	0.05614	-	0.05421	0.05628
NSGA-II	0.07447	0.07711	0.07972	-	0.07710
Average	0.06185	**0.06429**	0.06260	0.05763	

DTLZ2$_{BZ}$ with 8 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	0.08583	0.07711	0.07179	0.07824
REDGA-S-m	0.06905	-	0.08195	0.06341	**0.07147**
REDGA-X-m	0.07386	0.08171	-	0.06944	0.07500
NSGA-II	0.09882	0.10616	0.11782	-	0.10760
Average	0.08058	0.09123	**0.09229**	0.06821	

DTLZ2$_{BZ}$ with 10 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	0.09108	0.09182	0.08316	0.08869
REDGA-S-m	0.06916	-	0.07072	0.07926	**0.07305**
REDGA-X-m	0.07998	0.08554	-	0.06840	0.07797
NSGA-II	0.11608	0.12159	0.11480	-	0.11749
Average	0.08841	**0.09940**	0.09245	0.07694	

Table 12: Results of the reduction schemes with respect to the ϵ-Indicator in the DTLZ2$_{BZ}$ problem using a fixed-time stopping criterion.

periodically did not improve the performance of the reduction scheme. As somewhat expected, the REDGA-S-1 scheme did not obtain results as good as the other reduction schemes. A possible explanation is that, in spite of the fact that REDGA-S-1 carries out more evaluations than the other schemes in the given time, this advantage is not enough to counteract the negative effect caused by the loss of information. In this sense, the REDGA-S-m scenario represents a better tradeoff between these factors.

As in the previous problem, NSGA-II was the worst algorithm in the 0/1 knapsack problem regarding the ϵ-Indicator (see Table 13). Nonetheless, the REDGA-S-1 scheme presented a better performance than in DTLZ2$_{BZ}$, *i.e.*, with 4 objectives it was the second best and with 10 it was the best scheme. The reason is that knapsack's objective functions are more computationally expensive than those of the problem DTLZ2$_{BZ}$. This allowed that REDGA-S-1 could perform many more generations than any other scheme. This is a clear example that the balance between the computational cost of the objective functions and the overhead of the reduction scheme plays an important role on the success of the reduction scheme. Furthermore, it acts as a guide to decide what type of reduction scheme to choose. If the objective functions are expensive then it may be convenient to use an aggressive scheme such as REDGA-S-1; otherwise, the REDGA-S-m could be more appropriate.

6.2.2 *Effect of the Reduction Schemes on a MOEA's Search Ability*

In order to investigate how a reduction scheme affects the MOEA's convergence ability we compare the reduction schemes using the number of generations as the stopping criterion. In these experiments we used a population of 300 individuals for all the numbers of objectives considered, and all the algorithms were executed for 200 generations (60 000 evaluations). In this experiment we adopt only DTLZ2$_{BZ}$ since convergence can be easily measured given that the nondominated vectors of its true Pareto front have the property: $D = \sum_{i=1}^{s} f_i^2 = 1$, where s is the number of objectives. The distribution of the values of D for each algorithm are shown in Figure 16. The horizontal axis represents the D values obtained by each algorithm and the vertical axis denotes the frequency of a given D value. As well as in other studies [89, 117], Figure 16 shows that the performance of NSGA-II decays as the number of objectives increases. In addition, all the reduction schemes perform better than NSGA-II in all cases. This means that the reduction schemes, besides reducing execution time also help Pareto-based MOEAs to recover the search ability deteriorated by the inability of Pareto optimality to discriminate solutions in many-objective problems. In concordance with the fixed-time experiments, the REDGA-S-m achieves the best convergence with respect to the average D value presented in Table 14. Like all the algo-

Knapsack with 4 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	-	-	205	**205**
REDGA-S-m	-	-	-	-	
REDGA-X-m	-	-	-	-	
NSGA-II	241	-	-	-	241
Average	**241**			205	

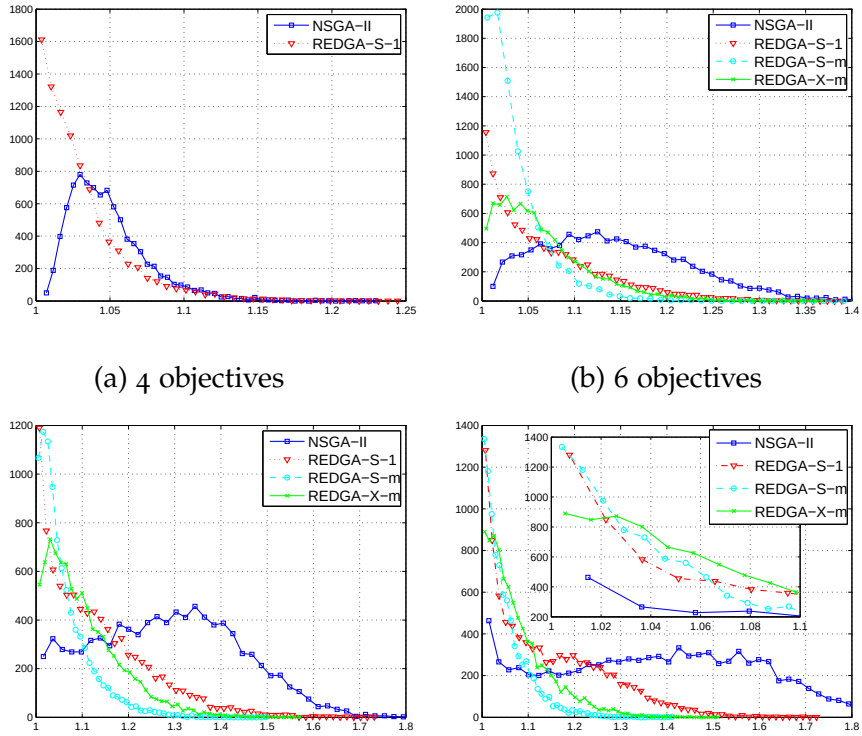
Knapsack with 6 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	408	264	318	**330.0**
REDGA-S-m	371	-	269	352	330.7
REDGA-X-m	372	403	-	306	360.3
NSGA-II	448	414	378	-	413.3
Average	397.0	**408.3**	303.7	325.3	

Knapsack with 8 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	646	478	505	543.0
REDGA-S-m	457	-	323	290	**356.7**
REDGA-X-m	441	465	-	345	417.0
NSGA-II	564	472	438	-	491.3
Average	487.3	**527.7**	413.0	380.0	

Knapsack with 10 objectives					
$I_{\epsilon+}(A, B)$	**REDGA-S-1**	**REDGA-S-m**	**REDGA-X-m**	**NSGA-II**	Average
REDGA-S-1	-	455	424	423	434.0
REDGA-S-m	503	-	411	376	**430.0**
REDGA-X-m	760	667	-	493	640.0
NSGA-II	533	455	522	-	503.3
Average	**598.7**	525.7	452.3	430.7	

Table 13: Results of the reduction schemes with respect to the ϵ-indicator in the 0/1 Knapsack problem using a fixed-time stopping criterion.

rithms, its convergence decreases with the number of objectives. However REDGA-S-m is the scheme less affected by the number of objectives.



(a) 4 objectives (b) 6 objectives

(c) 8 objectives (d) 10 objectives

Figure 16: D distribution on the problem DTLZ2BZ for different number of objectives. D = 1 corresponds to the true Pareto front.

6.2.3 *Final Remarks*

The first group of experiments based on a fixed-time stopping criterion showed that the reduction of objectives during the search is beneficial in spite of the loss of information since it also saves computational time. This means that the overhead introduced by the objective reduction method was small enough to speed up the execution of the MOEA even with the inexpensive objective functions used in the study. Although in all the cases studied in the first group of experiments the MOEA coupled with the reduction scheme achieved better results than the MOEA alone, we have to carefully select the parameters of the reduction scheme. There is an equilibrium point in the number of objectives that need to be removed in order to achieve

Obj		REDGA-S-1	REDGA-S-m	REDGA-X-m	NSGA-II
4	**Average**	**1.0305**	-	-	1.0488
	Std. Dev.	0.0289	-	-	**0.0266**
6	**Average**	1.0672	**1.0358**	1.0649	1.1445
	Std. Dev.	0.0609	**0.0334**	0.0496	0.0799
8	**Average**	1.1276	**1.0607**	1.1040	1.2863
	Std. Dev.	0.1113	**0.0561**	0.0805	0.1559
10	**Average**	1.1402	**1.0501**	1.0786	1.3787
	Std. Dev.	0.1234	**0.0487**	0.0690	0.2218

Table 14: Results of the reduction schemes with respect to the value D in the $\text{DTLZ2}_{\text{B}\,\text{Z}}$ problem using a fixed-generations stopping criterion.

the best tradeoff possible between the benefits and the damages obtained by the reduction scheme. To illustrate this, it is sufficient to consider that, although the REDGA-S scheme with only one reduction is the one that saves more time per generation, it did not present as good performance as a less aggressive configuration such as the REDGA-S-m. On the other hand, the periodic incorporation of the entire objective set did not improve the performance of the successive reduction scheme, which is simpler.

One important finding is that a reduction scheme, besides reducing the execution time of a MOEA, also helps to remedy the limitation of Pareto optimality for dealing with problems having a large number of objectives. The results showed that all the reduction schemes studied outperformed the original MOEA even when a stopping criterion based on a fixed number of generations was used. This shreds light into the usefulness of objective reduction schemes since they bring advantages both in efficiency and effectiveness.

6.3 The Conflict-Based Partitioning Framework

A general scheme for partitioning the objective space in several subspaces in order to deal with many-objective problems was introduced in [2]. In that approach the solution ranking and parent selection are independently performed in each subspace to emphasize the search within smaller regions of objective function space. Here, we propose a new partition strategy that creates objective subspaces based on the analysis of the conflict information obtained from the Pareto front approximation found by the underlying

MOEA. By grouping objectives in terms of the conflict among them, we aim to separate the Multiobjective Optimization Problem (MOP) into several subproblems in such a way that each subproblem contains the information to preserve as much as possible the structure of the original problem. Our approach is more closely related to the objective reduction approaches, specially those adopted during the search. However, its main difference with respect to them is the incorporation of all the objectives in order to cover the entire Pareto front. Deb and Saxena [31] proposed a method for reducing the number of objectives based on principal component analysis. Although some modifications can be made to this method in order to use it during the search, this method was designed as an *a posteriori* method. Brockhoff and Zitzler [14], and López Jaimes et al. [79] used similar objective reduction algorithms incorporated into a MOEA. However, in both cases, the non-conflicting objectives were discarded or aggregated to form a single objective.

6.3.1 *General Idea of the Partitioning Framework*

The basic idea of the partitioning framework is to divide the objective space into several subspaces so that a different portion of the population focuses the search on a different subspace. By partitioning the objective space into subspaces, we aim to emphasize the search within smaller regions of objective space. In other words, this framework divides the original optimization problem into several small subproblems. Instead of dividing the population into independent subpopulations, a fraction of the pool of parents for the next generation is selected based on a different subspace. This way, the pool of parents will be composed with individuals having a good performance in each subspace. Then, the crossover and mutation operators are applied as usual.

In the following, we first explain the general flow of the proposed method using NSGA-II as our framework. Then, we will explain in detail its distinctive features.

In our approach, we partition the M-dimensional space $\Phi = \{f_1, f_2, \ldots, f_M\}$ into N_S non-overlapping subspaces $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_S}\}$. Thus, the non-dominated sorting and truncation procedures of NSGA-II are modified in the following way. The union of the parent and offspring populations is sorted N_S times using a different subspace each time. Then, from each sorted population, the best $|P|/N_S$ solutions are selected to form a new parent population of size $|P|$. After this, the new population is generated by means of recombination and mutation using binary tournaments. Figure 17 shows a schematic view of the previously described procedure.

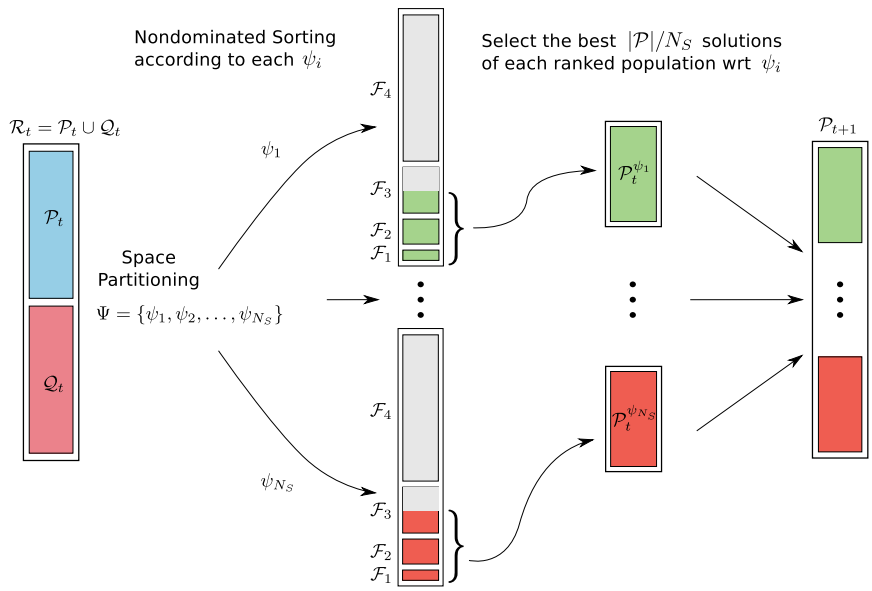


Figure 17: Non-dominated sorting and trucation based on different subspaces of a partition $\Psi = \{\psi_1, \psi_2, \ldots, \psi_{N_S}\}$.

The number of all possible ways to partition Φ into N_S subspaces is very large. Therefore, it is not feasible to search in all possible subspaces. Instead, a schedule of subspace sampling can be defined by using a partition strategy. In [2] we investigated three strategies to partition Φ: random, fixed, and shift partition strategy. The *random* strategy assigns at random the objectives to subspaces $\psi_s \in \Psi$. The *fixed* strategy deterministically assigns objectives $f_i \in \Phi$ to subspaces $\psi_s \in \Psi$ and keeps the same assignment throughout the generations. The *shift* strategy, at the first generation, assigns objectives in such a way that objectives assigned to a given ψ_s are ordered by their index i. Then, in subsequent generations, the objective with highest index in the s-th subspace is shifted to the $((s+1) \bmod N_S)$-th subspace, $\forall \psi_s \in \Psi$.

6.3.2 *A New Partition Strategy*

In this chapter, we investigate a new partition strategy using the conflict information among objectives. Namely, the first partition would contain the least conflicting objectives, the second one the next least conflicting objectives, and so forth. In previous studies [78, 79], the conflict information among objectives has been used to remove objectives after, and during the search. However, instead of removing the least conflicting objectives, we propose to integrate those objectives to form subspaces in such a way that all the objectives are optimized.

In the current literature it is possible to find several definitions of conflict among objectives (see e.g., [53, 1, 94, 12]). However, we used the definition proposed by Carlsson and Fullér [15, 16] since it is intuitive and, as we will explain later, it can be estimated using a low time complexity algorithm. Let be $S_{\mathcal{X}}$ a subset of $\mathcal{X}$, then, according to Carlsson and Fullér, two objectives can be related in the following ways (assuming minimization):

1. f_i is in conflict with f_j on $S_{\mathcal{X}}$ if $f_i(\mathbf{x}^1) \leqslant f_i(\mathbf{x}^2)$ implies $f_j(\mathbf{x}^1) \geqslant f_j(\mathbf{x}^2)$ for all $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.

2. f_i supports f_j on $S_{\mathcal{X}}$ if $f_i(\mathbf{x}^1) \geqslant f_i(\mathbf{x}^2)$ implies $f_j(\mathbf{x}^1) \geqslant f_j(\mathbf{x}^2)$ for all $\mathbf{x}^1, \mathbf{x}^2 \in S_{\mathcal{X}}$.

3. f_i and f_j are independent on $S_{\mathcal{X}}$, otherwise.

As stated in [12], the most conflicting objectives contain most of the information of the dominance structure of the problem. That is, if the non-conflicting objectives are removed from the problem, the Pareto front does not change. Therefore, by grouping objectives in terms of the conflict among them, we are trying to separate the MOP in subproblems in such a way that each subspace contains information to preserve most of the structure of the original problem. Nonetheless, there are few MOPs in which some objectives are totally independent from the others. Therefore, for a general case, we have to define degrees of conflict among the objectives.

Here, we suggest to use the correlation among the solution in PF_{approx} to estimate the conflict among objectives in the sense defined by Carlsson and Fullér. In this approach, each solution in PF_{approx} is an observation. A negative correlation between a pair of objectives means that one objective increases while the other decreases and vice versa. Thus, a negative correlation estimates the conflict between a pair of objectives. On the other hand, if the correlation is positive, then both objectives increase or decrease at the same time. That is, the objectives support each other. Furthermore, since the correlation coefficient values are in the range $[-1, 1]$, it is possible to define a measure of the degree of conflict between objectives. Thus, in our approach we interpret that the more negative the correlation between two objectives, the more the conflict between them. In [78] it was shown that using the correlation between objectives to identify the most conflicting objectives produces similar results than those obtained using a method described in [12] that explicitly measures the difference between the original and the reduced problem.

Next, we will introduce a new version of the basic partitioning framework. In order to implement the new partition strategy we should take into account that the conflict relation among the objectives changes during the search. This means that the conflict relation among the solutions in PF_{opt} might

differ from that observed in the current PF_{approx} found during the search. Thus, to deal with this situation we suggest a new partitioning framework in which the search is divided into several cycles. In turn, each of these cycles is divided into two phases, namely, an approximation phase followed by a partitioning phase. In the approximation phase all the objectives are used as usual to select the new parents population. The goal of this phase is to find a good approximation of the current PF_{opt}. At the beginning of the partitioning phase, the current PF_{opt} is used to compute the correlation matrix and creates a new partition of the objective space. In each cycle, the approximation phase is carried out during G_Φ generations, whereas the partitioning phase is carried out during G_Ψ generations using the partition created at the beginning of the cycle. This idea is graphically explained in Figure 18.

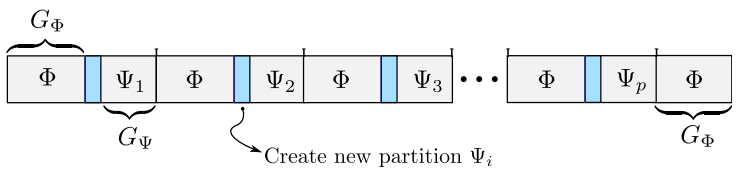


Figure 18: Alternation between the partition and the entire objective space.

The pseudocode of the entire proposed algorithm is described in Algorithm 4. In the current implementation all subspaces have the same dimension M/N_S in case $r = (M \bmod N_S)$ is zero. Otherwise, r of the N_S subspaces has dimension $M/N_S + 1$ and the rest M/N_S.

6.3.3 *Partitioning Using Conflict Information*

The correlation matrix is computed using the correlation coefficient (eq. 18) for each pair of objectives on the current parents population.

Definition 18 (Sample Correlation coefficient)**.** *The sample correlation coefficient,* r_{XY}*, is defined by* $r_{XY} = \sum_{i=1}^{m}(X_i - \bar{X})(Y_i - \bar{Y})/(m-1)s_X s_Y$*, where* $s_X > 0$ *and* $s_Y > 0$ *denote the sample standard deviations for the data sets* X *and* Y*, respectively, and* m *is the number of elements of each data set.*

Since we are interested in measuring the negative correlation between objectives, the correlation matrix was modified so that each entry, r_{f_i,f_j}, contains the value $1 - r_{f_i,f_j}$. Thus each value of this new "conflict matrix" is in the range $[0, 2]$. A result of zero indicates that objectives f_i and f_j completely support each other, and a value of 2 indicates that they are completely in conflict.

Algorithm 4 Pseudocode of our proposed partitioning MOEA.

Input: Evolutionary operators values, N_S (Num. of subspaces)
Output: Pareto front approximation

```
P_1 ← randomPopulation()
evaluate(P_1)
crowding(P_1)
Ψ ← {{f_1, ..., f_M}}    ▷ All the objectives in a single subspace.
phase ← INTEGRATION
G_change ← G_Φ
g ← 0
for t ← 1 until G_max do
    Q_t ← newPop(P_t)    ▷ selection, crossover and mutation.
    evaluate(Q_t)
    R_t ← P_t ∪ Q_t
    P_{t+1} ← sort&Truncation(R_t, |P_t|, Ψ)
    if g = G_change then
        g ← 0
        if phase = INTEGRATION then
            phase ← PARTITIONING
            Ψ ← conflictPartition(P_{t+1}, Φ, N_S)
            G_change ← G_Ψ
        else
            phase ← INTEGRATION
            Ψ ← {{f_1, ..., f_M}}
            G_change ← G_Φ
    g ← g + 1
```

Algorithm 5 Procedure of non-dominated sorting and truncation.

```
procedure sort&Truncation(R, |P|, Ψ)
    P* ← ∅
    for i ← 1 until |Ψ| do
        F^{ψ_i} ← nonDominatedSort(R, ψ_i)
        crowding(F^{ψ_i}, ψ_i)
        P^{ψ_i} ← truncation(F^{ψ_i}, |P|/|Ψ|)    ▷ |P^{ψ_i}| = |P|/|Ψ|
        P* ← P* ∪ P^{ψ_i}
    return P*    ▷ |P*| = |P|
```

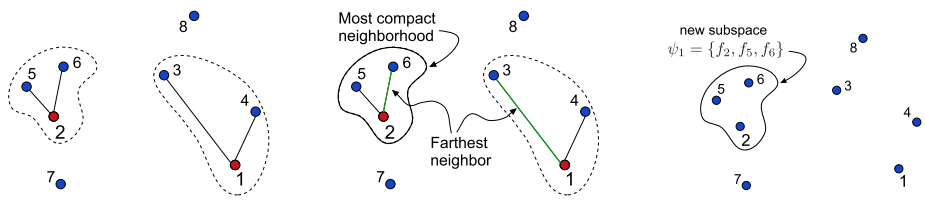
Then, the subspaces are created from the least conflicting subspace to the most conflicting subspace. The procedure to create subspaces of size k is the following:

1. Create q-sized neighborhoods around each objective f_i, where $q = k - 1$. The conflict between objectives takes the role of the distance. That is, the more the conflict between two objectives, the more distant they are in the "conflict" space. Figure 19(a) shows two of these neighborhoods of a hypothetical situation to form subspaces of size $k = 3$.

2. Select the most compact neighborhood, i.e., the neighborhood with the smallest distance to its q-th neighbor (farthest neighbor). Figure 19(b) shows the farthest neighbor for each of the two neighborhoods. In the example, the neighborhood on the left is the most compact one.

3. Finally, the objectives in the most compact neighborhood, including objective f_i, form a new subspace, and these objectives are removed from the conflict matrix.

This process is repeated until objectives are assigned to one subspace. Therefore, the first subspace created contains the least conflicting objectives, and the last subspace is formed by the most conflicting objectives. Algorithm 6 shows the pseudocode of this process.



(a) Divide the objective set into neighborhoods around each objective.

(b) Select the most compact neighborhood.

(c) The center and its neighbors form a new subspace.

Figure 19: Basic strategy to create subspaces using conflict information.

Algorithm 6 Partitioning Using Conflict Information.

```
procedure CONFLICTPARTITION(𝒫, Φ, N_S)
    cMatrix ← COMPUTECONFLICTMATRIX(𝒫)
    ADJUSTMATRIX(cMatrix)
    k ← (|Φ|/N_S) − 1
    Φ′ ← Φ = {f_1, ..., f_M}    ▷ Remaining objectives.
    for s ← 1 until N_S − 1 do
        for each objective f_i in Φ′ do
            V_{f_i} ← Ascending ordered list of the k-nearest neighbors
                      of f_i wrt the conflict using cMatrix.
        ψ_s ← V_{f_i} ∪ {f_i} : V_{f_i}[k] ⩽ V_{f_j}[k], ∀f_j ∈ Φ′
        Ψ ← Ψ ∪ ψ_s
        Φ′ ← Φ′ − ψ_s
    Ψ ← Ψ ∪ Φ′
    return Ψ
```

Finally, we have to explain some necessary modifications to the correlation matrix in order to use it to select the most conflicting objectives properly. If we used the original correlation matrix, it would be possible that some highly conflicting objectives might be placed in a subspace with a low conflict. For instance, if objective f_2 is in conflict with f_3 but not with f_1, then f_2 would be very close to f_1 and, thus f_2 would be placed in a low conflicting subspace even if it is one of the most conflicting objectives. To overcome this problem we carried out the following process to the correlation matrix:

- Find the maximum conflict value $c_{i,max}$ of each row i in the matrix (i.e., the maximum negative correlation value for each objective).
- Add the value $c_{i,max}$ to the column i. This means that we are assuming that if objective f_i is in conflict with some objectives, then it is in conflict with all the objectives.

6.4 Experimental Results

6.4.1 *Algorithms and Parameter Settings Employed*

As mentioned before, we used NSGA-II's framework to study the conflict-based partitioning strategy. Additionally, we wanted to investigate the advantages or disadvantages of the conflict-based strategy with respect to the random strategy. Therefore, we compare the original NSGA-II, and the NSGA-II using the conflict and random partition strategies.

In all the experiments we used the following parameters: a crossover probability of 0.9, a mutation probability of $1/n$ (n is the number of variables), setting the distribution indices for crossover and mutation to 15 and 20, respectively, in the case of continuous problems.

In all the algorithms we used a population of 200 individuals, and a total number of generations of 200. For all the configurations we carried out 30 runs for each MOEA. The results presented were averaged over the total number of executions. In our experiments we used from 4 to 15 objectives in each test problem. We used 2 subspaces for 4-9 objectives, and 3 subspaces for 10-15 objectives. In addition, for one of the test problems we also used 24 objectives to analyze the effect of the size of the subspaces.

6.4.2 *Test Problems Employed*

In order to show how the conflict-based strategy works, we used test problems in which the conflicting objectives can be defined a priori by the user. Namely, the problem $\text{DTLZ}_5(I, M)$ which is a variant proposed by Saxena and

Deb [31] based on the original DTLZ5. In this problem, from a total of M objectives, only I objectives are in conflict, whereas the rest are non-conflicting objectives that do not provide information to determine the Pareto optimal front. The conflicting objectives are the last $I-1$ objectives and any of the other objectives.

Additionally, we employed two test problems in which the conflicting relation among the objectives is not known a priori. One of these problems is the problem $DTLZ2_{BZ}$ proposed by Brockhoff and Zitzler [14] based on the original DTLZ2. When any of the objectives is removed from the original DTLZ2, the resulting Pareto front is reduced to a single non-dominated solution. The $DTLZ2_{BZ}$ variant avoids this problem, but it preserves the property that $\sum_{i=1}^{M}(z_i)^2 = 1$, for all $\mathbf{z} \in PF_{opt}$. The second problem is the 0/1 Knapsack with 300 items.

In both problems ($DTLZ5(I, M)$ and $DTLZ2_{BZ}$) we employed a similar configuration in order to maintain the test problem's complexity for every number of objectives. Specifically, we fixed the number of distance-related variables[1] to 20. The number of position-related variables[2] was set to $M-1$ in $DTLZ5(I, M)$ and $DTLZ2_{BZ}$.

6.4.3 *Quality Indicators Employed*

Since in many-objective problems it is not possible to use plots of the Pareto front approximations obtained to help in the interpretation of results, we have to rely on the results obtained by the quality indicators. For this reason, we used several indicators, and in some cases, we relied on parallel coordinates plots to interpret the results.

In order to evaluate the convergence achieved by the MOEAs we used generational distance (GD). In the case of $DTLZ2_{BZ}$ and $DTZL5(I, M)$ we took advantage of the geometry of their Pareto front to compute the exact generational distance. $DTLZ2_{BZ}$ and $DTLZ5(I, M)$ have the property: $\sum_{i=1}^{M}(z_i)^2 = 1$ for all $\mathbf{z} \in PF_{opt}$ This way, the generational distance was computed using $GD = (\sum_{i=1}^{M}(z_i)^2/|PF_{approx}|) - 1$. In the case of the Knapsack problem, the usual definition of generational distance was adopted, using as our reference Pareto front, the resulting non-dominated individuals of the union of PF_{approx} obtained by the three algorithms in all the runs for a given test problem.

1 Distance-related variables are related to the progress towards the Pareto optimal front.
2 Position-related variables generate solutions in the same local Pareto front.

Additionally, to directly compare the convergence of the MOEAs in all the test problems, we utilized the additive ϵ-indicator [130]. This indicator is defined as

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{\forall \mathbf{z}^2 \in B \ \exists \mathbf{z}^1 \in A : \mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2\}$$

for two nondominated sets A and B, where $\mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2$ iff $\forall i : z_i^1 \leqslant \epsilon + z_i^2$, for a given ϵ. In other words, $I_{\epsilon+}(A, B)$ is the minimum value such that aggregated to any objective vector in B, then $A \preceq B$. In general, $I_{\epsilon+}(A, B) \neq I_{\epsilon+}(B, A)$ so we have to compute both values. The smaller $I_{\epsilon+}(A, B)$ and larger $I_{\epsilon+}(B, A)$, the better A over B.

In order to evaluate diversity, we used the inverted generational distance (IGD). Similarly to GD, for this indicator we used the non-dominated solutions of all the PF_{approx} generated for a given test problem as our reference Pareto front.

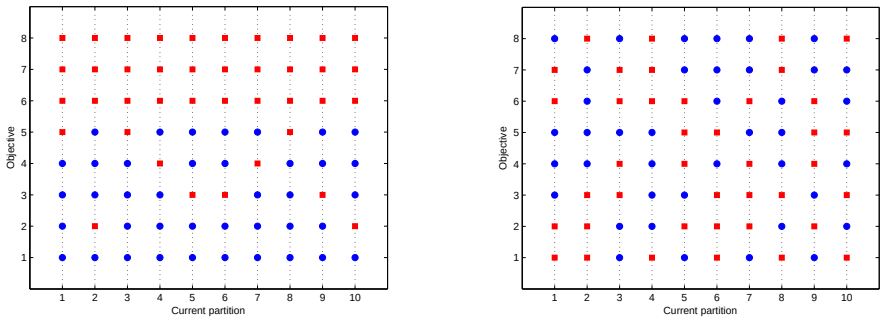
Finally, to assess both convergence and diversity, we adopted the hypervolume indicator. For $DTLZ2_{BZ}$, $DTLZ5(I, M)$ the reference point was $\mathbf{z}^{ref} = 1.5^M$. The results presented correspond to the normalized hypervolume using the enclosed hypervolume between the ideal point $\mathbf{z}^* = 0^M$ and the reference point. For the knapsack problem, the reference point was formed using the worst value in each objective of all the PF_{approx} generated for all the algorithms. In this case, the hypervolume was normalized using the hypervolume yielded by NSGA-II. Due to the high computational complexity of the hypervolume with respect to the number of objectives, we only computed this indicator for 4-10 objectives.

6.4.4 *Problems With Conflict Known A Priori*

In this section, we present the experimental results using the problem $DTLZ2(I, M)$. In these experiments, we used $I = 4$ conflicting objectives from a total of $M = 4, \ldots, 15$ objectives. For 4-9 objectives, 2 subspaces were used, whereas for 10-15 objectives, we employed 3 subspaces.

First, we want to show that the conflict-based strategy was able to correctly identify the conflicting objectives in most of the partitions generated during the search process. Figure 20 shows the subspaces generated by the conflict-based and random partition strategies during the search process. In this example, there is a total of $M = 8$ objectives. The conflicting objectives are objectives 6-8 and any of the other objectives. The objectives in the most conflicting subspace are denoted by squares, and the other subspace is denoted by circles.

In Figure 20(a) we can see that in the first partitions generated, some of the objectives were assigned to the wrong subspace. The reason of this



(a) Conflict-based partitions.

(b) Random partitions.

Figure 20: Subspaces generated using the conflict and random partition strategies on problem DTLZ5$(I = 4, M = 8)$. Objectives 6-8 and any of the other objectives are the conflicting objectives.

behavior is that in the first generations the current population does not yet represent an accurate sample of the real shape of the Pareto front. However, as the search progresses, the input PF_{approx} used to estimate the conflict approaches the true Pareto front. Therefore, in the last stages of the search, the conflict-based strategy was able to create the correct partition. On the other hand, by using the random strategy, the chances that the correct partition is created are very low. In the example shown in Figure 20(b), only the fourth partition generated contains the correct subspaces. Consequently, in most of the generations of the search, the selected parents emphasize objective subspaces that do not maximize the contribution to form the true Pareto front.

Figure 21 shows the results for the generational distance. The most evident fact in that plot is that the convergence of NSGA-II degrades dramatically when the number of objectives is more than 6. In fact, the convergence in terms of GD tends to diverge. A possible reason of this behavior is the generation of Dominance Resistant Solutionss (DRSs) in DTLZ5(I, M). These solutions are far from the true Pareto front, however, since they are nondominated solutions, they are candidates to form the new parents population. Since DRSs are boundary solutions, most of them will have the best crowding value. Therefore, these solutions will always be included in the new parents population. As mentioned in the introduction, the proportion of non-dominated solutions in a population increases exponentially with respect to the number of objectives. As a result, this problem gets worse when the number of objectives grows. In contrast, it seems that the GD values using any of the partition strategies, are not affected by the number of

objectives. In particular, we can see that the convergence obtained by using the conflict-based partition strategy is better than the one achieved by the random strategy.

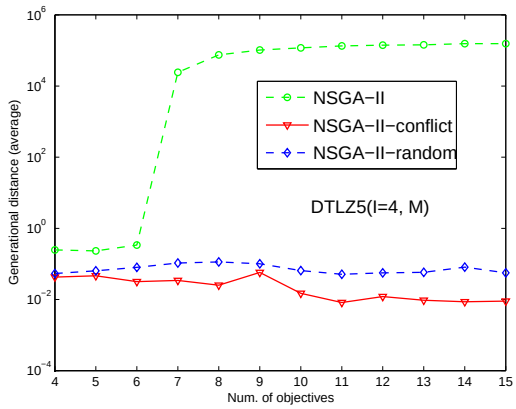


Figure 21: Generational distance results in problem $DTLZ5(I = 4, M)$. From 4-9 objectives it was generated a partition with 2 subspaces, while for 10-15 objectives it was generated one with 3 subspaces.

Nonetheless, this difference is only marginal. By inspecting the parallel coordinate plot presented in Figure 22 we realized that NSGA-II with the random strategy converges to the extremes of the Pareto front. That is, most of the solutions are close to 0 or 1 in one objective, but very few solutions are generated in between. In contrast, the conflict-based strategy covers all the trade-offs between the objectives.

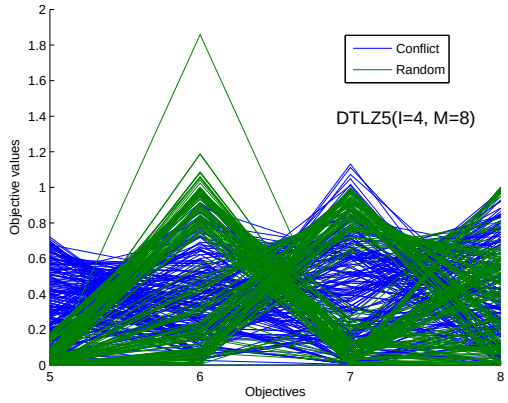


Figure 22: $DTLZ5(I = 4, M = 8)$: Parallel coordinate plot of the Pareto front approximations obtained with the random and the conflict partition strategies.

In order to measure this situation, we compute the inverted generational distance. Figure 23 shows that the conflict-based partition strategy achieves better values in terms of the inverted generational distance. This indicates a better distribution using the conflict-based partition strategy.

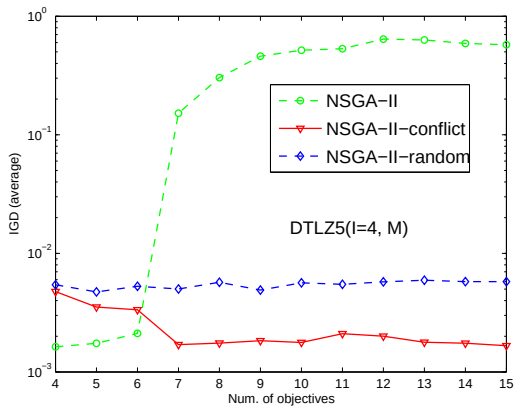


Figure 23: Inverted Generational distance results in problem DTLZ5($I = 4, M$). From 4-9 objectives a partition with 2 subspaces was generated, while for 10-15 objectives it was generated one with 3 subspaces was generated.

Next, we present the convergence assessment using the ϵ-indicator. To some degree, this indicator also takes into account the distribution of the Pareto front approximations compared. For example, let A be a non-dominated set which is well-distributed along the entire Pareto front, and B a subset of A concentrated on a small region of the Pareto front. A already dominates B, however some positive ϵ value must be added to A in such a way that B dominates every solution in A.

The results of the ϵ-indicator are presented in the matrices of subplots of Figure 24. We can interpret these results as follows. $I_{\epsilon+}(A, B)$ is the subplot located in row A, and column B of the matrix. The boxes in each subplot depict the results for each number of objectives considered. As we can see, the results of the ϵ-indicator indicate that NSGA-II is clearly outperformed by the NSGA-II using any of the partition strategies. With respect to the comparison of both partition strategies we can observe that the average ϵ values using the conflict-based strategy are better than those achieved by the random strategy, specially for 6 or more objectives. That is, $I_{\epsilon+}(\text{Conflict}, \text{Random}) < I_{\epsilon+}(\text{Random}, \text{Conflict})$ for any number of objectives.

Finally, we present the results of the hypervolume indicator. Since the hypervolume considers both convergence and distribution to assess two non-dominated sets, as we can see in Figure 25, the conflict-based partition strategy outperforms the random strategy. Additionally, it can be seen

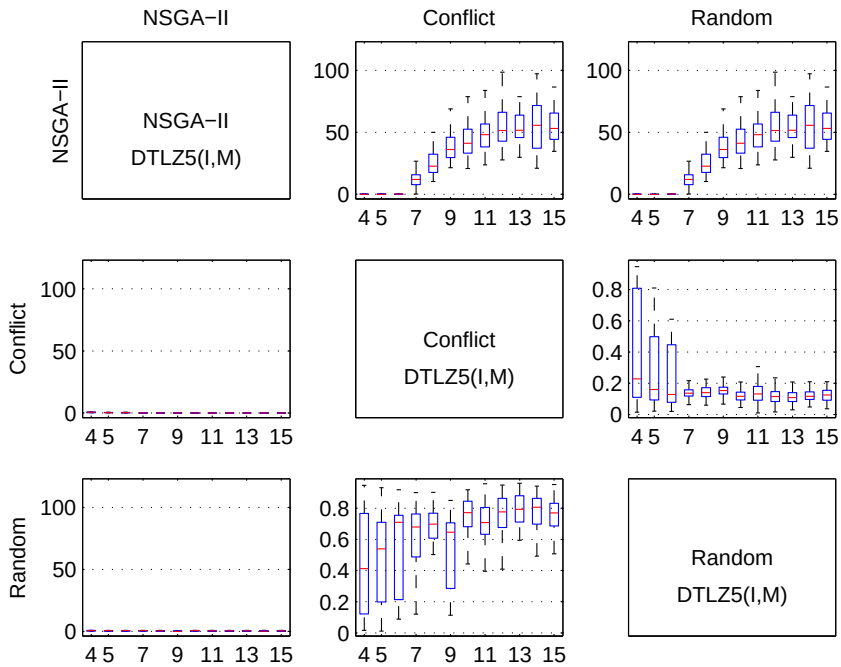


Figure 24: DTLZ5$(I = 4, M)$: ϵ-indicator results. The vertical axis of each subplot denotes the corresponding ϵ value, and the horizontal axis the boxplot for each number of objectives considered.

that the hypervolume obtained by the random strategy tends to decrease for 10 objectives. This can be explained by the fact that the chances that the conflicting objectives are assigned to the correct subspaces when they are randomly selected, decreases when more objectives are added to the problem.

Like in the previously analyzed indicators, the original NSGA-II achieved a poor performance in terms of the hypervolume indicator. However, it is worth noting a recurrent behavior using the hypervolume and other indicators. That is, for less than 5 or 6 objectives, NSGA-II presents a better or similar performance than that achieved by using a partition strategy. There are two facts that explain this behavior. Firstly, that the NSGA-II is still able to deal with that lower number of objectives. Second, since there are 4 conflicting objectives for 4-6 objectives, using 2 subspaces it is not possible that all the conflicting objectives are grouped into one subspace. Therefore, the trade-offs between objectives in different subspaces are not well represented. This suggest that it is convenient to assign all the highly correlated objectives to a single subspace. However, a large subspace might

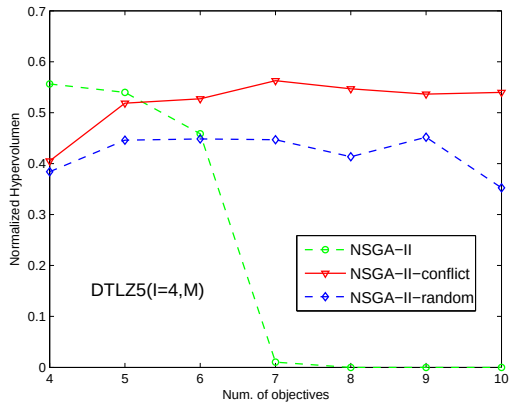


Figure 25: Normalized Hypervolume results for DTLZ5$(I = 4, M = 8)$.

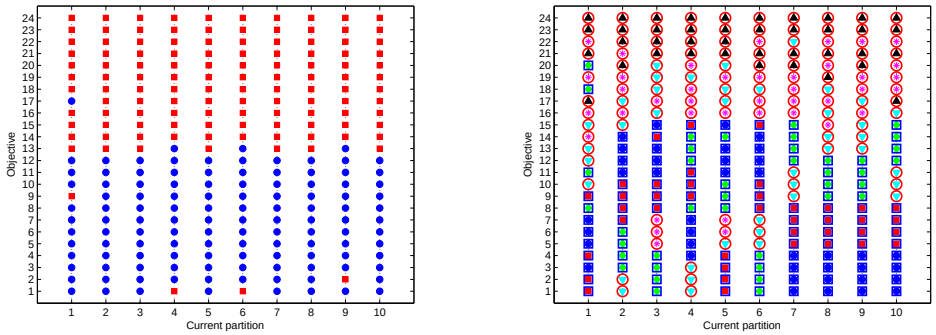
surpass the capacities of the underlying MOEA. In the next section we will analyze the effect of the size of the subspaces in the partition.

6.4.5 *Effect of the Size of the Subspaces*

In this section we analyze if it is better to have all the conflicting objectives together although in a large subspace, or small subspaces although the conflicting objectives are in different subspaces. To this end, we used DTLZ5(I, M) $M = 24$ objectives and $I = 12$ objectives in conflict. Then, we compare two different partitions, namely, one with two subspaces with 12 objectives each, and another one with 6 subspaces with 4 objectives each.

First, we want to show that both types of partitions are able to identify the conflicting objectives. However, in most cases, the partition with two subspaces achieved a better identification of the conflicting objectives. Figure 26(a) and Figure 26(b) show an example of the partitions created using 2 and 6 subspaces, respectively. In order to easily verify if the objectives in the partition with 6 subspaces were correctly identified, the objectives in subspaces 1-3 are marked with a square, and those in subspaces 4-6 (conflicting objectives) with circles.

Figure 27 shows the progress of the generational distance indicator during all the search process. Similarly to previous experiments, due to the dominance resistant solutions, NSGA-II diverges with respect to GD. However, what we want to emphasize is the fact that each partition strategy achieved a better convergence using 6 subspaces with 4 objectives. This suggests that it is preferable to have subspaces of moderate size, even if highly conflicting objectives have to be assigned to different subspaces. The optimal size of the subspaces depends on the capacities of the underlying MOEA. For example,



(a) 2 subspaces.

(b) 6 subspaces.

Figure 26: DTLZ5($I = 12, M = 24$): Subspaces generated using 2 subspaces with 12 objectives, and 6 subspaces with 4 objectives.

based on the experimental results observed so far, an appropriate size of the subspaces for NSGA-II would be between 4 and 6 objectives. However, for other MOEAs, like SPEA2 or PAES, the optimal subspace size might be different.

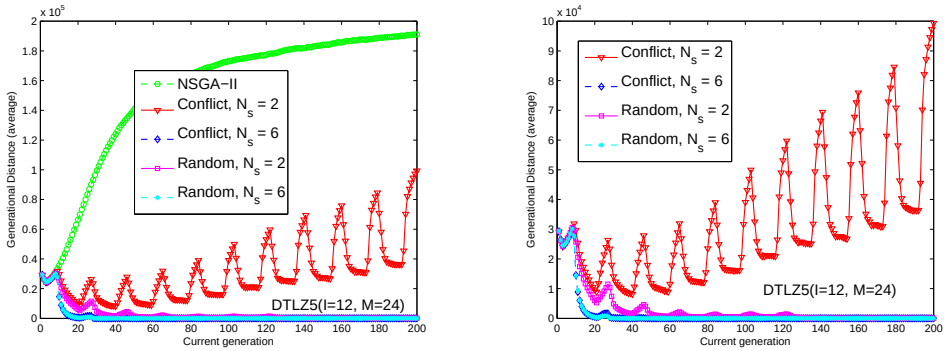


Figure 27: DTLZ5($I = 12, M = 24$): Online Generational Distance using a partition with 2 subspaces and another one with 6 subspaces.

Regarding the conflict and random strategies, it can be seen that the conflict-based strategy also diverges when using 2 subspaces. However, the random-based strategy with the same number of subspaces obtains good results. In order to find the reason of this behavior, we will analyze the performance using other quality indicators and plots. The parallel coordinate plot shown in Figure 28 indicates that by using a random strategy the solutions converge to the extremes of only a pair of objectives. In contrast,

the conflict-based strategy finds solutions that optimize more objectives and cover the mid trade-off regions of the Pareto front.

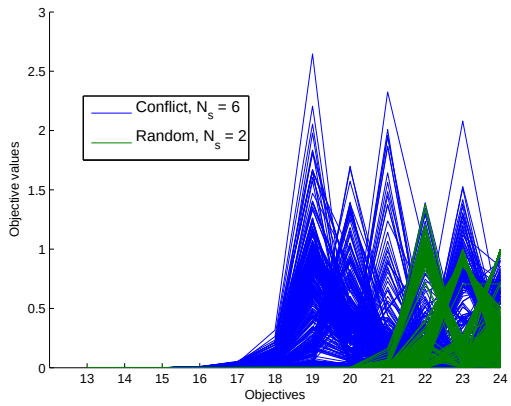


Figure 28: DTLZ5($I = 12, M = 24$): Parallel coordinate plot of the Pareto front approximations obtained by the conflict-based strategy with 6 subspaces, and the random strategy with 2 subspaces.

In order to quantitatively assess the distribution, we compare the algorithms using the inverted generational distance, whose results are shown in Table 15. Although the obtained generational distance of the conflict and random strategies are similar using 6 objectives (see Figure 27), the results of the inverted generational distance shown in Table 15 suggest that the conflict strategy with 6 subspaces achieved a better distribution of the solutions than the random strategy with 6 subspaces. In fact, the random strategy with 2 subspaces achieved a better IGD than the one yield using 6 objectives.

	NSGA-II	**Conflict**		**Random**	
		$N_S = 2$	$N_S = 6$	$N_S = 2$	$N_S = 6$
Average	0.18005	0.00838	**0.00570**	0.00682	0.00769
Std. Dev.	0.04695	0.00010	0.00047	0.00092	0.00029
Worst	0.26993	0.00849	**0.00758**	0.00791	0.00819
Best	0.07868	0.00788	**0.00454**	0.00591	0.00716

Table 15: IGD values for DTLZ5($I = 12, M = 24$) using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions.

Based on the ϵ-indicator results shown in Table 16, we can confirm that both partition strategies have a better performance using partitions with 6 subspaces. In the same way, the conflict strategy outperformed the other

algorithms in terms of the ϵ-indicator. The negative results in the column of NSGA-II indicate that, on average, the Pareto front approximations yielded by the conflict strategy with 6 subspaces and by both random strategies, dominate the Pareto front approximations obtained by NSGA-II.

$I_{\epsilon+}(A, B)$	**NSGA-II**	**Cft** $N_S = 2$	**Cft** $N_S = 6$	**Rnd** $N_S = 2$	**Rnd** $N_S = 6$
NSGA-II	x	14.4030	14.4030	14.4030	14.4030
Cft, $N_S = 2$	0.1977	x	0.9644	0.9644	0.9623
Cft, $N_S = 6$	-6.73e-6	0.0372	x	0.1974	0.1012
Rnd, $N_S = 2$	-6.73e-6	0.0107	0.4845	x	0.1407
Rnd, $N_S = 6$	-6.73e-6	0.6009	0.6010	0.6011	x

Table 16: $I_{\epsilon+}$ values for DTLZ5$(I = 12, M = 24)$ using 2 and 6 subspaces in each of the partitioning strategies, namely, random- and conflict-based partitions.

6.4.6 *Problems With Unknown Conflict*

In this section, we analyze the performance of the conflict and random partition strategies in problems in which the conflict relation among objectives is not known a priori. That is, the DTLZ2$_{BZ}$ and Knapsack problems.

Based on the symmetrical geometry of DTLZ2$_{BZ}$'s Pareto front (which is a sphere), it seems that the conflict between every pair of objectives is very similar. Therefore, we would expect that both partition strategies present a similar performance. Figure 29 shows the results for the generational distance and the inverted generational distance obtained in problem DTLZ2$_{BZ}$. As we expected, the experimental results show that both partition strategies obtained a similar performance in both indicators. However, the conflict-based strategy achieved a slightly better performance.

In a similar way, both algorithms achieved similar results with respect to the hypervolume and the ϵ-indicator (see Figure 30 and 31).

Although in DTLZ2$_{BZ}$, the conflict information was not useful to create the partitions, as we will see, in the Knapsack problem there is an interesting conflict relation among the objectives that allows the conflict-based strategy to perform better than the random strategy. Figure 32 shows the subspaces generated by the conflict strategy on the Knapsack problem with 9 objectives. As can be seen, as the search progresses, a particular partition is formed recurrently, namely $\Psi_3 = \{\{4, 5, 8\}, \{1, 3, 9\}, \{2, 6, 7\}\}$, where $\{4, 5, 8\}$ is the least conflicting subspace, and $\{2, 6, 7\}$ is the most conflicting one. This suggests

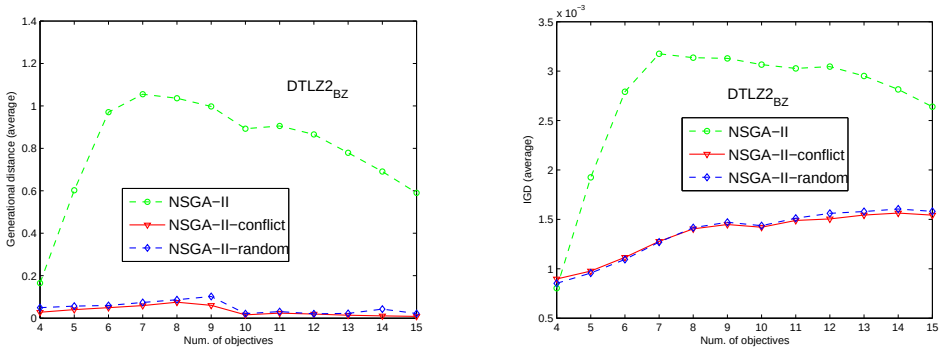


Figure 29: Generational Distance and inverted generation distance in problem $DTLZ2_{BZ}$.

that the conflict between certain objectives is considerably larger than the conflict between other objectives.

In order to measure the contribution of each subspace to the total conflict in the problem, we compute the following measure. For each subspace we compute the sum of the conflict between each pair of its objectives. We consider this sum as the conflict degree of each subspace. The sum of the conflict degree of each subspace is the total conflict of the problem. The ratio of the conflict degree of each subspace and the total conflict is called the conflict contribution. In Figure 33, we can clearly see that subspace 3 has a larger conflict contribution with respect to the other subspaces.

From the results obtained in the generational distance and in the inverted generational distance (see Figure 34) we can say that the conflict-based partition strategy achieved better Pareto front approximations than the random-based strategy in terms of both, convergence and distribution.

The results obtained with the hypervolume indicator (see Figure 35) confirm that the conflict-based strategy outperformed the random strategy. We can conclude that the differences in the degrees of conflict between each pair objectives was used by the conflict-based strategy to obtain better results than those obtained using a random partition.

6.4.7 *Final Remarks*

The experimental results showed that both the conflict-based and random partition strategies outperformed NSGA-II in all the test problems considered in this study. While NSGA-II even diverged in some test problems, the NSGA-II using any of the partition strategies maintained a good convergence regardless of the number of objectives. Regarding the two partition strategies,

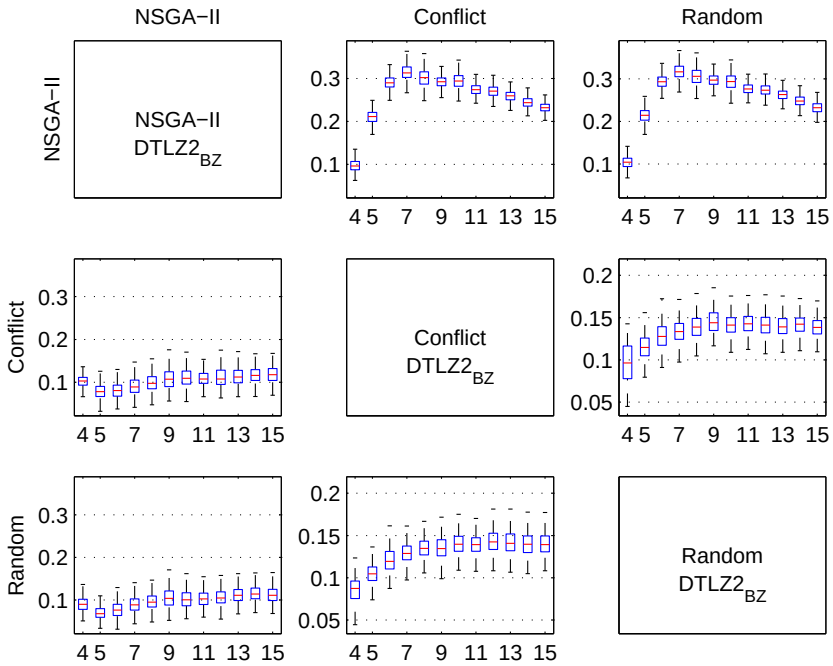


Figure 30: ϵ-indicator results on DTLZ2_{BZ}.

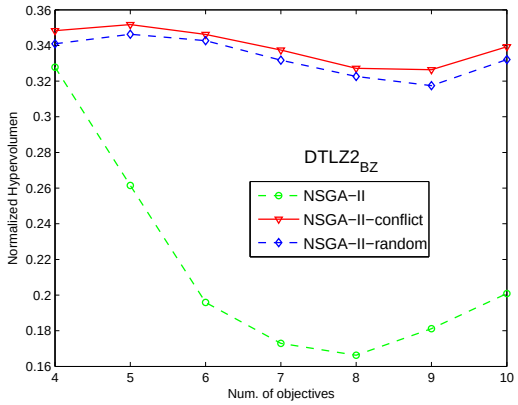


Figure 31: Hypervolume on DTLZ2_{BZ}.

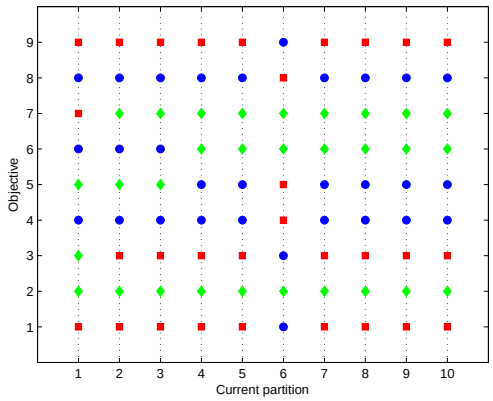


Figure 32: Generated subspaces by the conflict-based partition strategy on the Knapsack problem.

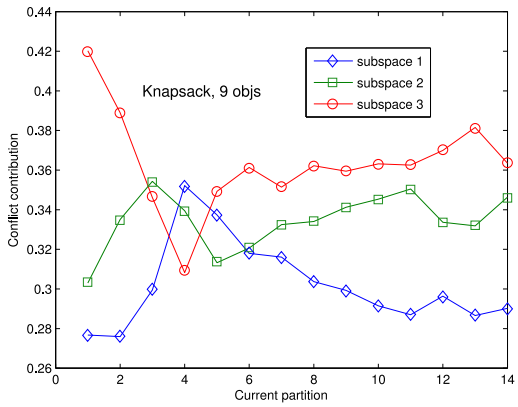


Figure 33: Conflict contribution of each of the three subspaces generated using the conflict partition strategy.

the conflict-based partition strategy achieved a better distribution of the solutions than that achieved by the random strategy. In some problems, by using the random strategy, convergence was concentrated on the extremes of the Pareto front.

In problems in which the degree of conflict between pairs of objectives was different, the conflict-based strategy presented a better performance. It is important to note, that in the case of the Knapsack problem, in which the conflict relation among the objectives is not known a priori, the conflict-based strategy was able to detect important dependencies among the objectives in terms of the conflict. The extracted conflict information allowed our

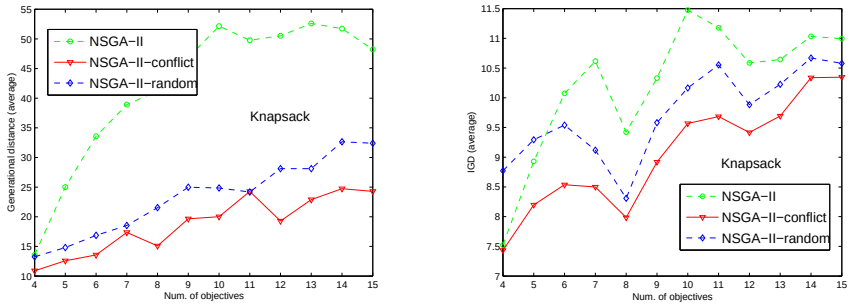


Figure 34: Generational Distance and inverted generation distance in the Knapsack problem.

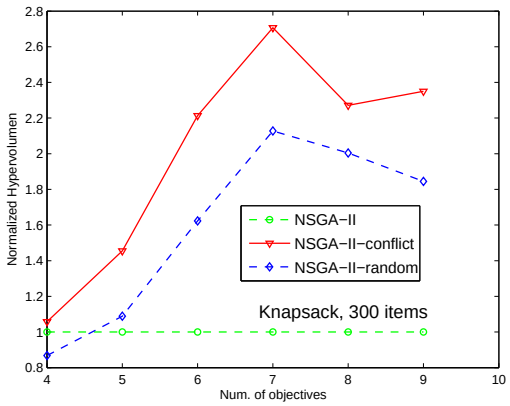


Figure 35: Normalized Hypervolume on the Knapsack problem. The hypervolumen values were normalized with respect to the hypervolume achieved by the NSGA-II.

proposed conflict-based partition strategy to achieve better results than the other algorithms.

Initially, one may think that grouping all the highly conflicting objectives in one subspace is the best choice. However, the experimental results showed that the best size of the subspaces considerably depends on the scalability of the underlying MOEA. For instance, if the underlying MOEA has good performance until 5 objectives, the size of each subspace should not exceed that limit.

From the experimental results we realized that in some problems the contribution of some subspaces to the overall conflict of the problem was very small. Therefore, an equal distribution of the resources (e.g., proportion

of parents, number of generations) to the subspaces might not be a good idea.

7

STUDY OF PREFERENCE RELATIONS IN MANY-OBJECTIVE OPTIMIZATION

Contents

7.1 Quantitative Analysis of the Preference Relations 92
7.2 Final Remarks 104

Multiobjective Evolutionary Algorithms (MOEAs) rely on preference relations to identify high-potential regions of the search space in order to converge to the optimal set. A preference relation is the mechanism to decide if a solution $\mathbf{x}$ is preferable over $\mathbf{y}$ in the search space. In single-objective optimization, the determination of the optimum among a set of given solutions is clear. However, in the absence of preference information, in multi-objective optimization there does not exist a unique or straightforward preference relation to determine if a solution is better than other. As we said in Chapter 2, the preference relation most commonly adopted is the *Pareto dominance relation* [91], which leads to trade-offs among the objectives. This set of trade-offs is the *Pareto optimal set*, and its image in objective space, the *Pareto front*.

As we learned in Chapter 4, although Pareto-based MOEAs have shown an acceptable performance in many real-world problems with 2 or 3 objectives, their performance poorly scale when the number of objectives in the MOP is increased. We also noted that one of the widely accepted reasons for this limitation is that the proportion of nondominated solutions in a population increases rapidly with the number of objectives (see e.g., [42]). As a result, in many-objective problems, the Pareto dominance relation is incapable of providing the necessary information to select the correct solutions in order to steer the search towards the Pareto optimal set. Although this limitation seems to affect only to Pareto-based Multiobjective Evolutionary Algorithms (MOEAs), many-objective problems pose some other difficulties common to any other multi-objective optimizer. For instance, the exponential growth of the number of points required to represent accurately a Pareto front with respect to the number of objectives, and the difficulty to visualize the Pareto front in more than 3 dimensions.

Current works in many-objective optimization have analyzed some preference relations qualitatively by using distribution of solutions' ranks (i.e.,

the number of possible ranks and the number of solutions in each rank), or quantitatively by adopting quality indicators commonly used in the field of evolutionary multiobjective optimization (i.e., hypervolume, coverage, or generational distance). However, we believe that using directly standard quality indicators is not appropriate to compare preference relations since their optimal sets are, roughly speaking, different subsets of the Pareto optimal set. In other words, the preference relations prefer different regions of the Pareto optimal front.

In the absence of particular decision maker's preferences, the generally accepted assumption is that the most interesting solution is the "knee" of the Pareto front, i.e., the region of maximum bulge on the Pareto curve [26]. For this reason, a $PF^{(1)}_{approx}$ generated using a particular preference relation should be preferred over a set $PF^{(2)}_{approx}$ achieved by another preference relation if it has more solutions around the knee than $PF^{(2)}_{approx}$. Therefore, in this chapter we present a comparative study which analyzes the performance of some preference relations based on the distance of their approximation sets to the knee of the Pareto front. The goal of this study is to reveal the advantages and disadvantages of the preference relations incorporated into a MOEA.

7.1 Quantitative Analysis of the Preference Relations

In this section we analyze the preference relations presented in Section 4.3.1, namely: Average Ranking (AR), Maximum Ranking (MR), Favour relation (FR), Preference Order Relation (POR), and Expansion preference relation (ER).

7.1.1 *Quality Indicators and Methods Employed*

As noticed earlier, the optimal solution set of each preference relation is, roughly speaking, a subset of PF_{opt}. As a consequence, although a preference relation is only applied on the current PF_{approx} and the archive is maintained using Pareto dominance, the preferred solutions by the preference relation in the primary population belong to a portion of PF_{opt}. Thus, in spite of the fact that the final PF_{approx} set may contain solutions over all the Pareto optimal front, the solutions included in the optimal solution set of the given preference relation are constantly exploited, and the remainder of the solutions may be suboptimal. We can see this situation by comparing the two PF_{approx} sets presented in Figure 36. These sets were obtained using the relations ER and AR. Clearly, these preference relations promote solutions in different regions of the Pareto front.

This poses a challenge to compare Pareto front approximations achieved using different preference relations. For instance, let's suppose we want to

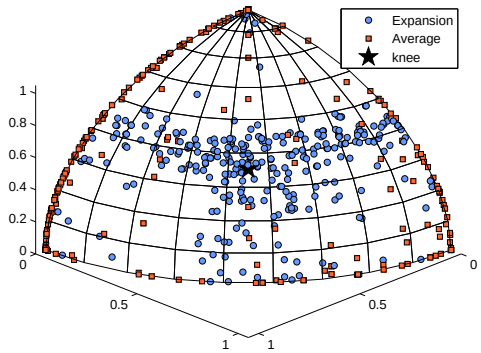


Figure 36: Pareto front approximations obtained by the *expansion* and the *average ranking* relations.

compare two preference relations, one that finds solutions in a small region in the middle part of the Pareto front, and another that finds solutions in a larger region, but in extreme regions of the Pareto front. If we use, for example, the hypervolume indicator, the preference relation with the larger region will have an inherent advantage over the other relation.

It's commonly accepted that decision makers often select a solution located in the middle part of the Pareto front [26], i.e., the knee of the Pareto front. Therefore, we believe that one natural criterion to evaluate preference relations is measuring the distance between the knee and the points in the PF_{approx} set generated using the preference relation. There exist different characterizations of the knee in the literature. Nonetheless, in this thesis we will consider that the knee of the Pareto front is the point with the minimum Chebyshev distance to the ideal point, $\mathbf{z}^*$, or an approximation of it. The weighted Chebyshev distance to $\mathbf{z}^*$ is defined in the following way:

$$d(\mathbf{z}, \mathbf{z}^*, \lambda) = \max_{1 \leqslant j \leqslant k} \{\lambda_j |z_j^* - z_j|\},$$

where k is the number of objectives. Defined this way, the knee of the Pareto front is the point in the feasible objective space, $\mathcal{Z}$, which corresponds to $\min_{\mathbf{z} \in \mathcal{Z}} d(\mathbf{z}, \mathbf{z}^*, \lambda)$. Here, we assume that $\lambda_i = \frac{1}{R_i}$, where R_i is the range of the i-th objective in PF_{opt}. Figure 36 shows the knee for the problem DTLZ2, which is located at the point $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$.

This way, a preference relation is better than other relation if its PF_{approx} set contains more solutions around the knee than the PF_{approx} set achieved by the other relation. In order to evaluate this situation we will plot the distribution of the Chebyshev distance from the ideal point to the points in a given PF_{approx}. The desired distribution is one with a peak near zero and that slowly decays towards the right since it will indicate that most of the solutions are situated close to the knee.

Similar to the approach followed by Corne and Knowles [23], we employed a simple MOEA to evaluate the preference relations included in this study. This way, we try to minimize the effect of some specialized techniques in such a way that the performance of the MOEA can be mainly attributed to the preference relation. Accordingly, the MOEA uses binary encoding, two point crossover, and uniform mutation. To select the parents we used a binary tournament based on the ranks assigned by the given preference relation. The MOEA is equipped with an archive which is truncated by removing a solution selected at random to introduce a new Pareto nondominated solution when the archive is full.

Additionally, for each preference relation we will plot the online generational distance achieved by the current nondominated set generated by the MOEA using the given relation. This way, we can figure out how fast the MOEA converges towards the Pareto front regardless of the spread of the solutions. This information can be useful if we want, for instance, to use a relaxed preference relation to quickly reach the Pareto front and, afterwards, to employ other preference relation to cover a broader extension of the Pareto front. The *generational distance* (GD) [116] is defined by $GD = \left(\sqrt{\sum_{i=1}^{n} d_i^2}\right)/n$, where n is the size of PF_{approx} and d_i is the Euclidian distance between each vector in PF_{approx} and the nearest member of PF_{opt}. Finally, to assess the distribution of the nondominated set obtained by the MOEA we use the *inverted generational distance* (IGD) which is obtained by interchanging the roles of PF_{opt} and PF_{approx} in the GD's definition.

7.1.2 *Experimental Settings*

We adopt the problems DTLZ2 and DTLZ7 to evaluate the performance of the preference relation. The problem DTLZ7 was selected to test the ability of the preference relations to converge towards the knee on problems with a non-convex and disconnected Pareto front[1]. We used 3, 5, 8, 10 and 15 objectives in each problem. Regarding the MOEA, in all the simulations we employed a crossover probability of 0.9 and a mutation probability of $1/\ell$, where ℓ is the length of the binary string needed to encode solutions with 5 digits of precision. For each preference relation the MOEA was run 30 times. In each run we used a population of 200 individuals during 300 generations, and an archive of size 300. The reported values of GD and IGD correspond to the average of the 30 runs, whereas the distributions of the Chebyshev distance were calculated using the union of the PF_{approx} sets achieved by each preference relation.

1 Objectives in DTLZ2 have the same range, however for DTLZ7 we needed to normalize its objectives using the minimum and maximum values of PF_{opt}.

Since the expansion relation requires a user-defined parameter, **S**, in the next section we present a preliminary analysis to investigate the influence of that parameter on the performance of ER.

7.1.3 *Analysis of the Expansion Preference Relation*

In order to investigate the influence of the parameter **S** on the expansion relation we solve DTLZ2 using three different values of **S**, namely $\mathbf{S} = 0.3$, $\mathbf{S} = 0.35$, $\mathbf{S} = 0.4$. Among these values, $\mathbf{S} = 0.3$ is the one that expands the most the dominance area, and, therefore, it may allow the MOEA to reach faster the Pareto front. In each run of these experiments we used a population of 200 individuals during 200 generations, and an archive size of 200.

For the sake of clarity, the online GD values presented in Figure 37 to 39 are plotted on a semi-logarithmic scale. From these plots it is clear that with $\mathbf{S} = 0.3$ it is obtained the fastest convergence to the Pareto front, while with $\mathbf{S} = 0.4$ it is obtained the slowest one. Nevertheless, we have to note that the smaller the **S** value, the smaller the Pareto region covered by the MOEA using the expansion relation.

In order to study the distribution of the solutions around the knee region, we use the distribution of the Chebyshev distances with respect to the origin. Figure 40 shows that for 5 objectives, the larger peak of the distributions for $\mathbf{S} = 0.3$ and $\mathbf{S} = 0.35$ is around 1. That is, most of the solutions are clustered in extreme regions of the Pareto front. However, with $\mathbf{S} = 0.35$ there is a considerable number of solutions near the knee of the front. In fact, this is the value that achieves the largest number of solutions around the knee. With respect to 10 and 15 objectives (see Figure 41), for the three values of **S**, the solutions are concentrated at similar distances from the knee. Since from the three values considered, $\mathbf{S} = 0.35$ represents the best trade-off between convergence and distribution around the knee, we used this value for the rest of the experiments.

7.1.4 *Analysis of All the Preference Relations*

Like in the previous analysis, in the experiments of this section we used the online generational distance and the distribution of the Chebyshev distances. However, we also used the inverted generational distance to measure both the spread and convergence to the Pareto front.

The results of the online generational distance on problem DTLZ2 are presented in Figure 42 to 45. From these plots we can clearly see that, on the one hand, the MOEA with the Pareto relation achieves the worst

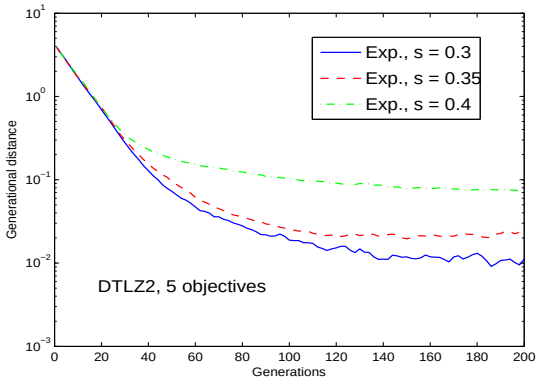


Figure 37: Online GD achieved with the expansion relation using different values for **S** in DTLZ2 with 5 objectives.

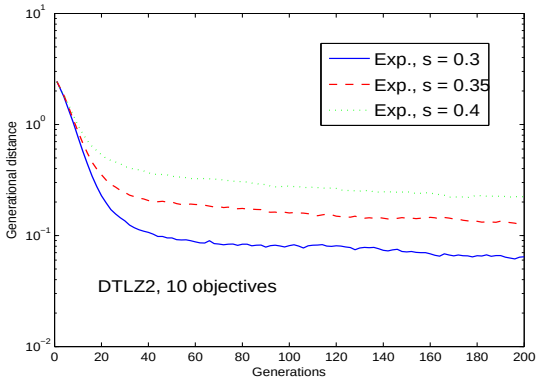


Figure 38: Online GD achieved with the expansion relation using different values for **S** in DTLZ2 with 10 objectives.

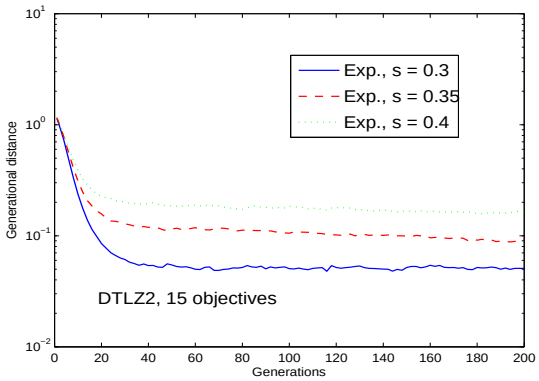


Figure 39: Online GD achieved with the expansion relation using different values for **S** in DTLZ2 with 15 objectives.

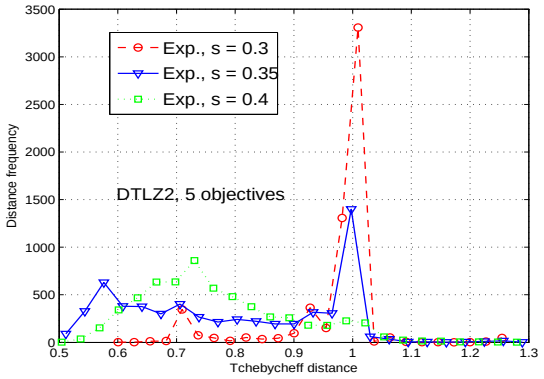


Figure 40: Distribution of the Chebyshev distance obtained with the expansion relation and using different values of **S** in DTLZ2 with 5 objectives.

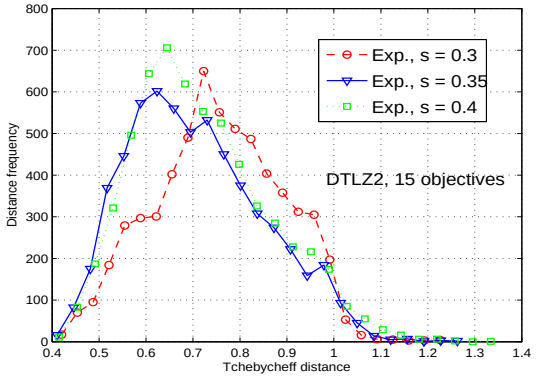


Figure 41: Distribution of the Chebyshev distance obtained with the expansion relation and using different values of **S** in DTLZ2 with 15 objectives.

convergence to the Pareto front during all the search (except for the favour relation). On the other hand, by employing the expansion relation, the MOEA converges very fast during the first 50 and 150 generations for 10 and 15, and 5 objectives, respectively. Then, AR, POR, and FR achieve a closer approximation to the Pareto front.

The second best convergence is obtained using AR, which at the end of the search achieves a better convergence than the expansion relation. The convergence obtained by the favour relation presents an interesting behavior. In the first half of the optimization it achieves a poor convergence. However in the second half, it improves dramatically the convergence and towards the end of the search it produces the best convergence. This behavior is explained by analyzing the IGD results presented in Figure 46 and the distribution of the Chebyshev distance shown in Figure 47 to 50. That is to say, Chebyshev distances show that most of the solutions generated by the MOEA using the favour relation are located far from the knee region (solutions with a Chebyshev distance of 1). In addition, the IGD value for every number of objectives is very poor on problem DTLZ2 (see Figure 46). These two facts suggest that the solutions promoted by the favour relation are concentrated in a small region of the Pareto front and, consequently, after a certain number of generations the solutions overexploit that region achieving very small values on the GD indicator but large values on IGD. This behavior, which is worsened with the number of objectives, is also presented using AR and POR.

On the other hand, MR, ER and the Pareto dominance relations present a wider distribution of the Chebyshev distances for any number of objectives. It is noticeable, though, that the MOEA with the expansion relation finds the closest solutions to the front's knee for any number of objectives. This can be checked by observing that the left tail of its distribution is closer to zero.

With respect to the problem DTLZ2 we can conclude that the best preference relation is the expansion relation since it helps the MOEA to converge quickly to the Pareto front and to maintain more solutions near to the knee of the Pareto front. Although AR and POR provide good convergence they promote solutions away from the knee.

In the problem DTLZ7 we only analyzed the distribution of the Chebyshev distances and the results of the IGD indicator. With 3 objectives, the distribution of the Chebyshev distances achieved by the expansion relation (Figure 51) shows that most of the solutions are located around a distance of 0.8, which is far from the knee of the front. The other preference relations present similar distributions where the solutions are concentrated around 0.75.

However, for more than 3 objectives (Figs. 52 to 54), most solutions achieved by the expansion relation are near the knee of the Pareto front and

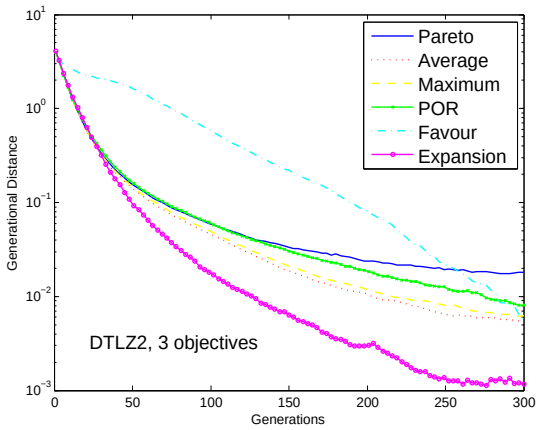


Figure 42: Online GD achieved by the preference relations in DTLZ2 with 3 objectives.

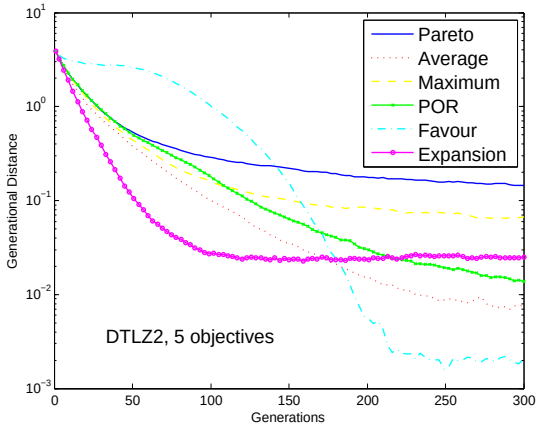


Figure 43: Online GD achieved by the preference relations in DTLZ2 with 5 objectives.

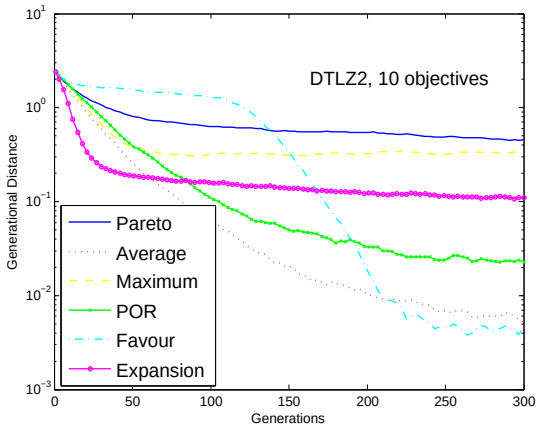


Figure 44: Online generational distance achieved by the preference relations in the problem DTLZ2 with 10 objectives.

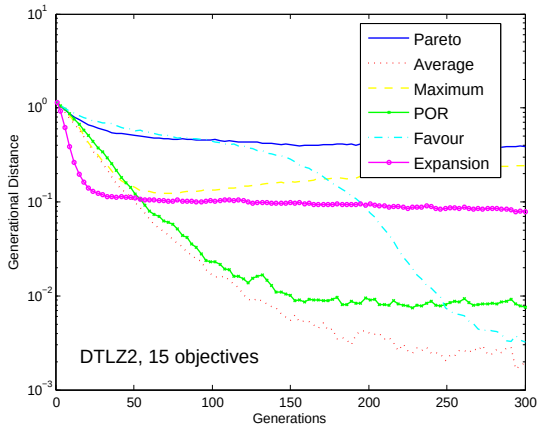


Figure 45: Online generational distance achieved by the preference relations in the problem DTLZ2 with 15 objectives.

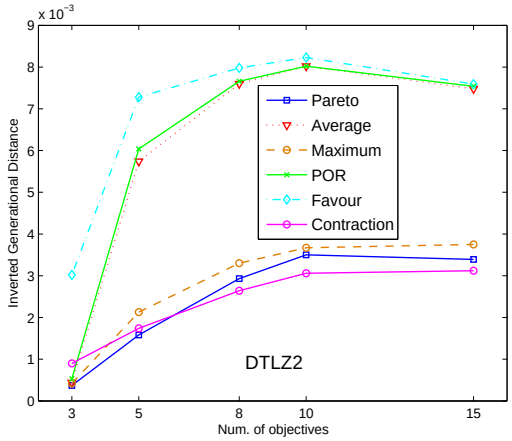


Figure 46: IGD achieved using the preference relations on DTLZ2.

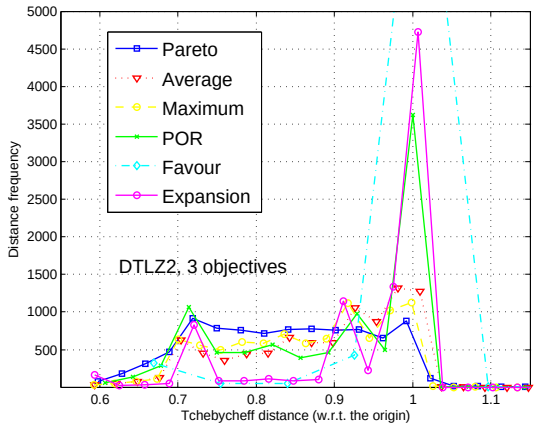


Figure 47: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 3 objectives.

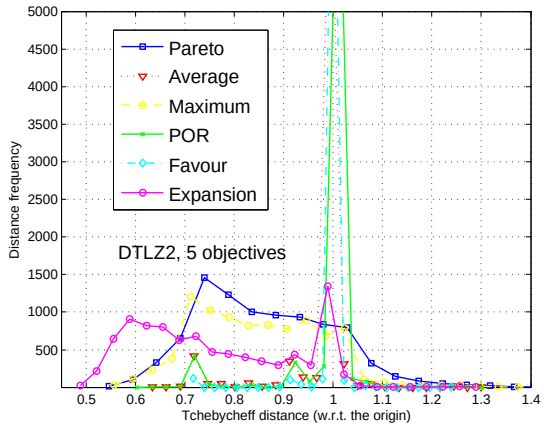


Figure 48: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 5 objectives.

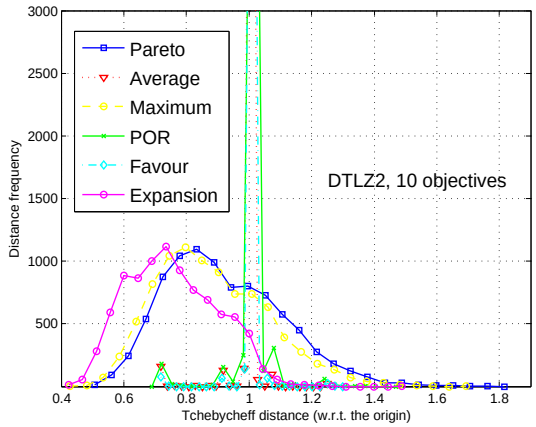


Figure 49: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 10 objectives.

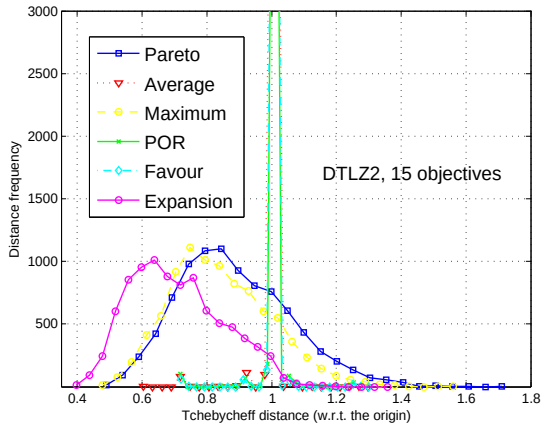


Figure 50: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ2 with 15 objectives.

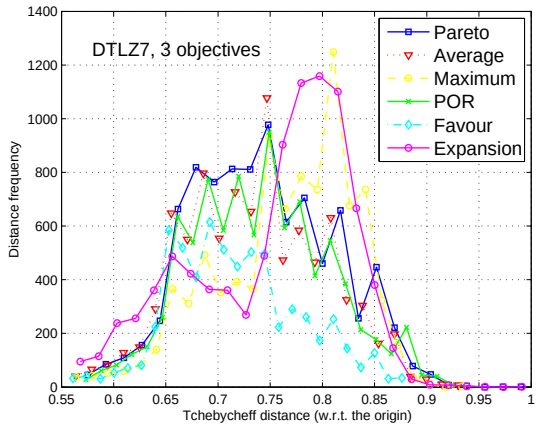


Figure 51: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 3 objectives.

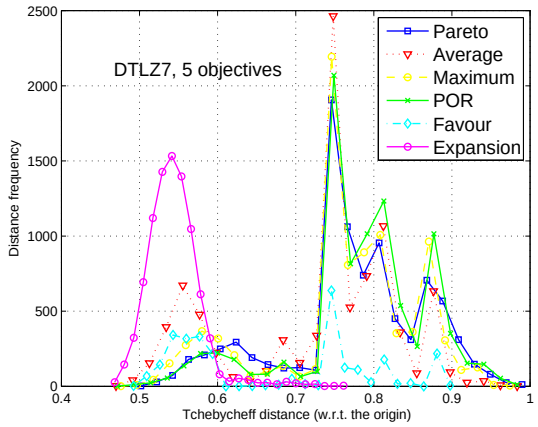


Figure 52: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 5 objectives.

they are the closest solutions to the knee. On the other hand, the MOEA using the Maximum and the Pareto relations generates solutions far away from the true Pareto front (the extreme solutions remain at a distance of 1 since the Chebyshev distance is normalized).

As it can be seen, with these two relations the convergence is worsened as the number of objectives is increased.

The results of the IGD indicator shown in Figure 55 confirm this observation since the maximum and Pareto relations obtain the worst values in this indicator. Since most of the solutions obtained by the expansion relation are clustered around the knee region, it obtained poor values in IGD. In turn, AR and POR obtained the best values in IGD.

In the distribution of the Chebyshev distance from these two relations we can see that there are three peaks in their distributions (specially in Figs. 53

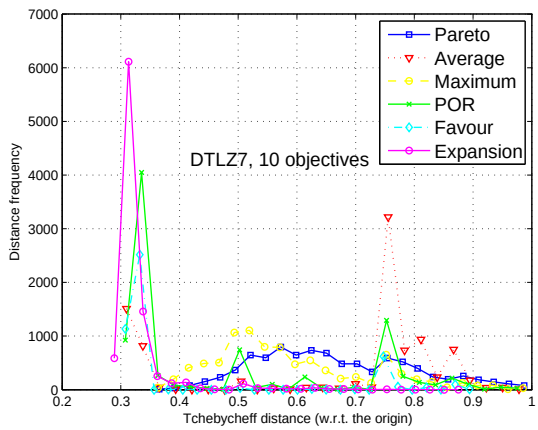


Figure 53: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 10 objectives.

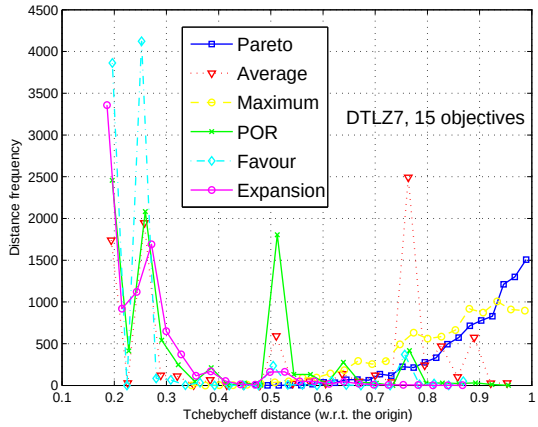


Figure 54: Distribution of the Chebyshev distance over the solutions generated using each preference relation in the problem DTLZ7 with 15 objectives.

and 54), one close to the knee, another in the middle of the distribution, and a third one on the right of the distribution.

This suggests that AR and POR yielded a diverse approximation set concentrated in three regions of the Pareto front, and hence their good performance with respect to IGD.

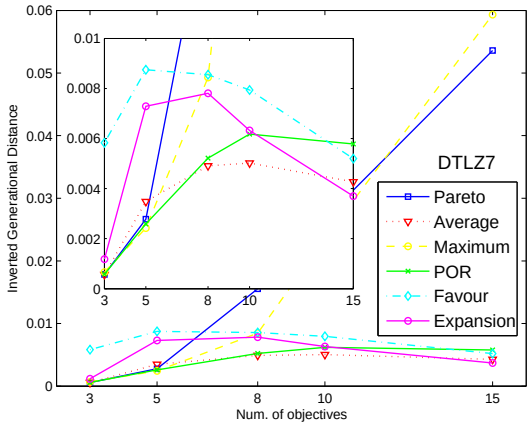


Figure 55: IGD achieved using the preference relations on DTLZ7.

7.2 Final Remarks

The experiments revealed that, in spite of the fact that some preference relations contribute to converge faster to the Pareto front than the Pareto dominance relation, they also stress the generation of solutions far from the knee region. This behavior was observed, for example, in the average ranking and preference order relations in problem DTLZ2. This fact limits the applicability of these relations since, in the general case, it is commonly assumed that the decision maker prefers solutions on the knee region. The *expansion* relation, on the other hand, presented a remarkable performance. In both problems this relation produced a fast convergence to the Pareto front and, in both problems, it achieved solutions very close to the knee region. The second best preference relation was the average ranking relation followed by the preference order and the favour relations. In terms of convergence this result agrees with the conclusions obtained by Corne and Knowles [23].

Although the expansion relation helped to produce solutions near the knee of the Pareto front, in problem DTLZ7, the solutions were concentrated in a small region around the knee. This introduces a trade-off between convergence and the size of the region covered. The parameter of the expansion preference relation opens interesting applications to the relation in MOEAs. For instance, it can be incremented gradually during the search in order

to approach quickly the Pareto front during the first half of the search and then cover the rest of the Pareto front in the second half of the search.

8

A NEW PREFERENCE RELATION TO DEAL WITH MANY-OBJECTIVE PROBLEMS

Contents

8.1 The Reference Point Approach and the Achievement Scalarizing Function 108
8.2 Chebyshev Relation to Guide the Search 111
8.3 An Interactive Method Using the Chebyshev Relation 117
8.4 Evaluation of the Interactive Method 119
8.5 Approximating the Pareto Front with the Chebyshev Relation 126
8.6 Evaluation of the Chebyshev Relation for Approximating the Entire Pareto Front 128
8.7 Final Remarks 145

As discussed in Chapter 2, solving a Multiobjective Optimization Problem involves three stages: model building, search, and decision making (preference articulation). Having a good approximation of the Pareto optimal set does not completely solve a multiobjective optimization problem. The decision maker (DM) still has the task of choosing the most preferred solution out of the approximation set. This task requires preference information from the DM.

Regardless of the stage at which preferences are incorporated into a Multiobjective Evolutionary Algorithm (MOEA), the goals are clear: the aim is to focus on a certain portion of the Pareto front by favoring certain objectives (or trade-offs) over others.

The incorporation of preferences takes a major role in multiobjective optimization problems with a high number of objectives (4 or more), i.e., many-objective optimization problems (see e.g., [42, 94, 117, 63, 64, 99]). The importance of preference incorporation stems from the fact that it can remedy some of the scalability issues observed in Pareto-based MOEAs when the number of objectives is increased (see Chapter 4). One of the scalability problems is the dimensionality of the Pareto front of many Multiobjective Optimization Problems (MOPs) when the number of objectives is high. Nonetheless, we have to point out that there exist some problems in which the dimensionality of the approximation of the Pareto does not grow with the number of objectives (see e.g., [34, 59]).

Therefore, one promising approach to deal with many-objective problems is the use of interactive optimization techniques to avoid the generation and evaluation of millions or even billions of nondominated points as shown in Chapter 4. Furthermore, by incorporating preferences we are inducing a finer ordering over the search space than the one induced by the Pareto dominance relation. Thus, the incorporation of preferences can also cope with the large proportion of nondominated solutions generated in the first generations of the search process.

In this chapter, we present a new preference relation based on an achievement scalarizing function [121]. The purpose of the new preference relation is twofold. On the one hand, it offers a simple approach to integrate decision maker's preferences into a MOEA without modifying the original structure of the MOEA. On the other hand, the proposed reference relation is intended to deal with many-objective problems.

The new preference relation divides the objective function space into two subspaces. The solutions in one of these subspaces are compared using the usual Pareto dominance relation, while the others are compared using the achievement scalarizing function. By means of a reference point, the proposed preference relation allows the decision maker to guide the search towards a certain region of the Pareto optimal front. Each component of the reference point represents the aspiration levels that the decision maker requires for each objective. Furthermore, by using a scalarizing function the developed preference relation induces a finer order on vectors of the objective space than that achieved by the Pareto dominance relation. For this reason, we propose to use the new preference relation to deal with many-objective problems.

8.1 The Reference Point Approach and the Achievement Scalarizing Function

The proposed preference relation is based on the achievement scalarizing function approach proposed by Wierzbicki [121, 122]. An achievement scalarizing function uses a reference point to capture the desired values of the objective functions.

Definition 19 (Achievement scalarizing function)**.** *An achievement scalarizing function (or achievement function for short) is a parameterized function* $s_{\mathbf{z}^{ref}}(\mathbf{z}) : \mathbb{R}^k \to \mathbb{R}$*, where* $\mathbf{z}^{ref} \in \mathbb{R}^k$ *is a reference point representing the decision maker's aspiration levels. Thus, the multiobjective problem is transformed into the following scalar problem:*

$$\begin{aligned} &\textit{Minimize} && s_{\mathbf{z}^{ref}}(\mathbf{z}) \\ &\textit{subject to} && \mathbf{z} \in \mathcal{Z}. \end{aligned} \tag{8.1}$$

A common achievement function, as pointed out in [87, 40], is that based on the Chebyshev distance (L_∞ metric).

Definition 20 (Chebyshev distance)**.** *For two vectors $\mathbf{z}^1, \mathbf{z}^2 \in \mathbb{R}^k$ the Chebyshev distance is defined by*

$$d_\infty(\mathbf{z}^1, \mathbf{z}^2) = \|\mathbf{z}^1 - \mathbf{z}^2\|_\infty = \max_{i=1,\dots,k} |z_i^1 - z_i^2|. \tag{8.2}$$

Based on the Chebyshev distance we can define an appropriate achievement function.

Definition 21 (Weighted achievement function)**.** *The weighted achievement function (or achievement function for short) is defined by*

$$s_\infty(\mathbf{z}, \mathbf{z}^{ref}) = \max_{i=1,\dots,k} \{\lambda_i(z_i - z_i^{ref})\} + \rho \sum_{i=1}^{k} \lambda_i(z_i - z_i^{ref}), \tag{8.3}$$

where $\mathbf{z}^{ref}$ is a reference point, $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_k]$ is a vector of weights such that $\forall i$ $\lambda_i \geqslant 0$ and, for at least one i, $\lambda_i > 0$, and $\rho > 0$ is an augmentation coefficient sufficiently small. The main role of ρ is to avoid the generation of weakly Pareto optimal solutions.

We should note that, unlike the Chebyshev distance, the achievement function does not use the absolute value in the first term. This small difference allows the achievement function to correctly assess solutions that improve the reference point.

The achievement function has some convenient properties over other scalarizing functions. As proved in [108], [87] and [40], the minimum of Eq. 8.3 is a Pareto optimal solution and we can find any ρ-properly Pareto optimal solution, i.e., solutions in which a finite improvement in one objective is possible only at the expense of a reasonable worsening in other objectives. Formally, ρ-properly Pareto optimal solutions are defined as follows.

Definition 22 (ρ-properly Pareto optimality)**.** *A solution $\mathbf{x}^* \in \mathcal{X}$ and its corresponding vector, $\mathbf{z}^* \in \mathcal{Z}$, are ρ-properly Pareto optimal (in the sense of Wierzbicki [121]) if*

$$(\mathbf{z}^* - \mathbb{R}^k_\rho \setminus \{0\}) \cap \mathcal{Z} = \emptyset,$$

where $\mathbb{R}^k_\rho = \{\mathbf{z} \in \mathbb{R}^k | \max_{i=1,\dots,k} z_i + \rho \sum_{i=1}^k z_i \geqslant 0\}$, and ρ is some scalar. The trade-offs among the objectives are bounded by ρ and $1/\rho$.

In most of the reference point methods, the exploration of the objective space is made by moving the reference point at each iteration [83]. In turn,

the weights are kept unaltered during the interactive optimization process. That is, weights do not define preferences, but they are mainly used for normalizing each objective function. Usually, the weights are set for all $i = 1, \ldots, k$ as

$$\lambda_i = \frac{1}{z_i^{\text{nad}} - z_i^{\star\star}}. \tag{8.4}$$

It is important to mention that the DM can provide both feasible and infeasible reference points, or more precisely, $\mathbf{z}^{\text{ref}} \in \mathcal{Z} + \mathbb{R}_+^k$ or $\mathbf{z}^{\text{ref}} \notin \mathcal{Z} + \mathbb{R}_+^k$, where $\mathbb{R}_+^k$ is the nonnegative orthant of $\mathbb{R}^k$. On the one hand, if $\mathbf{z}^{\text{ref}} \in \mathcal{Z} + \mathbb{R}_+^k$, then the minimization of Eq. 8.3 subject to $\mathbf{z} \in \mathcal{Z}$ should represent the maximization of the surplus $\mathbf{z} - \mathbf{z}^{\text{ref}} \in \mathbb{R}^k$. On the other hand, if $\mathbf{z}^{\text{ref}} \notin \mathcal{Z} + \mathbb{R}_+^k$, the minimization of Eq. 8.3 subject to $\mathbf{z} \in \mathcal{Z}$ minimizes the distance between the reference point and the Pareto optimal set.

8.1.1 *MOEAs Based On a Scalarizing Function*

This section reviews some MOEAs that employ the Chebyshev distance in order to rank the individuals of a population. It is important to note that, unlike our approach, these MOEAs use the Chebyshev distance (Eq. 8.2), and not the achievement function (Eq. 8.3) presented in Section 8.1. The main difference with respect to our work is that these methods, instead of defining a preference relation, use the Chebyshev distance in a more straightforward and *ad hoc* manner.

Alves and Almeida [3] proposed a MOEA that carries out multiple single-objective optimizations using a Chebyshev distance function in each optimization. At the beginning of the search a set of instances of the Chebyshev distance function is defined by generating a set of well-distributed weight vectors. This way, the weight vectors define different search directions. Thereafter, the MOEA sequentially and independently performs one search for each weight vector in order to find a small set of solutions around the direction determined by the corresponding weight vector.

Zhang and Li [126] proposed a similar approach. The important difference with respect to the previous approach is that in this method, each single objective optimization uses the information of parallel optimizations in its "neighborhood". A neighborhood is composed of single objective optimizations with nearby weight vectors. Thus, for example, crossover is restricted to solutions in the same neighborhood.

Similarly to the previous approaches, Soylu and Köksalan [106] proposed a MOEA in which the fitness of a solution is based on its weighted Chebyshev distance to the ideal point. The weight vector values used to compute the distance of a given solution are assigned specifically for that solution.

The weight vector values adopted are those that minimize the weighted Chebyshev distance from the given solution to the ideal point. The main purpose of that selection is to favor Pareto optimal solutions over weakly Pareto optimal solutions.

8.2 Chebyshev Relation to Guide the Search

The preference relation proposed here was designed keeping two goals in mind. First, we aimed to provide an easy way to integrate preferences into different types of MOEAs requiring only slight modifications to their structure. The second goal was to investigate the use of achievement functions when dealing with many-objective problems.

In the following sections, we introduce the new preference relation and its use as an interactive technique for multi- and many-objective optimization. Then, we present a modification that is required in order to use the proposed preference relation to generate an approximation of the whole Pareto set in many-objective optimization problems.

8.2.1 *User Reference Point Chebyshev Preference Relation*

The main idea of the proposed preference relation is to combine the Pareto dominance relation and the achievement function to compare solutions in objective function space. The achievement function will allow the incorporation of DM's preferences using a feasible or an infeasible reference point.

We can easily define a simple preference relation using the achievement function. For example, we could say that a vector $\mathbf{z}^1$ will be preferred to $\mathbf{z}^2$ if and only if $s_\infty(\mathbf{z}^1, \mathbf{z}^{\text{ref}}) < s_\infty(\mathbf{z}^2, \mathbf{z}^{\text{ref}})$. However, by doing so, we would obtain only one Pareto optimal solution, which we will denote by $\mathbf{z}^\star_\infty = \arg\min_{\mathbf{z}\in\mathcal{Z}} s_\infty(\mathbf{z}, \mathbf{z}^{\text{ref}})$. In order to find a set of solutions around the point $\mathbf{z}^\star_\infty$, we will allow a threshold, δ, in the preference relation. That is, we want to find the set of points, $\mathbf{z}$, such that $s_\infty(\mathbf{z}, \mathbf{z}^{\text{ref}}) \leqslant s^{\min} + \delta$, where $s^{\min} = \min_{\mathbf{z}\in\mathcal{Z}} s_\infty(\mathbf{z}, \mathbf{z}^{\text{ref}})$ (or in different terms, $s(\mathbf{z}^\star_\infty, \mathbf{z}^{\text{ref}})$). All these points would be located in the dark region shown in Figure 56.

Nevertheless, as we can see in the figure, we obtain both Pareto and dominated solutions. In order to obtain exclusively Pareto solutions we compare those solutions using the Pareto dominance relation. By doing so, only the Pareto nondominated solutions in the square region shown in Figure 57 are considered as the nondominated solutions with respect to the new preference relation developed. In some sense, we can consider that the new relation divides the feasible objective space in two parts as can be seen in Figure 56. The larger part of the feasible objective space is compared with

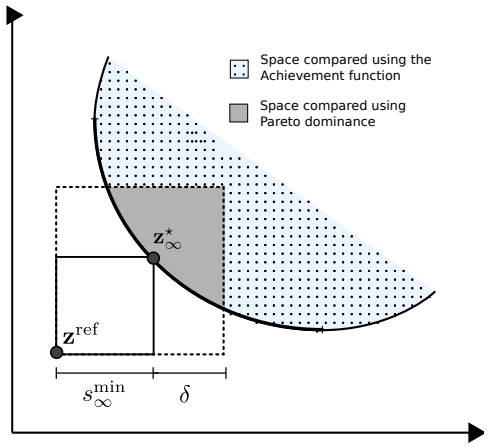


Figure 56: Illustration of how the objective space is divided, and how the vectors in each subspace are compared.

the achievement function, while the remainder of the space is compared adopting the usual Pareto dominance relation. For the sake of simplicity, we will refer to this new relation as the *Chebyshev preference relation*.

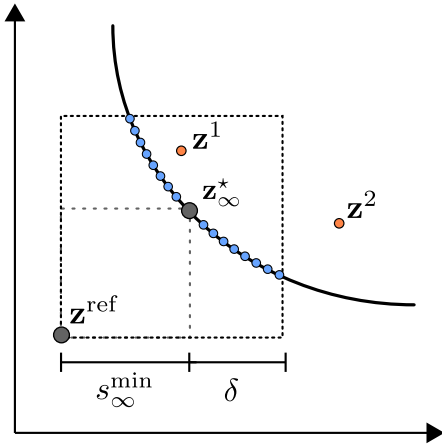


Figure 57: Nondominated solutions with respect to the Chebyshev relation.

Now, we can give a formal definition of the Chebyshev preference relation.

Definition 23 (Chebyshev preference relation)**.** *A solution* $\mathbf{z}^1$ *is preferred to solution* $\mathbf{z}^2$ *with respect to the Chebyshev relation* ($\mathbf{z}^1 \prec_{cheby} \mathbf{z}^2$)*, if and only if:*

1. $s_\infty(\mathbf{z}^1, \mathbf{z}^{ref}) < s_\infty(\mathbf{z}^2, \mathbf{z}^{ref}) \wedge$
   $\{\mathbf{z}^1 \notin N(\mathbf{z}^{ref}, \delta) \vee \mathbf{z}^2 \notin N(\mathbf{z}^{ref}, \delta)\}$, *or,*

2. $\mathbf{z}^1 \preceq_{pareto} \mathbf{z}^2 \wedge \{\mathbf{z}^1, \mathbf{z}^2 \in N(\mathbf{z}^{ref}, \delta)\}$,

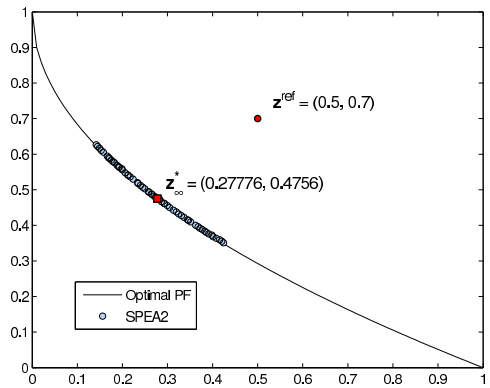
where $N(\mathbf{z}^{ref}, \delta) = \{\mathbf{z} \,|\, s_\infty(\mathbf{z}, \mathbf{z}^{ref}) \leqslant s^{\min} + \delta\}$. *That is, the set of vectors with an achievement better than* $s^{\min} + \delta$ *with respect to the vector of aspiration levels* $\mathbf{z}^{ref}$.

As an illustration of the preference relation, consider solutions $\mathbf{z}^1$ and $\mathbf{z}^2$ presented in Figure 57. Since $\mathbf{z}^2 \notin N(\mathbf{z}^{\text{ref}}, \delta)$ and $s_\infty(\mathbf{z}^1, \mathbf{z}^{\text{ref}}) < s_\infty(\mathbf{z}^2, \mathbf{z}^{\text{ref}})$, then $\mathbf{z}^1 \prec_{\text{cheby}} \mathbf{z}^2$.

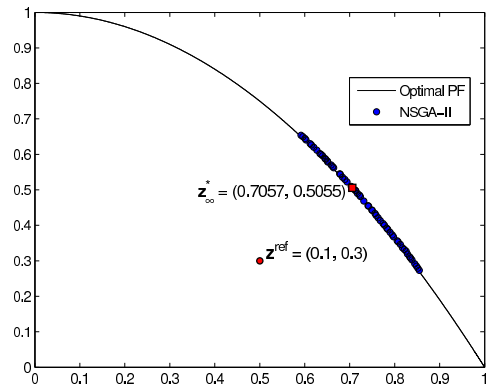
Figure 58 shows the use of the Chebyshev preference relation in the Nondominated Sorting Genetic Algorithm II (NSGA-II) [33] and the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [129]. As we can see in Figure 58, unlike some distance metrics, the achievement function (Eq. 8.3) allows a MOEA to find points in problems with nonconvex Pareto fronts. Moreover, the figure shows how the DM can provide both feasible and infeasible reference points. Also, we have to note the result obtained in problem DTLZ2. If we had used the Euclidean distance to define the preference relation, with $\mathbf{z}^{\text{ref}} = 0$ we had obtained nondominated solutions over the entire Pareto front. The reason for this, is that all the vectors in DTLZ2's Pareto optimal front are situated on a sphere of radius 1.

In order to incorporate the Chebyshev relation into the two previously mentioned MOEAs we only have to change the usual Pareto dominance checking procedure by the function that implements the new relation. In order to have an efficient procedure, the evaluation of the achievement function was computed and stored for each solution before each ranking process. This way, the comparisons required to rank the current population use the stored values of the achievement function.

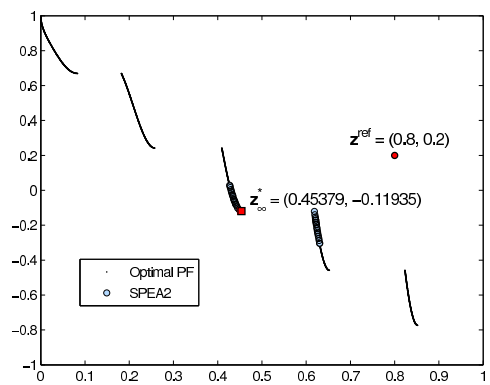
In practice, it might be difficult to set a value for the parameter δ since it does not have an upper bound that is known *a priori*. In order to have a better control of this parameter during the search, we can set it in terms of the proportion of the current range of the achievement function (namely, the difference between the minimum and maximum achievement with respect to a given solution set P). If τ is that proportion, then $\delta = \tau \cdot (s^{\max} - s^{\min})$, where $s^{\max} = \max_{\mathbf{z} \in P} s_\infty(\mathbf{z}, \mathbf{z}^{\text{ref}})$ and $s^{\min} = \min_{\mathbf{z} \in P} s_\infty(\mathbf{z}, \mathbf{z}^{\text{ref}})$. As a consequence, for $\tau = 0$ we would only find the minimum of the achievement function, whereas if $\tau = 1$, then we would get the usual Pareto dominance relation since for every solution $\mathbf{z} \in P$, $\mathbf{z} \in N(\mathbf{z}^{\text{ref}}, \delta)$. Therefore, the DM can use the value of τ for adjusting the size of the region of interest around solution $\mathbf{z}^\star_\infty$. For example, if $\tau = 0.1$ and P is the current approximation set of the



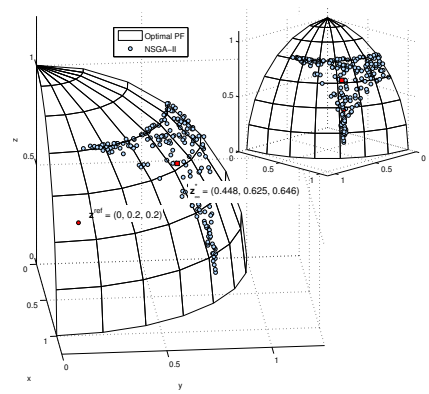
(a) ZDT1: feasible reference point and convex Pareto front.



(b) ZDT2: infeasible reference point and concave Pareto front.



(c) ZDT3: feasible reference point and nonconvex Pareto front.



(d) DTLZ2: infeasible reference point and concave Pareto front.

Figure 58: Illustration of the Chebyshev preference relation incorporated into NSGA-II and SPEA2, using feasible and infeasible reference points. $\mathbf{z}^{\star}_{\infty}$ is the optimum of the achievement function with respect to the current population of the MOEA. In all the examples, we used a threshold $\delta = 0.2$.

Pareto front, only solutions with an achievement value 10% greater (with respect to the range of s_∞ in P) than the one of $\mathbf{z}^\star_\infty$ will be found. In our approach, the values of the weight vector, λ, that appears in Eq. 8.1 are set according to Eq. 8.4. The vectors $\mathbf{z}^{\star\star}$ and $\mathbf{z}^{nad}$ are approximated using the current PF_{approx} achieved by the MOEA.

8.2.2 *Central-guided Chebyshev Preference Relation*

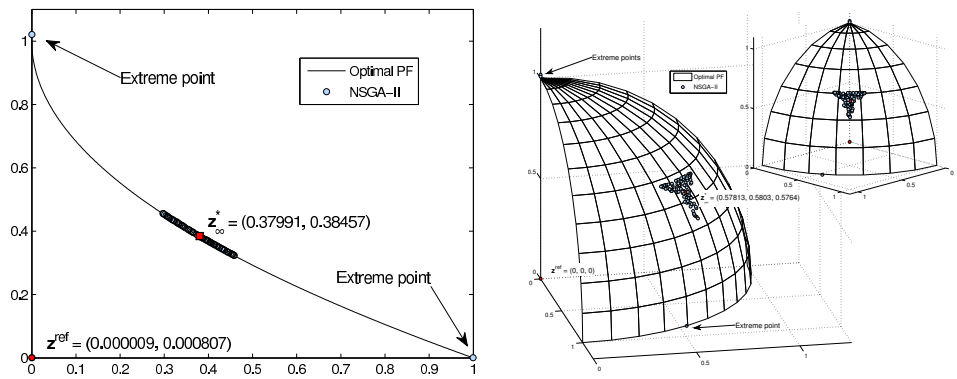
As previously mentioned, in many-objective problems the number of points needed to represent a Pareto front accurately grows exponentially with the number of objectives. Therefore, in many cases trying to approximate the whole Pareto front is not convenient. Additionally, in a many-objective context it might be very difficult for the DM to select a final solution.

When the DM does not have any knowledge about the MOP to be solved (e.g., trade-offs among the objectives, variation range of the objectives), a good idea might be to aim to converge to the ideal point, in which all the objectives are minimized simultaneously. In some cases, the solution that minimizes the distance to the ideal point is located in the central part of the Pareto front. If the Pareto front is symmetric, the closest solution to the ideal point is equivalent to the so-called knee of the front [26, 85, 9, 102].

The point of interest to us corresponds to the minimum of the achievement function using the ideal point, $\mathbf{z}^\star$, or an approximation of it, as the reference point.

In order to achieve the desired behavior we need to approximate the ideal point during the search process of the MOEA. To do so, we will use the lower bounds of the current approximation of the Pareto front. At each iteration we will determine one of the vectors that minimizes each objective separately. That is, we need to find the set of k vectors $\Phi = \{\mathbf{z}^1, \ldots, \mathbf{z}^k \mid z^i_i = z^{min}_i, i = 1, \ldots, k\}$, where $z^{min}_i = \min_{z \in PF_{approx}} z_i$.

There are some works, in which an evolutionary algorithm has been used to approximate the ideal point [102] or the nadir point [35]. Nonetheless, these approaches require a modification in a particular component of the MOEA (for instance, in the crowding operator or in the archive). In order to maintain the preference relation independent of an external module, e.g., an archive, we propose to modify the Chebyshev relation to implicitly maintain the extreme or boundary solutions Φ. To this end, besides emphasizing the points close to the central part of the Pareto front, the relation does not allow that extreme points are dominated.



(a) ZDT1. (b) DTLZ2.

Figure 59: Illustration of the central-guided Chebyshev preference relation incorporated into NSGA-II. In these plots $\mathbf{z}^{\text{ref}}$ is the approximation of the ideal point, and $\mathbf{z}^{\star}_{\infty}$ is the the vector that we consider the central point of the Pareto front.

Definition 24 (Central-guided Chebyshev preference relation)**.** *A solution* $\mathbf{z}^1$ *is preferred to solution* $\mathbf{z}^2$ *with respect to the central-guided Chebyshev preference relation* ($\mathbf{z}^1 \prec_{c\text{-}cheby} \mathbf{z}^2$) *if and only if:*

$$\mathbf{z}^1 \prec_{cheby} \mathbf{z}^2, \text{ and } \mathbf{z}^2 \notin \Phi.$$

Figure 59 shows the Pareto front approximation obtained by NSGA-II using the central-guided Chebyshev preference relation with the approximated ideal point as a reference point. The figure shows the extreme points of problems ZDT1 and DTLZ2. It is worth noting that in both problems, the approximation of the ideal points are very accurate.

In an interactive optimization process, those points are useful to estimate the range of the Pareto optimal front.

This variant of the proposed preference relation might be very useful in many-objective problems in which traditional visualization techniques, such as 2D or 3D plots, are no longer available. In this case, the DM can be assisted by the preference relation to find a set of solutions around the (usually) most interesting region of the Pareto front.

One of the advantages of the basic Chebyshev preference relation and the central-guided variant over other preference relations is their low time complexity. The evaluation of the achievement function for the entire population has complexity $O(km)$, where m is the size of the population and k is the number of objectives. Regarding the central-guided variant, the process of finding the extreme points has complexity $O(km)$. Therefore, the total process of the central-guided variant also has complexity $O(km)$. In order to illustrate the computational savings using the Chebyshev relation, let us take

as an example, the ranking procedures of NSGA-II and Multiobjective Genetic Algorithm (MOGA) [50]. Both NSGA-II's nondominated sorting [107, 33] and MOGA's nondominated ranking [49] have complexity $O(km^2)$ using the Pareto dominance relation. Using any of the Chebyshev relations we need to compare a single real value instead of a k-dimensional vector for each pair of solutions. Therefore, using the Chebyshev relation, these ranking procedures have complexity $O(km + m^2)$. Figure 60 shows the complexities of the ranking procedures using the Pareto relation, and any of the Chebyshev relations, respectively. In this discussion we have assumed that the entire population is exclusively compared using the achievement function. In practice, however, the actual complexity depends on the proportion of solutions compared using the achievement function and the usual Pareto dominance relation. Nonetheless, as the threshold τ decreases, the resulting complexity tends to the one defined above. For instance, if $\tau = 0.1$, approximately 90% of the population is compared using the Chebyshev relation.

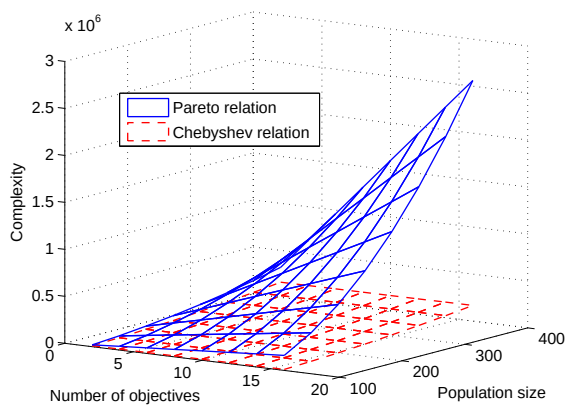


Figure 60: Plots of the complexity of NSGA-II and MOGA's ranking procedures using the Pareto dominance relation ($O(km^2)$), and the Chebyshev relation ($O(km + m^2)$).

8.3 An Interactive Method Using the Chebyshev Relation

The two variants of the Chebyshev preference relation can be used in an interactive way. When the DM does not have enough knowledge about the problem to provide a reference point, the central-guided Chebyshev relation can be used to obtain a first set of solutions. However, in real situations it is common that the DM counts with a previous best known solution of the given problem. In that case, the previous solution can serve as a

good reference point. Then, the process can follow the usual steps of the interactive techniques. That is, at each iteration the DM must provide new aspiration levels in the form of a reference point. Additionally, the DM can change the value of the threshold δ (or can change it proportionally using τ) that controls the size of the set of solutions. For example, the user can set $\tau = 0.5$ in order to obtain about half of the Pareto front around the reference point. This helps the DM to know the trade-offs among the objectives. At subsequent iterations, the value of τ could be reduced to concentrate the search towards a region of interest chosen by the DM. In order to show the set of solutions of the region of interest, some visualization tool designed for problems with more than three objectives could be used, such as parallel coordinates plots, heatmap graphs, or scatter plots (see e.g., [10]). To ease the visualization of the solutions, a technique for truncating the approximation set can be used. For example, a clustering technique can be employed, such as the one used in SPEA2 [129], or a technique similar to the archiving methods. Therefore, the interactive process requires an additional parameter indicating the number of solutions to visualize.

In a next step, the employed MOEA is again executed using the Chebyshev relation in order to find a new set of solutions that best satisfies the aspirations of the DM. This process continues until the DM is satisfied with a solution of the current set of solutions. Algorithm 7 shows the whole interactive process.

Algorithm 7 Interactive technique using the Chebyshev preference relation.

Step 1: Ask the DM to specify the threshold τ. If the DM has some knowledge about the problem, he/she can provide a reference point. Otherwise, the central-guided preference relation can be used to converge towards the ideal point.

Step 2: **If** a reference point was provided, **then**

Execute the MOEA using the Chebyshev relation with the reference point provided by the decision maker.

else

Execute the MOEA using the central-guided Chebyshev relation.

Step 3: Ask the DM to define how many solutions of the current approximation should be shown. Additionally, from the use of the central-guided relation the DM can be informed of the current ideal point in order to decide the new aspiration levels.

Step 4: **If** the DM is satisfied with some solution of the current set, **then**

STOP.

else

Go to **Step 1**.

8.4 Evaluation of the Interactive Method

8.4.1 *Airfoil Shape Problem with 2 Objectives*

In order to illustrate the interactive method presented in the previous section we will use a multiobjective aerodynamic airfoil shape optimization problem adapted from [111], and having 2 objectives. The goal is to optimize the shape of a standard-class glider, aiming at obtaining optimum performance for a sailplane.

Two conflicting objective functions are defined in terms of a sailplane average weight and operating conditions [111]:

Objective functions of the airfoil shape problem.

1. Min $f_1 = C_D/C_L$,
   s.t. $C_L = 0.63, Re = 2.04 \times 10^6, M = 0.12$.
2. Min $f_2 = C_D/C_L^{3/2}$,
   s.t. $C_L = 1.05, Re = 1.29 \times 10^6, M = 0.08$.

Objective f_1 represents the inverse of the glider's gliding ratio, whereas f_2 represents the sink rate. Both objectives are important performance measures for this aerodynamic optimization problem. C_D and C_L are the drag and lift coefficients. Each objective is evaluated at different prescribed flight conditions, given in terms of Mach and Reynolds numbers. The aim of solving this MOP, is to find a better airfoil shape, which improves a reference design.

In the present case study, a modified PARSEC airfoil representation [104] is used. Figure 61 illustrates the 12 basic parameters used for this representation: $r_{le_{up}}$ / $r_{le_{lo}}$ leading edge radius for upper/lower surfaces, X_{up}/X_{lo} location of maximum thickness for upper/lower surfaces, Z_{up}/Z_{lo} maximum thickness for upper/lower surfaces, Z_{xxup}/Z_{xxlo} curvature for upper/lower surfaces, at maximum thickness locations, Z_{te} trailing edge coordinate, ΔZ_{te} trailing edge thickness, α_{te} trailing edge direction, and β_{te} trailing edge wedge angle. The PARSEC geometry representation adopted allows us to define independently the leading edge radius, both for upper and lower surfaces (the original representation uses the same value both for upper and lower surfaces). Thus, 12 variables are used in total. Their allowable ranges are defined in Table 17.

Geometry parameterization of the problem.

The PARSEC airfoil geometry representation uses a linear combination of shape functions for defining the upper and lower surfaces. These linear combinations are given by:

$$Z_{upper} = \sum_{n=1}^{6} a_n x^{(n-1)/2}, \quad Z_{lower} = \sum_{n=1}^{6} b_n x^{(n-1)/2} \tag{8.5}$$

	A720		NLF0416	
Variable	**Lower**	**Upper**	**Lower**	**Upper**
$r_{le_{up}}$	0.0085	0.0126	0.0055	0.0215
$r_{le_{lo}}$	0.0020	0.0040	0.0055	0.0215
α_{te}	7.0000	10.0000	-2.0000	21.0000
β_{te}	10.0000	14.0000	1.0000	15.0000
Z_{te}	-0.0060	-0.0030	-0.0200	0.0200
ΔZ_{te}	0.0025	0.0050	0.0000	0.0000
X_{up}	0.4100	0.4600	0.2875	0.5345
Z_{up}	0.1100	0.1300	0.0880	0.1195
Zxx_{up}	-0.9000	-0.7000	-1.0300	-0.4200
X_{lo}	0.2000	0.2600	0.3060	0.5075
Z_{lo}	-0.0230	-0.0150	-0.0650	-0.0500
Zxx_{lo}	0.0500	0.2000	-0.0490	0.8205

Table 17: Parameter ranges for the PARSEC airfoil representation for problems A720 (2 and 3 objs.) and NLF0416 (6 objs.).

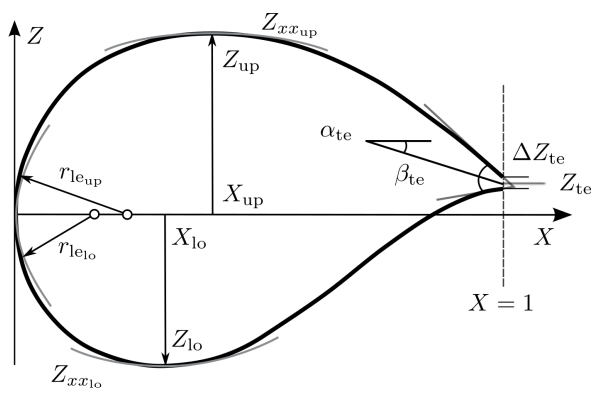


Figure 61: PARSEC airfoil parametrization.

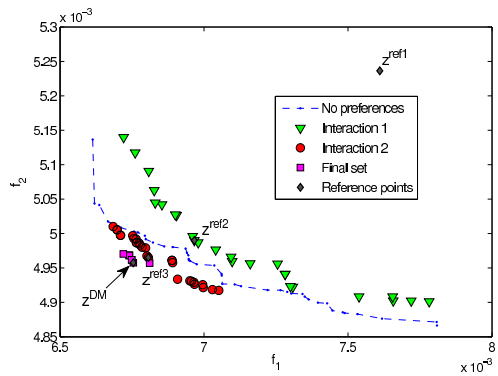


Figure 62: Simulation of the interactive method.

The coefficients a_n, and b_n are determined as function of the 12 geometric parameters by solving two systems of linear equations, one for each surface. It is important to note that the geometric parameters $r_{le_{up}}/r_{le_{lo}}$, X_{up}/X_{lo}, Z_{up}/Z_{lo}, $Z_{xx_{up}}/Z_{xx_{lo}}$, Z_{te}, ΔZ_{te}, α_{te}, and β_{te} are the actual design variables in the optimization process. In turn, coefficients a_n, b_n serve as intermediate variables for interpolating the airfoil's coordinates, which are used by the Computational Fluid Dynamics (CFD) solver (we used the Xfoil CFD code [38]) for its discretization process.

Next, we will show a simulation of the interactive process using NSGA-II with the Chebyshev relation based on a reference point. We adopted the following parameters for NSGA-II: a crossover probability of 0.9, a mutation probability of $1/n$ (n is the number of decision variables), and the distribution indices for crossover and mutation were set as 15 and 20, respectively. A population composed of 60 individuals was employed.

In the first step of the process, we used $\tau = 0.8$ in order to get a global perspective of the entire Pareto front. As a reference point we employed the vector $\mathbf{z}^{ref} = [0.007610, 0.005236]$. This reference point corresponds to the evaluation of a reference airfoil shape A720 [111] in both objectives. Then, NSGA-II was executed for 15 generations. The resulting approximation set is shown in Figure 62 (denoted by triangles). As can be seen, the reference point was dominated by almost all solutions in the approximation set. This illustrates how the relation is able to correctly compare solutions better than the reference point provided. On the other hand, due the nature of the objective space of the problem, only 25 solutions, from the total of 60, are nondominated. Therefore, in this case, the clustering technique to reduce the size of the approximation set was not needed.

Since the initial reference point was improved, we decided to chose one solution of the approximation set as the next reference point, namely, the nearest solution to the ideal point (diamond). For the next execution the

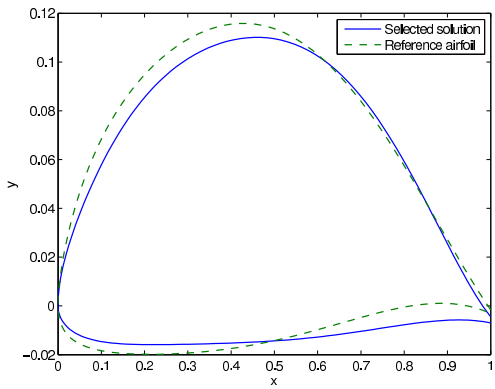


Figure 63: Airfoil of the most preferred solution from the simulation of the interactive method.

region of interest was reduced to $\tau = 0.2$. Similar to the previous DM interaction, the next reference point was the nearest solution of PF_{approx} to the ideal point. In order to obtain a final approximation to select the most preferred solution, the region of interest was reduced to a small region using $\tau = 0.05$. This time NSGA-II was executed for 40 generations. At this stage only 8 solutions were obtained and the most preferred solution for the DM was the one with objective values $[0.006754, 0.004957]$. Figure 63 shows the airfoils corresponding to the initial reference point and to the most preferred solution. In this example, an improvement of approximately 11.24% and of 5.32% was attained for the first and second objective, respectively. From a practical point of view, these improvements are quite significant in increasing the aerodynamic efficiency of the sailplane.

Figure 62 also shows the PF_{approx} achieved by NSGA-II with no preferences during the same number of generations than that used in the interactive method. As one can expect, the final approximation set obtained articulating preferences is closer to the ideal point than the one generated with no preferences. This can be explained by the fact that the incorporation of preferences concentrate all the function evaluations to improve the region of interest. On the other hand, when the task is to approximate the entire Pareto front, some function evaluations are used to approximate regions outside the region of interest. These are clearly different tasks, and therefore, a fair performance comparison is not possible. Nonetheless, we want to emphasize the computational savings of using an interactive approach with respect to the use of an *a posteriori* approach, specially when the function evaluations are expensive in terms of CPU time.

Beside the parameters of the MOEA, the parameters that have to be selected by the DM are the reference point, τ, and the number of solution to be visualized. The selection of a new reference point impose a low cognitive

load since its interpretation is intuitive to the DM. In turn, the parameter τ can be easily set since it is given in terms of the current range approximation of the Pareto front.

8.4.2 *Airfoil Shape Problem with 3 Objectives*

Here, we will evaluate the interactive method using two airfoil shape optimization problems with 3 and 6 objectives, respectively. This time, we will simulate the DM using the Chebyshev achievement function. Specifically, at each interaction point, the new reference point will be the solution in the current PF_{approx} with the best achievement value (which is to be minimized). For the simulation of the 3-objective problem we used 4 interaction points with the DM during the search, and for the 6-objective problem we used 3 interaction points. The parameters at each interaction point are shown in Table 18. The initial threshold for both problems was set to $\tau = 0.8$.

Problem		**Int. 1**	**Int. 2**	**Int. 3**	**Int. 4**
3-obj	Gen	15	35	55	80
	τ	0.5	0.2	0.1	0.025
6-obj	Gen	15	35	55	–
	τ	0.43	0.18	0.025	–

Table 18: Parameter values at each interaction point.

In order to evaluate the performance of the interactive method, for each run, the best achievement value of the final PF_{approx} was measured. As a reference, we also computed the best achievement value obtained by NSGA-II with no preferences. The 3-objective problem is a variant of the problem A720 in which the first and third objectives are objectives f_1 and f_2 of the 2-objective problem of the previous section. The second objective is defined as

- Min $f_2 = C_D/C_L$,
  s.t. $C_L = 0.86, Re = 1.63 \times 10^6, M = 0.1$.

The bounds for the variables are the same described in Table 17. For this problem, we used the vector $[0.007610, 0.005895, 0.005236]$ as our initial reference point. The results for the 3-objective problem are shown in Table 19. As can be seen, both approaches yield achievement values results less than zero, which means that the reference point was improved in all cases. In addition, as expected, the interactive approach obtained better results than the approach with no preferences articulated. The solution with the best

	Best	Median	Worst	Std. dev.
Preferences	-0.2196	-0.2111	-0.1982	0.0047
No prefs.	-0.2183	-0.2020	-0.1816	0.0101

Table 19: Statistics of the achievement function values obtained with preferences and without them in the 3-objective problem.

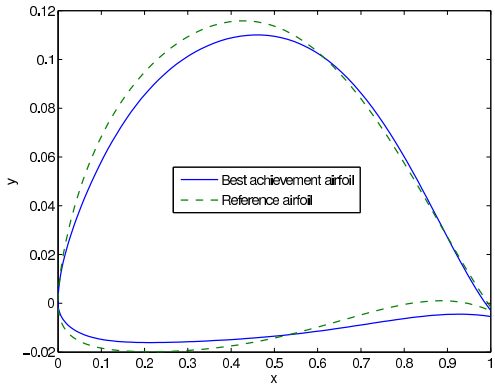


Figure 64: Airfoil with the best achievement value and the reference airfoil for the problem with 3 objectives.

achievement value was $[0.006772, 0.005244, 0.004960]$. Objectives were improved by 11.01%, 11.04% and 5.27%, respectively. The airfoil of this solution is presented in Figure 64, along with that of the reference point.

8.4.3 *Airfoil Shape Problem with 6 Objectives*

The 6-objective problem was taken from [120]. The goal of this problem is to optimize the airfoil shape of a low-speed unmanned aerial vehicle to cover a range of different flight condition (e.g., take-off and cruise). The 6 objectives to be minimized are described in Table 20, and the bounds for the variables are presented in Table 17.

As a reference point we employed a representative profile of the NLF series, namely the NLF0416 [105],

$$\mathbf{z}^{\text{ref}} = [0.00523, 0.00595, 0.01048, 0.33373, 0.90135, 2.93083].$$

The results presented in Table 21 show that for this problem the reference point was not improved by any of the two approaches. However, the interactive approach found better airfoils than those obtained by the approach without preferences. The solution corresponding with the best achievement value found by the interactive approach is the following: $[0.004962, 0.007022, 0.007275, 0.346273, 0.920056, 2.929393]$.

Objective	Comments
$f_1 = C_d$	$C_l = 0.5, Re = 4 \times 10^6, Ma = 0.3$
$f_2 = C_d/C_l^{3/2}$	$Re = 4 \times 10^6, Ma = 0.3$
$f_3 = C_{m_0}^2$	$Re = 4 \times 10^6, Ma = 0.3$
$f_4 = 1/C_{max}^2$	$Re = 4 \times 10^6, Ma = 0.3$
$f_5 = 1/C_l^2$	$\alpha = 5°, Re = 2 \times 10^6, Ma = 0.15$
$f_6 = 1/x_{tr}$	$\alpha = 5°, Re = 2 \times 10^6, Ma = 0.15$

Table 20: Objectives of the airfoil design problem with 6 objectives.

	Best	Median	Worst	Std. dev.
Preferences	0.0047	0.0473	0.0914	0.0183
No prefs.	0.0157	0.2506	0.4787	0.1480

Table 21: Statistics of the achievement function values obtained with preferences and without them in the 6-objective problem.

This solution improves objectives f_1, f_3 and f_6 by an amount of 5.12%, 30.58% and 0.04%, respectively. The airfoil of this solution is presented in Figure 65. Since this problem has local Pareto fronts, we believe that this feature avoids improving the reference point. For this reason, in the next section we analyze the relation of the convergence and the size of the Region of Interest (ROI) in the presence of several local Pareto fronts.

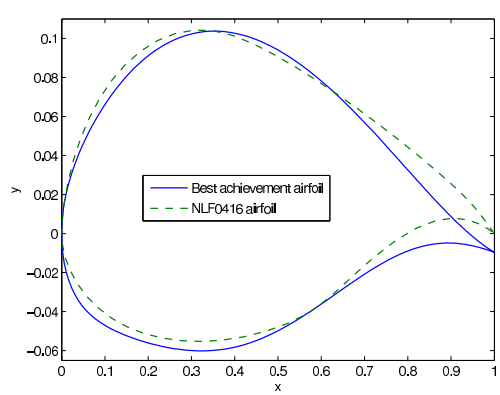


Figure 65: Airfoil with the best achievement value and the reference airfoil for the problem with 6 objectives.

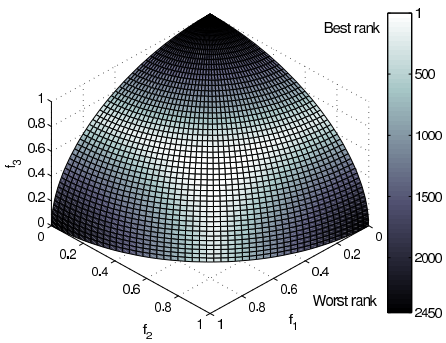


Figure 66: Ranks generated using the central-guided Chebyshev preference relation on a population of Pareto-nondominated solutions. Best ranked solutions are located in the central region of the Pareto front.

8.5 Approximating the Entire Pareto Front Using the Chebyshev Relation

In contrast to the previous section, here we propose the use of the Chebyshev relation to approximate the whole range of the Pareto front as it is usual in the field of evolutionary multiobjective optimization. Although, in general, the number of points of a discrete approximation of the Pareto front increases exponentially with the number of objectives, there are some particular problems in which this is not the case. Those problems are called degenerate problems [59] since their Pareto front has less than $k-1$ dimensions (i.e., the expected dimensionality of a k-objective MOP), even if the feasible objective function space is a k-dimensional object. In this case, while the number of objectives can be large, the number of points to represent the Pareto front can be affordable. An example of a degenerate problem is the DTLZ5(I, M) problem [99]. This problem has M objectives, however, its Pareto front is an $(I-1)$-dimensional object $(I < M)$.

As previously mentioned, the Chebyshev relation can help to rank solutions considered as incomparable by the Pareto dominance relation. Figure 66 shows the distribution of the ranks assigned to a set of Pareto-nondominated solutions by the nondominated sorting coupled with the central-guided Chebyshev relation. As can be seen, solutions located in the central region of the Pareto front obtain the best ranks, while the solutions at the extreme portions of the Pareto front have the worst ranks. In addition, since the reference point is automatically set in the central-guided relation, we believe that this relation is particularly suitable to approximate the entire Pareto front in many-objective problems.

As we saw in the previous section, the proposed preference relation can be used to concentrate the search on a small subset of the entire Pareto front. However, in some situations we want to discover the whole Pareto front in order to learn about the problem, i.e., to understand the structure of the possible set of solutions, the degree of conflict and the trade-offs among the

objectives. In order to approximate the entire Pareto front, we suggest to use the central-guided Chebyshev relation to find the entire Pareto front. We could start the search using a small value of τ so that the population could converge quickly to a reduced region in the central part of the Pareto front. This way, we would avoid the problem of having a large proportion of Pareto-nondominated solutions during the first generations of the search. Once a Pareto optimal solution has been found, we could use a large value of τ to distribute the solutions along the entire Pareto front since all the solutions in the Pareto front would be equally ranked. However, in general, it is difficult to determine if the population has reached the Pareto optimal front or, at least, if there is no progress towards the Pareto front [84, 118]. Instead of introducing a new mechanism to determine if the MOEA should be stopped, we adopted an scheduling scheme to define the value of τ. That is, at the beginning of the search a small value of τ is adopted, and such value is gradually increased so that, at the end of the search it reaches a value of $\tau = 1$ (i.e., the usual Pareto dominance relation) in order to find solutions over the whole Pareto front. A gradual increment of the value of τ may help to deal with situations in which the MOEA reaches local Pareto optimal fronts. There are many possibilities to update the value of τ at each generation t. In this thesis we propose increasing τ according to the function:

$$\tau_t = \left(\frac{t}{g_{max}}\right)^p, \tag{8.6}$$

where t is the current generation, g_{max} is the total number of iterations, and $p > 0$ is a constant that determines how fast the value of τ is increased. Figure 67 shows some possible values for p. In some way, p controls the proportion of the search using a small τ. For example, if we use a large value of p, then during most of the search a small value of τ will be adopted. On the other hand, with a small value of p, a small τ will be used only during the few initial generations. Additionally, in both of these cases, the value of τ will change from a small value to 1 in only a few generations.

In order to deal with a wide range of MOPs, we recommend a value of $p = 2$.

The Chebyshev preference relation has some advantages over other relations proposed to deal with many-objective optimization problems. All the relations presented in Chapter 4 only emphasize a region of the Pareto front. However, the location and size of that region is fixed by the nature of the relation. Furthermore, as pointed out in [77], in some cases this region is far from the central part of the Pareto front. In contrast, the central-guided Chebyshev relation naturally converges towards the central part of the Pareto front, and the size of the emphasized region can be easily modified during the search.

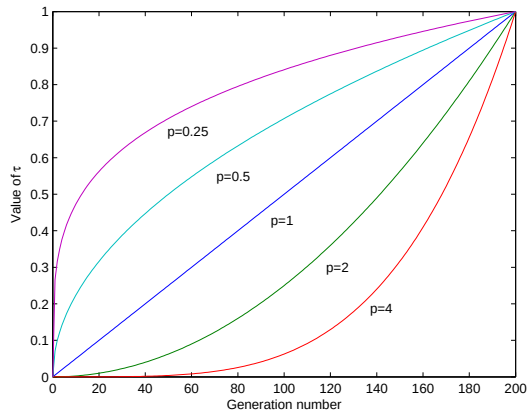


Figure 67: Growth of the normalized threshold, τ, with respect to the current generation.

8.6 Evaluation of the Chebyshev Relation for Approximating the Entire Pareto Front

In the next results we employed the central-guided Chebyshev relation with a value $p = 2$ for updating the value of τ.

Some studies [67, 61, 117, 92] have shown that NSGA-II and SPEA2 present scalability problems when the number of objectives is increased. For this reason, we selected these algorithms in order to investigate if the Chebyshev relation improves their convergence. Furthermore, we can show that the Chebyshev preference relation can be implemented in MOEAs with an archive or without it. Therefore, the comparative study includes four MOEAs, namely, NSGA-II, SPEA2 and their respective counterparts using the Chebyshev preference relation.

In the case of NSGA-II we used the following standard parameter values: a crossover probability of 0.9, a mutation probability of $1/n$ (n is the number of decision variables), and the distribution indices for crossover and mutation were set as 15 and 20, respectively.

For SPEA2, we adopted the following standard parameter values: an individual crossover probability of 1, an individual mutation probability of 1, a variable crossover probability of 0.5, a mutation probability of $1/n$, and a variable swap probability of 0.5.

In both algorithms we used a population of 200 individuals, and a total number of generations of 200. For all the configurations we carried out 30 runs for each MOEA. The results presented were averaged over the total of this number of runs.

8.6.1 *Test Problems Employed*

First, we used the DTLZ5(I, M) degenerate problem in order to show how, in spite of having a low dimensional Pareto front, some Pareto-based MOEAs have difficulties to achieve a good approximation of the Pareto front. Additionally, if we use $I = \{2, 3\}$ for DTLZ5(I, M), it is possible to approximate the Pareto front with a reasonable number of points. In a similar way, it is also possible to visualize the approximation set using a 2D or a 3D plot. In the next set of experiments we used $M = \{3, 4, 6, 8, 10, 12\}$ objectives and we adopted a fixed value of $I = 3$.

We adopted four scalable problems with different characteristics to assess the Chebyshev relation. These test problems include some having nonconvex or disconnected Pareto fronts. Three problems are taken from the DTLZ test suite [34]. Problems DTLZ1 and DTLZ2 were selected for having a nonconvex Pareto front and an easy way to compute the generational distance without using a discretization of the Pareto optimal front. DTLZ7 was selected to test the behavior of the Chebyshev relation when it is not clear how to determine the central point of the Pareto front. The last problem was selected from the WFG test suite [59]. We chose problem WFG6 because it has a symmetric Pareto front which allows us to easily measure some quality indicators even with a high number of objectives. Like most of the WFG problems, WFG6 is a nonseparable one, which makes it harder than the DTLZ problems considered in this work. Additionally, we thought that it would be a good opportunity to test the WFG test suite in many-objective problems, since most of the works found in the literature [67, 73, 94, 117, 36, 92] are focused on the DTLZ problems. The incorporation of new test suites could lead us to better understand the sources of difficulty of some many-objective problems.

In all MOPs, we employed a similar configuration, namely, $k - 1$ position-related variables. In order to maintain the test problem's complexity for every number of objectives, we fixed the number of distance-related variables to 5 for DTLZ1, and for the other test problems to 20. In our experiments we used 3, 4, 6, 8, 10 and 12 objectives in each test problem.

8.6.2 *Quality Indicators Employed*

In some many-objective problems it is not possible to generate a discrete representation of the Pareto optimal front. Therefore, we need to resort to indicators in which knowing the Pareto optimal front is not necessary.

In order to evaluate the convergence achieved by the MOEAs we used the generational distance (GD), or a similar indicator. We took advantage of the geometry of DTLZ1 and DTLZ2 to compute the generational distance. The nondominated vectors of DTLZ1 and DTLZ2's Pareto front have the

property $\sum_{i=1}^{k} z_i = 0.5$ and $\sum_{i=1}^{k} z_i^2 = 1$, respectively. This way, we can compute the generational distance using $GD = ((\sum_{i=1}^{k} z_i)/|P|) - 0.5$ and $GD = ((\sum_{i=1}^{k} z_i^2)/|P|) - 1$, respectively. In DTLZ7, we used the value of the auxiliary function $g(\mathbf{x})$. This function is used to evaluate DTLZ7's objective functions. The Pareto optimal front of DTLZ7 is achieved when $g(\mathbf{x}) = 0$. Thus, we compute the value of this function as a way to measure the convergence in DTLZ7.

In order to directly compare the performance of the MOEAs in all the test problems, we used a binary indicator, namely, the additive ϵ-indicator [130]. This indicator is defined as

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{\forall \mathbf{z}^2 \in B \ \exists \mathbf{z}^1 \in A : \mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2\}$$

for two nondominated sets A and B, where $\mathbf{z}^1 \preceq_{\epsilon+} \mathbf{z}^2$ iff $\forall i : z_i^1 \leqslant \epsilon + z_i^2$, for a given ϵ. In other words, $I_{\epsilon+}(A, B)$ is the minimum value such that aggregated to any objective vector in B, then $A \preceq B$. In general, $I_{\epsilon+}(A, B) \neq I_{\epsilon+}(B, A)$, so we have to compute both values. The smaller $I_{\epsilon+}(A, B)$ and larger $I_{\epsilon+}(B, A)$, the better A over B.

Finally, to assess both convergence and diversity, we adopt the hypervolume indicator. As recommended in [117], for DTLZ5(I, M), DTLZ2 and WFG6, the reference point to compute the hypervolume was $\mathbf{z}^{hyp} = 1.1^k$, whereas for DTLZ1, $\mathbf{z}^{hyp} = 0.7^k$. The results presented are normalized with respect to the hypervolume achieved by the Pareto optimal set.

8.6.3 *Assessing Convergence*

The plot shown in Figure 69 presents the results of the generational distance (GD) when varying the number of objectives. Judging by the values, it is evident that, on the one hand, the convergence of the original MOEAs is considerably degraded when the number of objectives is high. On the other hand, the MOEAs using the central-guided Chebyshev relation yield a good convergence in terms of GD even when the number of objectives is increased.

From the hypervolume results presented in Figure 68 one can confirm that Pareto-based MOEAs fail to converge, whereas their Chebyshev counterparts are less affected by the number of objectives. Although the hypervolume values obtained by the Chebyshev-based MOEAs slowly decreases with the number of objectives, we can still say that the resulting PF_{approx} is well-distributed along the Pareto front.

The results of the ϵ-indicator are presented in the matrices of the subplots of Figs. 70, 81 and 82. We can interpret these results as follows. $I_{\epsilon+}(A, B)$ is the subplot located in row A, and column B of the matrix. The boxes in each subplot depict the results for each number of objectives. As we can see, the

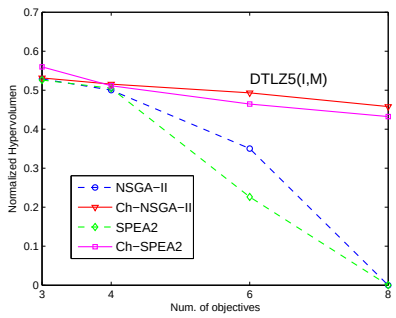


Figure 68: Hypervolume results for problem DTLZ5(I, M) with different numbers of objectives. The results are averaged over 30 runs.

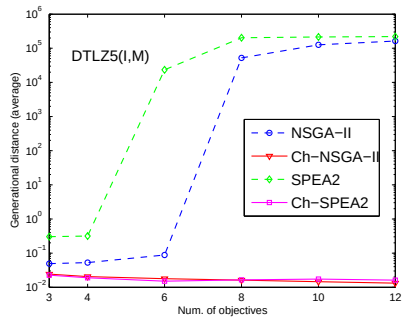


Figure 69: Generational distance for problem DTLZ5(I, M) with different numbers of objectives. The results are averaged over 30 runs.

results of the ϵ-indicator confirm that the original MOEAs are outperformed by their counterparts using the Chebyshev preference relation. In particular, the PF_{approx} sets obtained by NSGA-II and SPEA2 are practically dominated by the Chebyshev-based MOEAs since the values $I_{\epsilon+}(\text{Ch-MOEA}, \text{MOEA})$ are almost zero. Contrarily, a large ϵ value needs to be added to the PF_{approx} sets of the Chebyshev-based MOEAs in such a way that they can be dominated by the PF_{approx} sets of the Pareto-based MOEAs.

Figs. 71 and 72 present the results of the online GD (i.e., the GD values obtained at each generation) in problem DTLZ1 with 3 and 6 objectives. As we can see, even with 3 objectives, each MOEA using the Chebyshev relation achieves a better GD with respect to its original counterpart. This difference in GD is remarkably stressed for 6 objectives (compare Figs. 71 and 72). It is worth noting that the convergence of the MOEAs with the Chebyshev relation is maintained almost unaltered when the number of objectives is increased from 3 to 6. The progress of the generational distance when the number of objectives is increased can be better appreciated in Figure 73. From this plot it is clear to see that the convergence ability of the original MOEAs decreases quickly. One important fact to note in Figure 72 is that, starting from a certain generation number, SPEA2's GD value diverges when the number of generations increases. This behavior was also observed for 8, 10 and 12 objectives. For those numbers of objectives, NSGA-II presented the same behavior. In the next section we will explain a possible reason for this fact. In problem DTLZ2 we can appreciate a similar situation, although with a high number of objectives it is possible to note a degradation in the convergence ability of the Chebyshev-based MOEAs (see Figs. 75 and 76). In terms of the

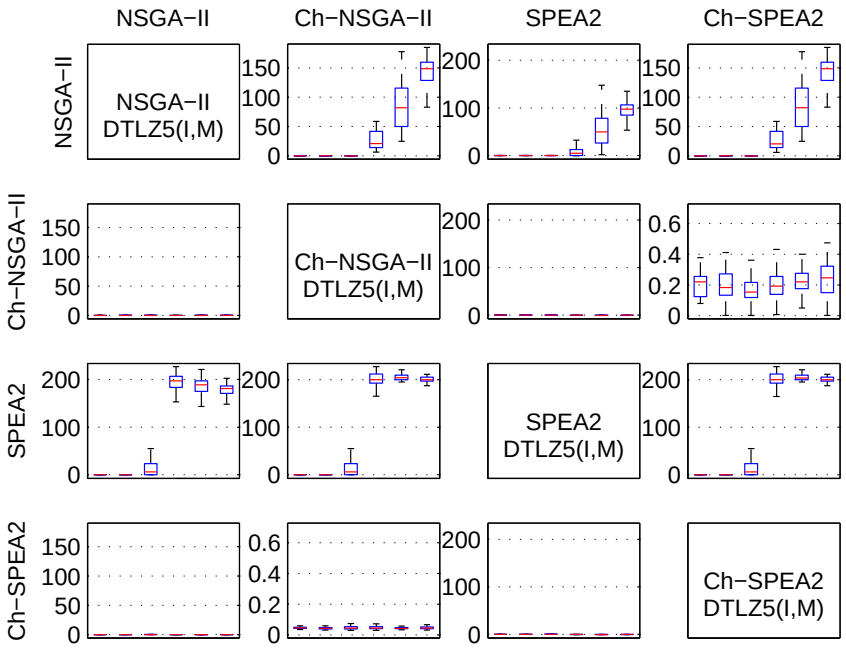


Figure 70: Results of the ϵ-indicator for problem DTLZ5(I, M). Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively.

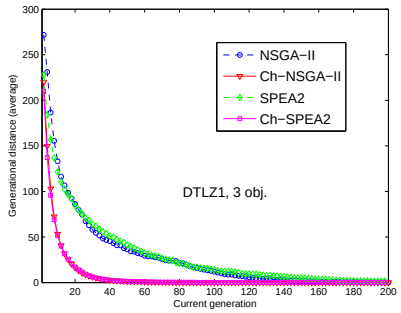


Figure 71: Online generational distance for problem DTLZ1 using 3 objectives. The results are averaged over 30 runs.

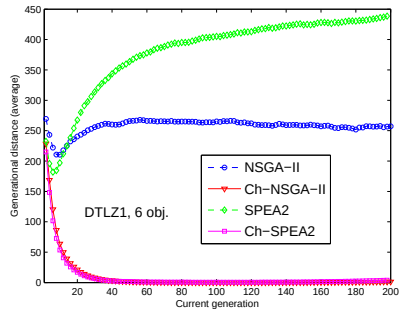


Figure 72: Online generational distance for problem DTLZ1 using 6 objectives. The results are averaged over 30 runs.

hypervolume obtained, we can also conclude that the performance observed by the Chebyshev-based MOEAs is not as good as the one noted in DTLZ1.

Regarding DTLZ7, Chebyshev-based MOEAs also achieve an important convergence scalability in terms of the generational distance. However, in this problem the Chebyshev-based SPEA2 achieved a better convergence than the Chebyshev-based NSGA-II since, as can be seen in Figure 79, the value of $g(\mathbf{x})$ achieved by SPEA2 is very close to zero, the optimal value.

The results of the ϵ-indicator are presented in the matrices of the subplots of Figs. 80, 81 and 82. As we can see, the results of the ϵ-indicator agree with those of the generational distance. That is, each MOEA is outperformed by the version that uses the Chebyshev preference relation. In addition, we can see that the performance of the Pareto-based MOEAs decreases with the number of objectives. That is, the values for each subplot $I_{\epsilon+}(\text{MOEA}, \text{Ch-MOEA})$ tend to increase from 3 to 12 objectives. In DTLZ1, DTLZ2 and DTLZ7, the difference in favor of the Chebyshev MOEAs is large. For instance, in DTLZ7, the values $I_{\epsilon+}(\text{Ch-MOEA}, \text{MOEA})$ (see Figure 82) are, for any number of objectives, close to zero. In contrast, many of the worst values $I_{\epsilon+}(\text{MOEA}, \text{Tch-MOEA})$ are above 10. Like with GD, with respect to the ϵ-indicator the Chebyshev-based NSGA-II outperformed the Chebyshev-based SPEA2 in DTLZ2, but in DTLZ7 the opposite result is obtained.

8.6.4 *Dominance Resistant Solutions in DTLZ Problems*

By analyzing some plots and performance indicator results we hinted that the divergence problems of the Pareto-based MOEAs when the number of generations increases was due to the so-called Dominance Resistant

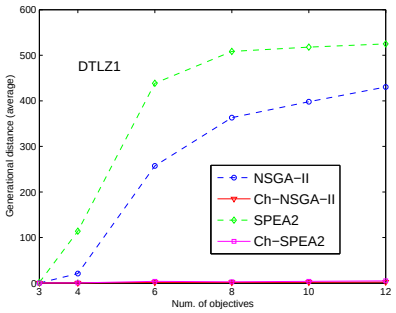


Figure 73: Generational distance for problem DTLZ1 with different numbers of objectives. The results are averaged over 30 runs.

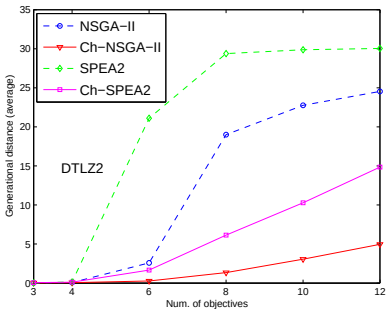


Figure 74: Generational distance for problem DTLZ2 with different numbers of objectives. The results are averaged over 30 runs.

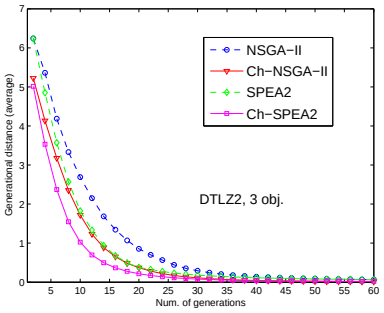


Figure 75: Online generational distance for problem DTLZ2 using 3 objectives. The results are averaged over 30 runs.

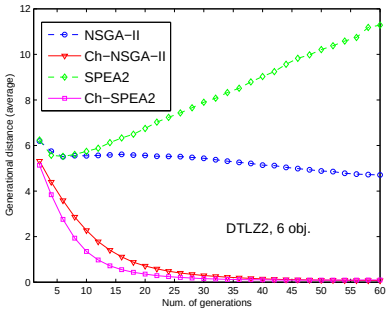


Figure 76: Online generational distance for problem DTLZ2 using 6 objectives. The results are averaged over 30 runs.

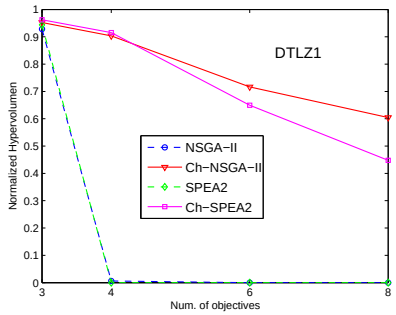


Figure 77: Hypervolume results for problem DTLZ1 with different numbers of objectives. The results are averaged over 30 runs.

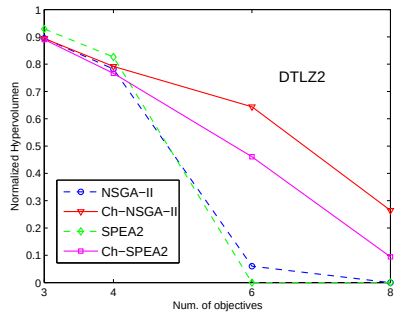


Figure 78: Hypervolume results for problem DTLZ2 with different numbers of objectives. The results are averaged over 30 runs.

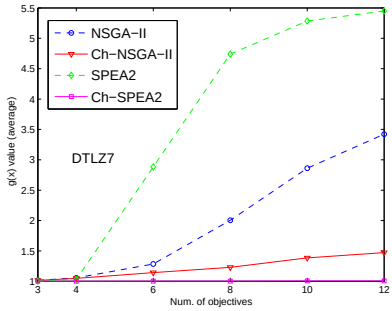


Figure 79: $g(\mathbf{x})$ values for problem DTLZ7 with different numbers of objectives. The results are averaged over 30 runs.

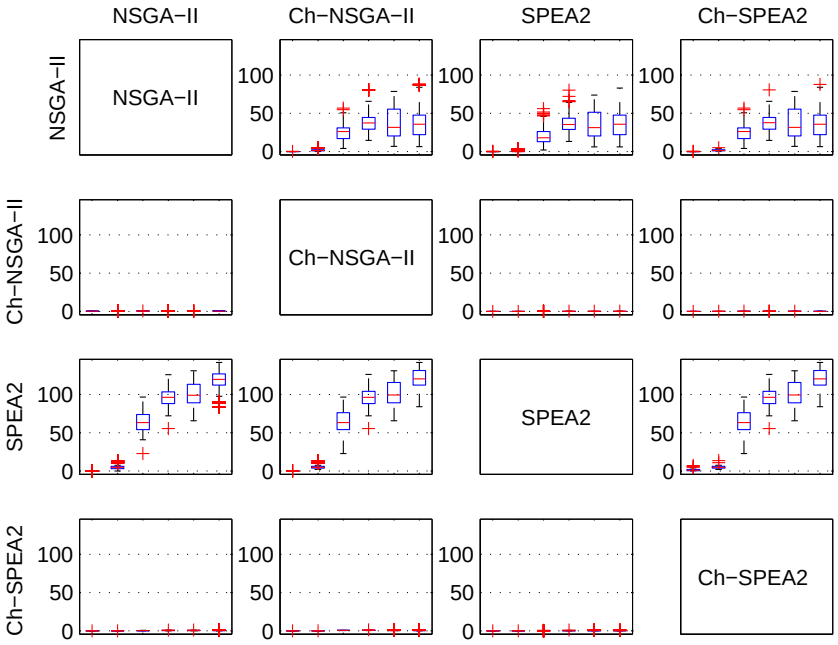


Figure 80: Results of the ϵ-indicator for problem DTLZ1. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively.

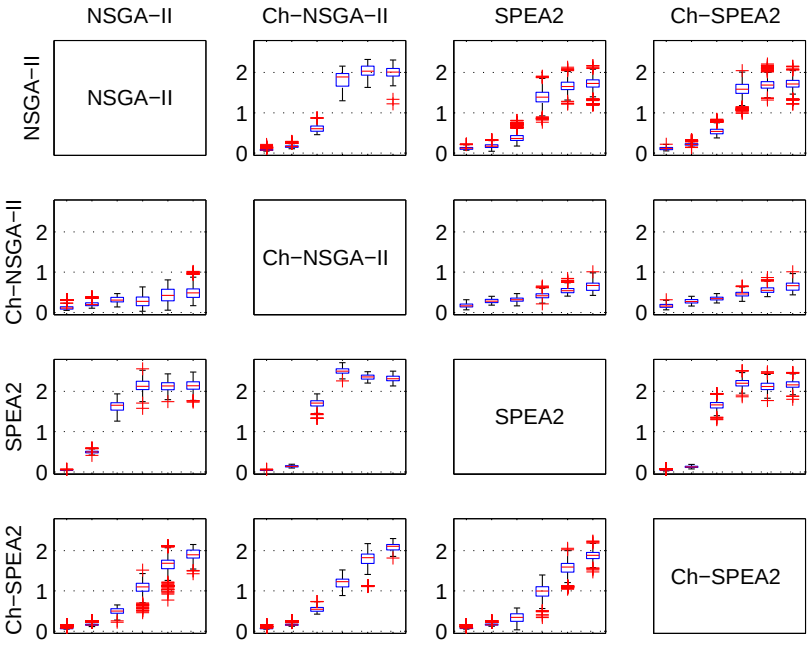


Figure 81: Results of the ϵ-indicator for problem DTLZ2. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively.

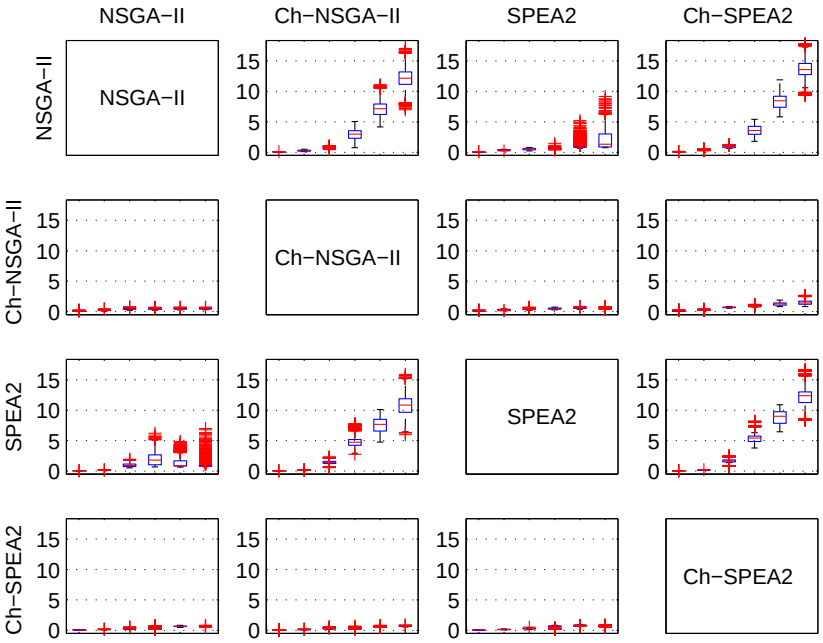


Figure 82: Results of the ϵ-indicator for problem DTLZ7. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively.

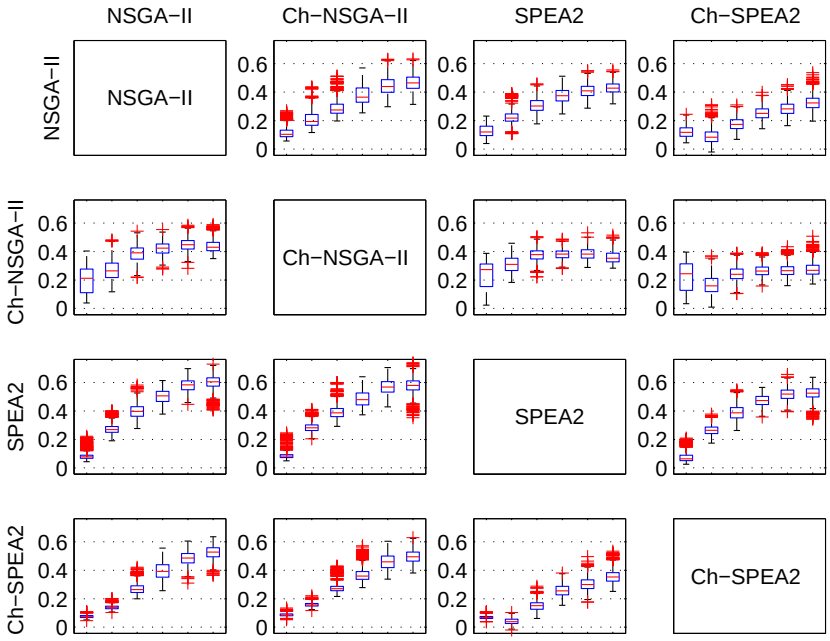


Figure 83: Results of the ϵ-indicator for problem WFG6. Each subplot presents the values for 3,4,6,8,10 and 12 objectives, respectively.

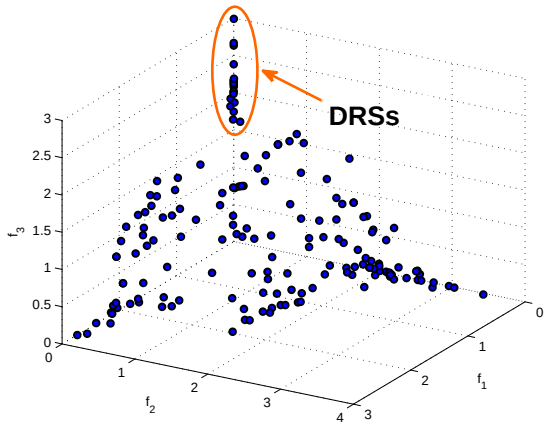


Figure 84: Illustration of some dominance resistant solutions (DRSs) in problem DTLZ2.

Solutionss (DRSs). Dominance resistant solutions are those with a poor value in at least one of the objectives, but with near optimal values in the others. Figure 84 shows an example of DRSs solutions in DTLZ2. Although the pointed group of solutions in the figure have poor values in objective f_3, they are nondominated solutions because they have values close to zero in the objectives f_1 and f_2. Dominance resistant solutions were first noted by Ikeda et al. [71], and Hanne [57]. These authors suggested that DRSs may easily appear in problems with many objectives. Deb et al. [34] and Huband et al. [59] also provided a similar assumption.

Next, we present experimental evidence that shows that DRSs are the culprit of the scalability problems in the DTLZ test problems considered in this chapter. In the DTLZ test problems, DRSs are located nearby each of the axes of the objective space (as shown in Figure 84). The reason for this is the shape of the feasible space in the DTLZ problems. The left plot in Figure 85 shows the feasible space of DTLZ2 with two objectives.

In order to avoid DRSs we suggest to introduce a slight slope in the edges of the feasible space. That is, we want a feasible space like the one presented in the plot on the right-hand side of Figure 85. To do so, we only have to add a second term to each objective of DTLZ1, DTLZ2 and DTLZ7. In the case

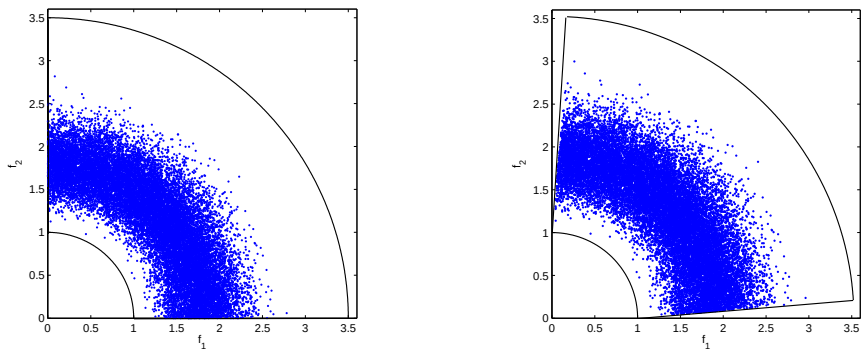


Figure 85: Feasible objective function space of DTLZ2 (left) and the extended DTLZ2 (right) and 20 000 solutions generated at random. Both problems have the same Pareto optimal front. However, the extended version avoids dominance resistant solutions.

of DTLZ2 we obtain the extended $\text{DTLZ2}_{\text{ext}}$ problem defined in Eq. 8.7. It is possible to introduce a similar extension to DTLZ1 and DTLZ7.

$$
\begin{aligned}
\text{Min} \quad & \mathbf{f} = [f_1, \ldots, f_k], \text{ where} \\
f_1(\mathbf{x}) \quad & = \rho \cdot g(\mathbf{x}_M) + [1 + g(\mathbf{x}_M)] \cos(\theta_1) \ldots \\
& \quad \cos(\theta_{M-2}) \cos(\theta_{M-1}), \\
f_2(\mathbf{x}) \quad & = \rho \cdot g(\mathbf{x}_M) + [1 + g(\mathbf{x}_M] \cos(\theta_1) \ldots \\
& \quad \cos(\theta_{M-2}) \sin(\theta_{M-1}), \\
\vdots \quad & \vdots \\
f_{k-1}(\mathbf{x}) \quad & = \rho \cdot g(\mathbf{x}_M) + [1 + g(\mathbf{x}_M)] \cos(\theta_1) \sin(\theta_2), \\
f_k(\mathbf{x}) \quad & = \rho \cdot g(\mathbf{x}_M) + [1 + g(\mathbf{x}_M)] \sin(\theta_1) \\
\text{where} \quad & g(\mathbf{x}_M) = \textstyle\sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2, \\
& \theta_i = x_i \pi/2, \\
& \rho \text{ is a small real number, and} \\
& 0 \leqslant x_i \leqslant 1, \quad \text{for } i = 1, 2, \ldots, n.
\end{aligned}
\tag{8.7}
$$

Figs. 86 and 87 show the online GD values for NSGA-II and SPEA2 using different numbers of objectives in problems $\text{DTLZ1}_{\text{ext}}$ and $\text{DTLZ2}_{\text{ext}}$, respectively.

As those figures clearly show, after a few generations, for every number of objectives both MOEAs were able to approximate very accurately the true Pareto front. We can only note a slight deterioration in convergence when the number of objectives is increased. We believe that this deterioration

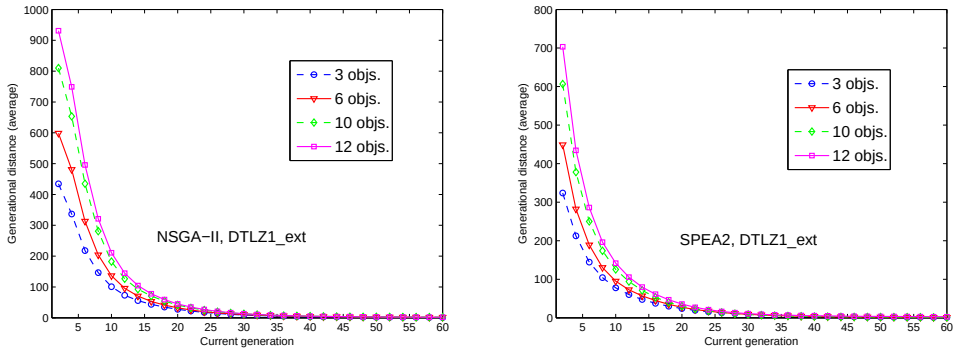


Figure 86: Online generational distance for NSGA-II and SPEA2 using the extended version of **DTLZ1**. The extended version removes dominance resistant solutions.

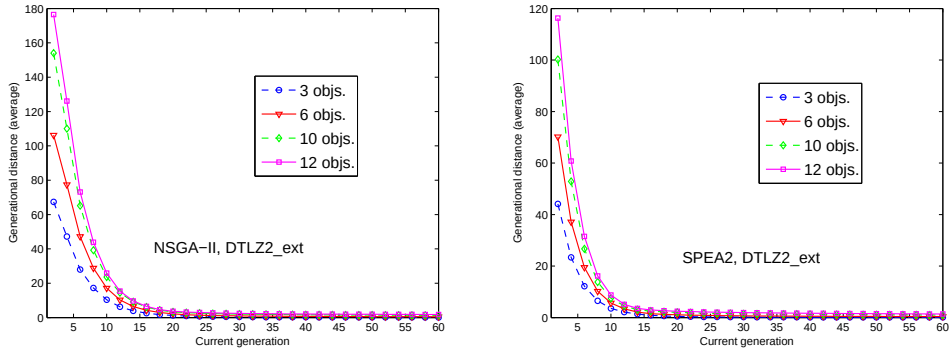


Figure 87: Online generational distance for NSGA-II and SPEA2 using the extended version of **DTLZ2**. The extended version removes dominance resistant solutions.

is caused by the large proportion of nondominated solutions in the first population. Since the extended DTLZ problems do not promote DRSs, we can conclude that the main source of difficulty when scaling the number of objectives of the DTLZ test problems is the presence of dominance resistant solutions. Other DTLZ test problems not included in this chapter have a similar feasible search space to that of DTLZ1 or DTLZ2. Therefore, we can expect that other DTLZ test problems will also have DRSs.

NSGA-II and SPEA2 are more sensitive to DRSs because of their internal mechanism to preserve extreme solutions in the populations. That is, the crowding operator of NSGA-II always keeps the best solutions in each of the objectives. In turn, SPEA2's archive truncation method guarantees the preservation of boundary solutions. Although in most of the MOPs these mechanisms represent a benefit, in some other problems, like those of the DTLZ test suite, this mechanism may promote DRSs and hinder converge towards the Pareto optimal front. In fact, these solutions are copied as many times that at the end of the search, they have replaced other nondominated solutions closer to the Pareto front. This explains why the GD values of NSGA-II and SPEA2 get worse as the number of generations increases.

On the other hand, the Chebyshev relation compares most of the solutions using the achievement function. Thus, although DRSs are equally ranked by the Pareto relation, the Chebyshev relation ranks DRSs worse than other nondominated solutions located nearby the Pareto front. As a result, as it was shown in the experiments using the original DTLZ test problems, the Chebyshev relation can effectively discard dominance resistant solutions.

We have to note however, that, in general, DRSs might not be the main source of difficulty when the number of objective is increased.

8.6.5 *Evaluation using WFG6*

With respect to the problem WFG6, the Chebyshev-based MOEAs outperformed their Pareto-based counterparts only marginally for some numbers of objectives. For instance, the ϵ-indicator values (see Figure 83) $I_{\epsilon+}(\text{SPEA2}, \text{Ch-SPEA2})$ for 6,8,10 and 12 objectives are slightly greater than those of $I_{\epsilon+}(\text{Ch-SPEA2}, \text{SPEA2})$. In contrast, in terms of the hypervolume indicator, the results shown in Figure 88, indicate that the Chebyshev-based MOEAs clearly outperformed the original MOEAs. In addition, we can see that the performance with respect to the hypervolume does not degrade with an increasing number of objectives.

The problem WFG6 has a feasible objective space very similar to that of the extended DTLZ2 (see plot on the right-hand side of Figure 85). Therefore, it does not present DRSs like those observed in the DTLZ problems. Thus, it

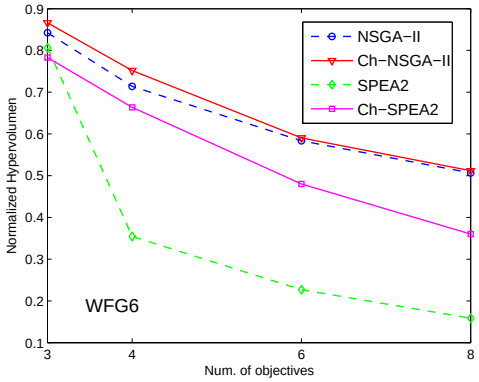


Figure 88: Hypervolume results for problem WFG6 with different numbers of objectives. The results are averaged over 30 runs.

seems that the difficulty of WFG6 is not related to the number of objectives. We believe that the main difficulty of some WFG problems is their nonseparability, which is the main difference that they have with respect to the DTLZ problems.[1] Additionally, although we used 20 distance-related variables in all the problems studied, to approach the Pareto optimal front in the WFG problems is harder than in the DTLZ problems, even with 2 or 3 objectives. In the following, we will show an experiment using only 2 objectives. Figure 89 shows the solutions found by the Pareto-based NSGA-II and the Chebyshev-based NSGA-II using a threshold of zero (i.e., we only look for the solution in the central part of the Pareto front). On the one hand, one might expect that generating only one solution would be much faster than generating a set of them. However, as the figure shows, the single solution generated using the Chebyshev relation is barely ahead of the solutions generated by the Pareto relation. This implies that WFG6 is difficult even with 2 objectives. For this reason we believe that the growth of the proportion of nondominated solutions when the number of objectives increases is not the main difficulty in problem WFG6, but its nonseparability. Therefore, we expect that MOEAs that carry out many single objective searches (e.g., [61]) will have problems with WFG6 when adopting many objectives.

In addition, we can also see in Figure 89, that even if the number of evaluations is considerably increased, the performance using both preference relations does not drastically improve the results obtained. This might help to explain why the improvement achieved by the Chebyshev-based MOEAs with respect to their Pareto-based counterparts is only marginal.

1 It is worth remembering that, as pointed out in [59], all the DTLZ problems are separable.

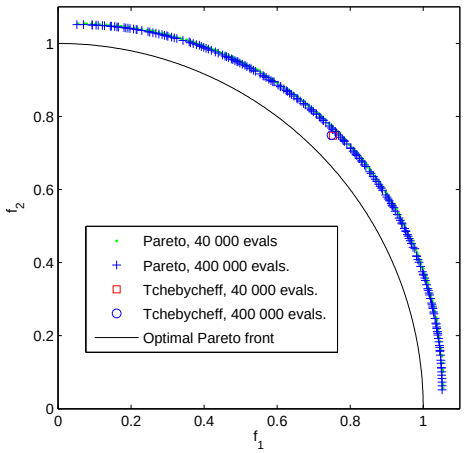


Figure 89: Approximation set obtained using the Pareto dominance and Chebyshev dominance relation (threshold of zero) with both 40 000 and 400 000 evaluations.

8.7 Final Remarks

In this chapter, we have proposed a new preference relation based on an achievement scalarizing function. The purpose of the new preference relation is twofold. On the one hand, it offers an easy approach to integrate decision maker's preferences into a MOEA without modifying the original structure of the MOEA. On the other hand, the proposed reference relation is intended to deal with many-objective problems.

The interactive optimization process proposed was evaluated using an airfoil shape design problem. From solving that problem we can see that setting the parameters of the method represents a low cognitive load.

The results regarding the use of the central-guided relation to approximate the entire Pareto front showed that the MOEAs used in this study considerably improved their convergence ability by using the proposed preference relation in most of the test problems considered. Besides outperforming their Pareto-based counterpart, the experiments revealed that the Chebyshev relation is less affected by the increment in the number of objectives. In a detailed analysis of these results we concluded that the main source of difficulty when increasing the number of objectives in the DTLZ problems considered in this chapter is the presence of dominance resistant solutions. We also showed that the Chebyshev relation was successful in discarding this type of solutions. However, in the test problem WFG6, the Chebyshev-based MOEAs marginally outperformed the Pareto-based versions with respect to the

ϵ-indicator. The experimental results suggest that the main difficulty of the WFG6 problem is not the increase of the number of objectives, but its nonseparability, which is present even in problems with 2 or 3 objectives. In this sense, we recommend using the DTLZ test suite to test the ability of a MOEA to deal with dominance resistant solutions (specially with a high number of objectives).

9 CONCLUSIONS AND FUTURE WORK

Contents

9.1 Conclusions 147
9.2 Future Work 150

9.1 Conclusions

In this thesis we have proposed several techniques to deal with problems with a large number of objectives (many-objective problems). Those techniques can be classified in two main classes: 1) reduction of the number of objectives of the problem during the search process or, a posteriori, and 2) use of new preference relation to generate a finer solution ordering. In the next two sections we will give a brief summary of the techniques that have been proposed in this thesis followed by the conclusions obtained by their experimental evaluation.

9.1.1 *Objective Reduction During the Search or After It*

1. Regarding this kind of approach we proposed two algorithms to reduce the number of objectives given a Pareto front approximation of a MOP. The underlying idea of these objective reduction algorithms is to identify the nonconflicting objectives (or nonconflicting to some degree) in order to discard them. Those algorithms can be directly used in the decision making process, i.e., once an approximation of the Pareto front has been found.

   From the experimental study of these two algorithms we conclude that, although some conflict may exists among all the objectives of a problem, there are some objectives with a small contribution to the structure of the optimal Pareto front. In other words, the removal of those objectives represents small changes in the final Pareto front in terms of some quality indicator. The experimental assessment of the objective reduction algorithms revealed that they successfully identified the nonredundant objectives. In addition, they have some advantages over

similar algorithms. First, our approaches have a lower time complexity. This characteristic is important if we want to integrate an objective reduction algorithm during the search since the efficiency of such integration depends on the complexity of the reduction technique. Second, the proposed reduction techniques showed a competitive performance with respect to two other similar proposals.

From a general point of view, the removal of those objectives can help to the problem designer or the decision maker to gain knowledge about the relation and importance of the objectives according to the conflict. With regard to the decision making process, the removal of the nonconflicting objectives eases the visualization of the approximation of the Pareto front. In cases with a moderate number of objectives (i.e., 4 to 7), the reduced objective set might be visualized using traditional 3D plots.

2. Later on, we incorporated one of those algorithms into a MOEA in order to improve its search ability to approximate the entire Pareto front. We designed two schemes in which the objective reduction algorithms are used following different approaches. In one of the schemes the objectives are gradually reduced, and at the end of the search all the objectives are used again in order to cover the entire Pareto front.

   From this proposal we can conclude that *i*) the overhead of the objective reduction algorithms is less than the computational savings by omitting some objectives, and *ii*) that the online objective reduction helps to remedy the limitation of Pareto optimality for dealing with many-objective problems. The computational savings represent an important advantage for solving real-world problems with expensive objective functions. Additionally, the use of a small set of objectives during the search makes possible the adoption of expensive ranking schemes (e.g., those based on the hypervolume indicator) in problems with a high number of objectives.

   In an effort to minimize the loss of information by removing some objectives, we proposed a second scheme in which the objectives are not removed, but the set of objectives is partitioned to form subspaces according to their degree of conflict. Thus, this scheme allows that all the objectives are used during the search, decreasing this way, the loss of information. We experimented with different partition strategies, and we conclude that the partitions based on conflict information achieve better results than those by using other partition strategies. Another important finding is that is more important to keep small subspaces although objectives in highly conflict are placed in different subspaces.

9.1.2 *New preference relations*

In this approach we first carried out a comparative study of different preference relations that have been used to cope with many-objective problems. The main goal of that study was to serve as a starting point to propose a new preference relation.

The comparative study of the preference relations revealed that, in spite of the fact that some preference relations contribute to converge faster to the Pareto front than the Pareto dominance relation, they also stress the generation of solutions far from the knee region (i.e., the middle region of the Pareto front). This fact limits the applicability of these relations since, in the general case, it is commonly assumed that the decision maker prefers solutions on the knee region.

According to our experience, a good preference relation to deal with many objective problems should have the following properties:

1. It should be compatible with the Pareto dominance relation. That is, if a given solution is better than other in terms of the Pareto dominance relation, it must also be better using the alternative preference relation.

2. Its complexity should not be exponential with respect to the number of objectives.

3. It should allow the user to determine the size and location of the region of interest.

4. The number of levels in which a set of solutions is ranked should remains constant as much as possible when the number of objectives is increased.

Based on the discoveries of that study we designed a new preference relation to compare solutions in objective space. Besides the ability of comparing nondominated solutions, the proposed preference relation was designed to incorporate preferences.

The new preference relation divides the objective function space into two subspaces. The solutions of one of these subspaces are compared adopting the usual Pareto dominance relation, while the other is compared using the achievement function. Besides finding the optimal solution of the achievement function, the new preference relation allows us to find a set of solutions around such an optimal solution. Additionally, the size and range of that set can be regulated by the DM. In order to incorporate preferences into a MOEA, we only need to change the Pareto dominance checking functions by the new preference relation.

The preference relation was first incorporated into an interactive process, in which DM's preferences are incorporated iteratively during the search process. Then, the preference relation was integrated into a MOEA but in an *a priori* fashion.

The proposed interactive optimization process was evaluated using an airfoil shape design problem. From solving that problem we can see that setting the parameters of the method represents a low cognitive load for the DM.

The results regarding the use of the *a priori* approach to approximate the entire Pareto front showed that the MOEAs used in this study considerably improved their convergence ability by using the proposed preference relation in most of the test problems considered. Besides outperforming their Pareto-based counterpart, the experiments revealed that the Chebyshev relation is less affected by the increment in the number of objectives. In a detailed analysis of these results we concluded that the main source of difficulty when increasing the number of objectives in the DTLZ problems considered in this thesis is the presence of dominance resistant solutions. We also showed that the Chebyshev relation was successful in discarding this type of solutions.

9.2 Future Work

Currently, both objective reduction algorithms use a linear similarity measure (i.e., correlation coefficient) to estimate the conflict among objectives. However it would be interesting to try a non-linear similarity measure (e.g., kernel methods or correntropy [75]) to investigate if the effectiveness of the techniques is improved in a significant way.

In all the proposed schemes that use conflict information during the search there is an important element that could improve the performance of a MOEA. That element is the mechanism to populate the Pareto front when only some portion of the Pareto has been discovered. In the current implementation of our proposed schemes we only use all the objectives in order to fill the gaps of the Pareto front. However, it would be interesting to experiment with specialized methods to reconstruct the Pareto front from some locally nondominated points. For instance, the method could be hybridized with a local search method.

Regarding the Chebyshev preference relation, one possible path of future research is to add to it the ability to guide the search towards several reference points given by the decision maker. That way, the decision maker could simultaneously explore several regions of interest.

BIBLIOGRAPHY

[1] Per J. Agrell. On redundancy in multi criteria decision making. *European Journal of Operational Research*, 98(3):571–586, 1997.

[2] Hernán E. Aguirre and Kiyoshi Tanaka. Many-Objective Optimization by Space Partitioning and Adaptive e-Ranking on MNK-Landscapes. In Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *EMO 2009*, pages 407–422. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009.

[3] Maria Joao Alves and Marla Almeida. MOTGA: A multiobjective Tchebycheff based genetic algorithm for the multidimensional knapsack problem. *Computers & Operations Research*, 34(11):3458–3470, November 2007.

[4] D.V. Arnold. *Noisy optimization with evolution strategies*. Kluwer Academic Publishers, 2002. ISBN 1402071051.

[5] Julio Barrera, Juan J. Flores, and Claudio Fuerte-Esquivel. Generating complete bifurcation diagrams using a dynamic environment particle swarm optimization algorithm. *Journal of Artificial Evolution and Applications*, 2008:7:1–7:8, January 2008. ISSN 1687-6229.

[6] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson. On the average number of maxima in a set of vectors and applications. *J. ACM*, 25(4):536–543, 1978. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/322092.322095.

[7] P. J. Bentley and J. P. Wakefield. Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms. In P. K. Chawdhry, R. Roy, and R. K. Pant, editors, *Soft Computing in Engineering Design and Manufacturing*, Part 5, pages 231–240, London, June 1997. Springer Verlag London Limited. (Presented at the 2nd On-line World Conference on Soft Computing in Design and Manufacturing (WSC2)).

[8] Jürgen Branke, Thomas Kaußler, and Hartmut Schmeck. Guiding Multi Objective Evolutionary Algorithms Towards Interesting Regions.

Technical Report 398, Institute für Angewandte Informatik und Formale Beschreibungsverfahren, Universität Karlsruhe, Karlsruhe, Germany, February 2000.

[9] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding Knees in Multi-Objective Optimization. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 722–731, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[10] Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors. *Multiobjective Optimization. Interactive and Evolutionary Approaches*. Springer. Lecture Notes in Computer Science Vol. 5252, Berlin, Germany, 2008.

[11] Jan Braun, Johannes Krettek, Frank Hoffmann, and Torsten Bertram. Multi-Objective Optimization with Controlled Model Assisted Evolution Strategies. *Evolutionary Computation*, 17(4):577–593, Winter 2009.

[12] Dimo Brockhoff and Eckart Zitzler. Are All Objectives Necessary? On Dimensionality Reduction in Evolutionary Multiobjective Optimization. In Thomas Philip Runarsson, Hans-Georg Beyer, Edmund Burke, Juan J. Merelo-Guervós, L. Darrell Whitley, and Xin Yao, editors, *Parallel Problem Solving from Nature - PPSN IX, 9th International Conference*, pages 533–542. Springer. Lecture Notes in Computer Science Vol. 4193, Reykjavik, Iceland, September 2006.

[13] Dimo Brockhoff and Eckart Zitzler. Offline and Online Objective Reduction in Evolutionary Multiobjective Optimization Based on Objective Conflicts. TIK Report 269, Institut für Technische Informatik und Kommunikationsnetze, ETH Zürich, April 2007.

[14] Dimo Brockhoff and Eckart Zitzler. Improving Hypervolume-based Multiobjective Evolutionary Algorithms by Using Objective Reduction Methods. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 2086–2093, Singapore, September 2007. IEEE Press.

[15] C. Carlsson and R. Fullér. Interdependence in fuzzy multiple objective programming. *Fuzzy Sets and Systems*, 65(1):19–30, 1994.

[16] C. Carlsson and R. Fullér. Multiple criteria decision making: The case for interdependence. *Computers and Operations Research*, 22(3):251–260, 1995.

[17] A. Charnes and W. W. Cooper. *Management Models and Industrial Applications of Linear Programming*, volume 1. John Wiley, New York, 1961.

[18] Kazuhisa Chiba, Akira Oyama, Shigeru Obayashi, Kazuhiro Nakahashi, and Hiroyuki Morino. Multidisciplinary design optimization and data mining for transonic regional-jet wing. *AIAA Journal of Aircraft*, 44(4):1100–1112, July–August 2007. DOI: 10.2514/1.17549.

[19] Carlos A. Coello Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. An International Journal*, 1(3):269–308, Agosto 1999.

[20] Carlos A. Coello Coello and Gary B. Lamont, editors. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, 2004. ISBN 981-256-106-4.

[21] Carlos A. Coello Coello and Gregorio Toscano Pulido. Multiobjective Optimization using a Micro-Genetic Algorithm. In Lee Spector, Erik D. Goodman, Annie Wu, W.B. Langdon, Hans-Michael Voigt, Mitsuo Gen, Sandip Sen, Marco Dorigo, Shahram Pezeshk, Max H. Garzon, and Edmund Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.

[22] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.

[23] David Corne and Joshua Knowles. Techniques for Highly Multiobjective Optimisation: Some Nondominated Points are Better than Others. In Dirk Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 773–780, London, UK, July 2007. ACM Press.

[24] Dragan Cvetković and Ian C. Parmee. Preferences and their Application in Evolutionary Multiobjective Optimisation. *IEEE Transactions on Evolutionary Computation*, 6(1):42–57, February 2002.

[25] Dragan Cvetković and Ian C. Parmee. Use of Preferences for GA–based Multi–objective Optimisation. In Wolfgang Banzhaf, Jason Daida, Agoston E. Eiben, Max H. Garzon, Vasant Honavar, Mark Jakiela, and Robert E. Smith, editors, *GECCO–99: Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1504–1509, Orlando, Florida, USA, 1999. Morgan Kaufmann Publishers.

[26] Indraneel Das. On characterizing the "knee" of the Pareto curve based on Normal-Boundary Intersection. *Structural Optimization*, 18(2–3): 107–115, October 1999.

[27] Indraneel Das. A preference ordering among various Pareto optimal alternatives. *Structural and Multidisciplinary Optimization*, 18(1):30–35, August 1999. doi: 10.1007/BF01210689.

[28] Kalyanmoy Deb. Evolutionary Algorithms for Multi-Criterion Optimization in Engineering Design. In Kaisa Miettinen, Marko M. Mäkelä, Pekka Neittaanmäki, and Jacques Periaux, editors, *Evolutionary Algorithms in Engineering and Computer Science*, chapter 8, pages 135–161. John Wiley & Sons, Ltd, Chichester, Reino Unido, 1999.

[29] Kalyanmoy Deb. Solving Goal Programming Problems Using Multi-Objective Genetic Algorithms. In *1999 Congress on Evolutionary Computation*, pages 77–84, Washington, D.C., July 1999. IEEE Service Center.

[30] Kalyanmoy Deb and Abhay Kumar. Light Beam Search Based Multi-objective Optimization using Evolutionary Algorithms. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 2125–2132, Singapore, September 2007. IEEE Press.

[31] Kalyanmoy Deb and Dhish Kumar Saxena. Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 3353–3360, Vancouver, BC, Canada, July 2006. IEEE.

[32] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature VI*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[33] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[34] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Multi-Objective Optimization Test Problems. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 825–830, Piscataway, New Jersey, May 2002. IEEE Service Center.

[35] Kalyanmoy Deb, Shamik Chaudhuri, and Kaisa Miettinen. Towards Estimating Nadir Objective Vector Using Evolutionary Approaches. In Maarten Keijzer, editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 643–650, Seattle, Washington, USA, July 2006. ACM Press.

[36] Francesco di Pierro, Shoon-Thiam Khu, and Dragan A. Savić. An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. *IEEE Transactions on Evolutionary Computation*, 11(1):17–45, February 2007.

[37] Nicole Drechsler, Rolf Drechsler, and Bernd Becker. Multi-Objected Optimization in Evolutionary Algorithms Using Satisfyability Classes. In Bernd Reusch, editor, *International Conference on Computational Intelligence, Theory and Applications, 6th Fuzzy Days*, pages 108–117, Dortmund, Germany, 1999. Springer-Verlag. Lecture Notes in Computer Science Vol. 1625.

[38] Mark Drela. XFOIL: An Analysis and Design System for Low Reynolds Number Aerodynamics. In *Conference on Low Reynolds Number Aerodynamics*, University Of Notre Dame, IN, June 1989.

[39] Francis Ysidro Edgeworth. *Mathematical Physics*. P. Keagan, 1881.

[40] Matthias Ehrgott. *Multicriteria Optimization*. Springer, Berlin, second edition, 2005.

[41] Michael Emmerich and Boris Naujoks. Metamodel Assisted Multiobjective Optimisation Strategies and their Application in Airfoil Design. In I.C. Parmee, editor, *Adaptive Computing in Design and Manufacture VI*, pages 249–260, London, 2004. Springer.

[42] M. Farina and P. Amato. On the Optimal Solution Definition for Many-criteria Optimization Problems. In *Proceedings of the NAFIPS-FLINT International Conference'2002*, pages 233–238, Piscataway, New Jersey, June 2002. IEEE Service Center.

[43] M. Farina and P. Amato. A fuzzy definition of "optimality" for many-criteria optimization problems. *IEEE Transactions on Systems, Man, and Cybernetics Part A—Systems and Humans*, 34(3):315–326, May 2004.

[44] J. Figueira, S. Greco, and M. Ehrgott. *Multiple Criteria Decision Analysis: State of the Art Surveys*. Springer Verlag, Boston, Dordrecht, London, 2005.

[45] José Rui Figueira, Arnaud Liefooghe, El-Ghazali Talbi, and Andrzej P. Wierzbicki. A parallel multiple reference point approach for multi-objective optimization. *European Journal of Operational Research*, 205(2): 390–400, 2010.

[46] Jörg Fliege. An efficient interior-point method for convex multicriteria optimization problems. *Mathematics of Operations Research*, 31:825–845, November 2006. ISSN 0364-765X.

[47] Lawrence J. Fogel. *Artificial Intelligence through Simulated Evolution. Forty Years of Evolutionary Programming*. John Wiley & Sons, Inc., Nueva York, 1999.

[48] Carlos M. Fonseca and Peter J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.

[49] Carlos M. Fonseca and Peter J. Fleming. An Overview of Evolutionary Algorithms in Multiobjective Optimization. *Evolutionary Computation*, 3(1):1–16, Spring 1995.

[50] Carlos M. Fonseca and Peter J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part I: A Unified Formulation. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):26–37, 1998.

[51] Carlos M. Fonseca and Peter J. Fleming. Multiobjective Optimization and Multiple Constraint Handling with Evolutionary Algorithms—Part II: A Application Example. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, 28(1):38–47, 1998.

[52] Tomas Gal and Thomas Hanne. Consequences of dropping nonessential objectives for the application of mcdm methods. *European Journal of Operational Research*, 119(2):373–378, 1999.

[53] Tomas Gal and Heiner Leberling. Redundant objective functions in linear vector maximum problems and their determination. *European Journal of Operational Research*, 1(3):176–184, 1977.

[54] Chariklia A. Georgopoulou and Kyriakos C. Giannakoglou. A multi-objective metamodel-assisted memetic algorithm with strengthbased local refinement. *Engineering Optimization*, 41(10):909–923, 2009.

[55] David Edward Goldberg. *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989.

[56] David Edward Goldberg and J. Richardson. Genetic algorithms with sharing for multimodal function optimization. In John. J. Grefenstette, editor, *Genetic Algorithms and their Applications: Proceedings of the Second International Conference on Genetic Algorithms*, pages 41–49, Hillsdale, Nueva Jersey, EE. UU., 1987. Lawrence Erlbaum.

[57] Thomas Hanne. Global Multiobjective Optimization with Evolutionary Algorithms: Selection Mechanisms and Mutation Control. In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello, and David Corne, editors, *First International Conference on Evolutionary Multi-Criterion Optimization*, pages 197–212. Springer-Verlag. Lecture Notes in Computer Science No. 1993, 2001.

[58] Jeffrey Horn and Nicholas Nafpliotis. Multiobjective Optimization using the Niched Pareto Genetic Algorithm. Technical Report IlliGAl Report 93005, University of Illinois at Urbana-Champaign, Urbana, Illinois, EE. UU., 1993.

[59] Simon Huband, Luigi Barone, Lyndon While, and Phil Hingston. A Scalable Multi-objective Test Problem Toolkit. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 280–295, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[60] Evan Hughes. Evolutionary multi-objective ranking with uncertainty and noise. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, EMO '01, pages 329–343, London, UK, 2001. Springer-Verlag. ISBN 3-540-41745-1.

[61] Evan J. Hughes. Evolutionary Many-Objective Optimisation: Many Once or One Many? In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 1, pages 222–227, Edinburgh, Scotland, September 2005. IEEE Service Center.

[62] Evan J. Hughes. Radar Waveform Optimisation as a Many-Objective Application Benchmark. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 700–714, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[63] Evan J. Hughes. Fitness Assignment Methods for Many-Objective Problems. In Joshua Knowles, David Corne, and Kalyanmoy Deb, editors, *Multi-Objective Problem Solving from Nature: From Concepts to Applications*, pages 307–329. Springer, Berlin, 2008.

[64] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 2424–2431, Hong Kong, June 2008. IEEE Service Center.

[65] A. Jaszkiewicz and R. Slowinski. The light beam search approach -an overview of methodology and applications. *European Journal of Operational Research*, 113(2):300–314, 1999.

[66] Y. Jin and J. Branke. Evolutionary optimization in uncertain environments - a survey. *IEEE Transactions On Evolutionary Computation*, 9(3): 303–317, 2005.

[67] V. Khare, X. Yao, and K. Deb. Performance Scaling of Multi-objective Evolutionary Algorithms. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 376–390, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[68] Joshua Knowles. ParEGO: A Hybrid Algorithm With On-Line Landscape Approximation for Expensive Multiobjective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66, February 2006.

[69] Joshua Knowles and David Corne. Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 757–771, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[70] Joshua D. Knowles and David W. Corne. The Pareto Archived Evolution Strategy: A New Baseline Algorithm for Multiobjective Optimisation. In *1999 IEEE Congress on Evolutionary Computation*, pages 98–105, Washington, D.C., July 1999. IEEE Service Center.

[71] Ikeda Kokolo, Kita Hajime, and Kobayashi Shigenobu. Failure of Pareto-based MOEAs: Does Non-dominated Really Mean Near to

Optimal? In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation 2001 (CEC'2001)*, volume 2, pages 957–962, Piscataway, New Jersey, May 2001. IEEE Service Center.

[72] Peter M. Kruse, Joachim Wegener, and Stefan Wappler. A highly configurable test system for evolutionary black-box testing of embedded systems. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, GECCO '09, pages 1545–1552, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-325-9.

[73] Saku Kukkonen and Jouni Lampinen. Ranking-Dominance and Many-Objective Optimization. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3983–3990, Singapore, September 2007. IEEE Press.

[74] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multiobjective Optimization. *Evolutionary Computation*, 10:263–282, September 2002. ISSN 1063-6560.

[75] John A. Lee and Michel Verleysen. *Nonlinear Dimensionality Reduction*. Information Science and Statistics. Springer, 2007.

[76] Antonio López Jaimes and Alfredo Arias-Montaño Carlos A. Coello Coello. Preference Incorporation to Solve Many-Objective Airfoil Design Problems. In *2011 IEEE Congress on Evolutionary Computation (CEC'2011)*, New Orleans, USA, June 5–8 2011. IEEE Press. To appear.

[77] Antonio López Jaimes and Carlos A. Coello Coello. Study of Preference Relations in Many-objective Optimization. In *2009 Genetic and Evolutionary Computation Conference (GECCO'2009)*, pages 611–618, Montreal, Canada, July 8–12 2009. ACM Press.

[78] Antonio López Jaimes, Carlos A. Coello Coello, and Debrup Chakraborty. Objective Reduction Using a Feature Selection Technique. In *2008 Genetic and Evolutionary Computation Conference (GECCO'2008)*, pages 674–680, Atlanta, USA, July 2008. ACM Press.

[79] Antonio López Jaimes, Carlos A. Coello Coello, and Jesús E. Urías Barrientos. Online Objective Reduction to Deal with Many-Objective Problems. In Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, pages 423–437. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009.

[80] Antonio López Jaimes, Luis Vicente Santana Quintero, and Carlos A. Coello Coello. Ranking methods in many-objective evolutionary algorithms. In Raymond Chiong, editor, *Nature-Inspired Algorithms for Optimisation*, pages 413–434. Springer, Berlin, 2009.

[81] Antonio López Jaimes, Hernán Aguirre, Kiyoshi Tanaka, and Carlos A. Coello Coello. Objective Space Partitioning Using Conflict Information for Many-Objective Optimization. In Robert Schaefer, Carlos Cotta, Joanna Kołodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature–PPSN XI, 11th International Conference, Proceedings, Part I*, pages 657–666. Springer, Lecture Notes in Computer Science Vol. 6238, Kraków, Poland, September 2010.

[82] Antonio López Jaimes, Carlos A. Coello Coello, Hernán Aguirre, and Kiyoshi Tanaka. Adaptive Objective Space Partitioning Using Conflict Information for Many-Objective Optimization. In Ricardo H.C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Grecco, editors, *Evolutionary Multi-Criterion Optimization, 6th International Conference, EMO 2011*, pages 151–165, Ouro Preto, Brazil, April 2011. Springer. Lecture Notes in Computer Science Vol. 6576.

[83] Mariano Luque, Kaisa Miettinen, Petri Eskelinen, and Francisco Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2):450–462, 2009.

[84] Luis Martí, Jesús García, Antonio Berlanga, and José Manuel Molina. An approach to stopping criteria for multi–objective optimization evolutionary algorithms: The MGBM criterion. In *2009 IEEE Conference on Evolutionary Computation (CEC 2009)*, pages 1263–1270, Piscataway, New Jersey, 2009. IEEE Press.

[85] C.A. Mattson, A.A. Mullur, and A. Messac. Smart Pareto filter: Obtaining a minimal representation of multiobjective design space. *Engineering Optimization*, 36(6):721–740, 2004.

[86] Zbigniew Michalewicz and David B. Fogel. *How to Solve It: Modern Heuristics*. Springer, Berlin, 2000.

[87] Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, EE. UU., 1998.

[88] P. Mitra, CA Murthy, and SK Pal. Unsupervised feature selection using feature similarity. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):301–312, 2002.

[89] Sanaz Mostaghim and Hartmut Schmeck. Distance based ranking in many-objective particle swarm optimization. In Günter Rudolph, Thomas Jansen, Simon M. Lucas, Carlo Poloni, and Nicola Beume, editors, *PPSN*, volume 5199 of *Lecture Notes in Computer Science*, pages 753–762. Springer, 2008. ISBN 978-3-540-87699-1.

[90] Shigeru Obayashi and Daisuke Sasaki. Visualization and Data Mining of Pareto Solutions Using Self-Organizing Map. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 796–809, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[91] Vilfredo Pareto. *Cours D'Economie Politique*. F. Rouge, 1896.

[92] Kata Praditwong and Xin Yao. How Well Do Multi-Objective Evolutionary Algorithms Scale to Large Problems. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3959–3966, Singapore, September 2007. IEEE Press.

[93] Robin C. Purshouse and Peter J. Fleming. Conflict, Harmony, and Independence: Relationships in Evolutionary Multi-criterion Optimisation. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 16–30, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[94] Robin C. Purshouse and Peter J. Fleming. On the Evolutionary Optimization of Many Conflicting Objectives. *IEEE Transactions on Evolutionary Algorithms*, 11(6):770–784, December 2007.

[95] Jorge E. Rodríguez, Andrés L. Medaglia, and Juan P. Casas. Approximation to the Optimum Design of a Motorcycle Frame using Finite Element Analysis and Evolutionary Algorithms. In Ellen J. Bass, editor, *Proceedings of the 2005 IEEE Systems and Information Engineering Design Symposium*, pages 277–285. IEEE Press, 2005.

[96] Jorge E. Rodríguez, Andrés L. Medaglia, and Carlos A. Coello Coello. Design of a motorcycle frame using neuroacceleration strategies in moeas. *Journal of Heuristics*, 15:177–196, April 2009. ISSN 1381-1231.

[97] R. S. Rosenberg. *Simulation of genetic populations with biochemical properties*. PhD thesis, University of Michigan, Ann Arbor, Michigan, USA, 1967.

[98] Hiroyuki Sato, Hernán E. Aguirre, and Kiyoshi Tanaka. Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 5–20, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[99] Dhish Kumar Saxena and Kalyanmoy Deb. Dimensionality Reduction of Objectives and Constraints in Multi-Objective Optimization Problems: A System Design Perspective. In *2008 IEEE Congress on Evolutionary Computation (CEC'2008)*, pages 3203–3210, Hong Kong, June 2008. IEEE Service Center.

[100] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, USA, 1984.

[101] J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[102] Oliver Schütze, Marco Laumanns, and Carlos A. Coello Coello. Approximating the Knee of an MOP with Stochastic Search Algorithms. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, pages 795–804. Springer. Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008.

[103] Pratyush Sen and Jian-Bo Yang. *Multiple criteria decision support in engineering design*. Springer, London, UK, 1998. ISBN 3-540-19932-2.

[104] Helmut Sobieczky. Parametric Airfoils and Wings. In K. Fuji and G. S. Dulikravich, editors, *Notes on Numerical Fluid Mechanics, Vol.. 68*, pages 71–88, Wiesbaden, 1998. Vieweg Verlag.

[105] D.M. Somers, United States. National Aeronautics, Space Administration. Scientific, Technical Information Branch, and Langley Research Center. *Design and Experimental Results for a Natural-Laminar-Flow Airfoil for General Aviation Application*. National Aeronautics and Space Administration, Scientific and Technical Information Branch, 1981.

[106] Banu Soylu and Murat Köksalan. A favorable weight-based evolutionary algorithm for multiple criteria problems. *IEEE Transactions on Evolutionary Computation*, 14(2):191–205, April 2010.

[107] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994.

[108] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, 546 pp, 1986.

[109] Ralph E. Steuer and Eng-Ung Choo. An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, 26(1):326–344, 1983.

[110] P. Stewart, D. A. Stone, and P. J. Fleming. Design of robust fuzzy-logic control systems by multi-objective evolutionary methods with hardware in the loop. *Engineering Applications of Artificial Intelligence*, 17(3):275 – 284, 2004. ISSN 0952-1976.

[111] András Szöllös, Miroslav Smíd, and Jaroslav Hájek. Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. *Advances in Engineering Software*, 40(6):419–430, 2009.

[112] K.C. Tan, E.F. Khor, and T.H. Lee. *Multiobjective Evolutionary Algorithms and Applications*. Springer-Verlag, London, 2005. ISBN 1-85233-836-9.

[113] Olivier Teytaud. On the Hardness of Offline Multi-objective Optimization. *Evolutionary Computation*, 15:475–491, December 2007.

[114] Lothar Thiele, Kaisa Miettinen, Pekka J. Korhonen, and Julian Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17:411–436, September 2009. ISSN 1063-6560.

[115] Rasmus K. Ursem. Multinational gas: Multimodal optimization techniques in dynamic environments. In *GECCO*, pages 19–26, 2000.

[116] David A. Van Veldhuizen. *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[117] Tobias Wagner, Nicola Beume, and Boris Naujoks. Pareto-, Aggregation-, and Indicator-Based Methods in Many-Objective Optimization. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages

742–756, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403.

[118] Tobias Wagner, Heike Trautmann, and Boris Naujoks. OCD: Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based on Statistical Testing. In Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *Evolutionary Multi-Criterion Optimization. 5th International Conference, EMO 2009*, pages 198–215. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009.

[119] Edward J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85:664–675, 1990.

[120] Upali K. Wickramasinghe, Robert Carrese, and Xiaodong Li. Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm. In *2010 IEEE Congress on Evolutionary Computation*, pages 1–8, Barcelona, Spain, 2010. IEEE Service Center.

[121] A.P. Wierzbicki. The use of reference objectives in multiobjective optimisation. In Fandel G. and Gal T., editors, *MCDM theory and Application, Proceedings*, number 177 in Lecture notes in economics and mathematical systems, pages 468–486, Hagen, 1980. Springer Verlag.

[122] A.P. Wierzbicki. A methodological guide to multiobjective optimization. In K. Iracki, K. Malanowski, and S. Walukiewicz, editors, *Optimization Techniques, Part 1*, volume 22 of *Lecture Notes in Control and Information Sciences*, pages 99–123. Springer, Berlin, 1980.

[123] Piotr Wozniak. Preferences in multi-objective evolutionary optimisation of electric motor speed control with hardware in the loop. *Applied Soft Computing*, 11(1):49 – 55, 2011. ISSN 1568-4946.

[124] Saúl Zapotecas Martínez and Carlos A. Coello Coello. A Memetic Algorithm with Non Gradient-Based Local Search Assisted by a Meta-Model. In Robert Schaefer, Carlos Cotta, Joanna Kolodziej, and Günter Rudolph, editors, *Parallel Problem Solving from Nature–PPSN XI*, volume 6238, pages 576–585, Kraków, Poland, September 2010. Springer. Lecture Notes in Computer Science.

[125] Milan Zeleny. *Linear Multiobjective Programming*, volume 95 of *Lecture notes in economics and mathematical systems*. Springer Verlag, Berlin, 1974.

[126] Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[127] Eckart Zitzler and Simon Künzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[128] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, Noviembre 1999.

[129] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty, editors, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pages 95–100, Athens, Greece, 2002.

[130] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.