

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

UNIDAD ZACATENCO
DEPARTAMENTO DE COMPUTACIÓN

Un nuevo algoritmo evolutivo multi-objetivo basado en una medida de desempeño

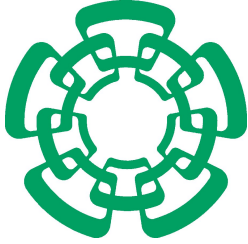
Tesis que presenta
Cynthia Adriana Rodríguez Villalobos

para obtener el Grado de
Maestra en Ciencias en la Especialidad de Computación

Director de la Tesis
Dr. Carlos A. Coello Coello

México,D.F.

Septiembre 2011



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

ZACATENCO CAMPUS
COMPUTER SCIENCE DEPARTMENT

A new multi-objective evolutionary algorithm based on a performance assessment indicator

Submitted by
Cynthia Adriana Rodríguez Villalobos

as the fulfillment of the requirement for the degree of
Master in Computer Science

Advisor
Dr. Carlos A. Coello Coello

Mexico City

September 2011

ABSTRACT

Many problems in engineering involve the simultaneous optimization of several conflicting objectives. In most cases, *multi-objective optimization problems* (MOPs) don't have a unique optimal solution but a set of solutions representing different trade-offs amongst the objectives.

Multi-objective evolutionary algorithms (MOEAs) have been very successful in solving MOPs due to their simplicity and applicability. Most MOEAs use the Pareto dominance relation as a selection criterion which is effective for 2 or 3 objectives but scales poorly since the proportion of incomparable solutions that it generates, quickly increases with the number of objectives.

In this thesis we present a MOEA which uses the Δ_p quality indicator in its selection mechanism instead of the Pareto dominance relation. Our proposed approach is shown to produce competitive results for bi- and tri-objective problems and shows promising results for problems with many (4 or more) objectives.

RESUMEN

Muchos problemas de ingeniería requieren la optimización simultánea de varios objetivos que se encuentran en conflicto. Generalmente, los problemas de optimización multi-objetivo (POMs) no tienen una solución única sino un conjunto de soluciones representando diferentes compromisos entre los objetivos.

Los algoritmos evolutivos multi-objetivo (AEMOs) han sido exitosos en la solución de POMs debido a su simplicidad y facilidad de aplicación. La mayoría de los AEMOs utilizan la dominancia de Pareto como criterio de selección, la cual es efectiva en problemas con 2 ó 3 objetivos pero no es escalable ya que la proporción de soluciones incomparables que genera crece rápidamente con el número de objetivos.

En esta tesis presentamos un AEMO que utiliza el indicador de calidad Δ_p en su mecanismo de selección en lugar de la dominancia de Pareto. Nuestra propuesta obtiene resultados competitivos para problemas con 2 y 3 objetivos y presenta resultados prometedores para problemas con muchos (4 o más) objetivos.

ACKNOWLEDGMENTS

*I'm antisocial, they say. I don't mix.
It's so strange. I'm very social indeed.
It all depends on what you mean by social, doesn't it?
Social to me means talking to you about things like this.*
— *Fahrenheit 451*, Ray Bradbury

First and foremost, I would like to thank my advisor, Dr. Carlos A. Coello Coello, for his guidance and support through the development of this work. I would also like to thank Dr. Luis Gerardo de la Fraga and Dr. Oliver Schütze for agreeing to read and review this thesis.

I want to thank all my colleagues at CINVESTAV, specially Adriana Lara López, Alfredo Arias Montaña, Antonio López Jaimes, Saúl Zapotecas Martínez, and Sergio Luis Pérez Pérez for providing feedback on my work and making my stay at CINVESTAV enjoyable.

I also want to thank the staff of the Computer Science department, specially Sofía Reza and Felipa Rosas, for their invaluable assistance throughout my stay at CINVESTAV.

Moreover, I wish to thank Dr. Francisco Javier Zaragoza Martínez, M.Sc. Oscar Alvarado Nava, M.Sc. Jaime Grabinsky y Steider, and Dr. Rafael López Bracho from UAM-A, as well as Dr. Guillermo Morales Luna from CINVESTAV-IPN for all the advise and encouragement given the past couple of years.

Special thanks to my parents and my sister for all the patience and support they've given me throughout my studies.

Finally, I would like to thank CONACyT, for the economic support provided. The work reported in this thesis was derived from the CONACyT project entitled "*Escalabilidad y Nuevos Esquemas Híbridos en Optimización Multiobjetivo*" (Ref. 103570), whose Principal Investigator is Dr. Carlos A. Coello Coello.

CONTENTS

1	INTRODUCTION	1
2	MULTI-OBJECTIVE OPTIMIZATION	3
2.1	Basic concepts	4
2.2	Multi-objective optimization algorithms	8
3	EVOLUTIONARY COMPUTATION	11
3.1	Neo-Darwinism	12
3.2	Evolutionary algorithms	13
3.2.1	Paradigms	17
3.3	Multi-objective evolutionary algorithms	19
3.3.1	Elitism	19
3.3.2	Fitness assignment	20
4	QUALITY INDICATORS FOR MULTI-OBJECTIVE OPTIMIZATION	23
4.1	Unary quality indicators	24
4.1.1	Properties of unary quality indicators	24
4.1.2	Discussion of some unary quality indicators	25
4.1.2.1	The hypervolume indicator	25
4.1.2.2	The GD, IGD, and Δ_p indicators	27
4.1.2.3	Spacing indicators	29
4.2	Binary quality indicators	30
4.2.1	Discussion of some binary quality indicators	30
4.2.1.1	The ϵ -indicators	30
4.2.1.2	The coverage indicator	32
5	INDICATOR-BASED MULTI-OBJECTIVE SEARCH	33
5.1	Indicator-based archiving algorithms	34
5.1.1	LAHC	34
5.1.2	ESP	35
5.2	Indicator-based selection	36
5.2.1	IBEA	36
5.2.2	SMS-EMOA	38
5.2.3	Iterative approach to indicator-based multi-objective optimization	40
5.2.4	MO-CMA-ES	41

Contents

5.2.5	DF-SMS-EMOA	43
5.3	Set-oriented multi-objective evolutionary algorithms	44
5.3.1	SPAM	44
5.3.2	SPAM ⁺	45
5.4	Closing remarks	46
6	OUR PROPOSED APPROACH	47
6.1	Differential evolution	48
6.2	Fitness assignment	49
6.2.1	Individual contribution to the GD_p indicator	49
6.2.2	Individual contribution to the IGD_p indicator	50
6.2.3	Individual contribution to the Δ_p indicator	51
6.3	Reference set construction	51
6.4	The algorithm	53
7	PERFORMANCE ASSESSMENT	55
7.1	ZDT test problems	56
7.2	DTLZ test problems	59
7.3	Scalability of our approach	67
7.4	Closing remarks	69
8	CONCLUSIONS AND FUTURE WORK	71
8.1	Conclusions	71
8.2	Future work	73
A	TEST PROBLEMS	75
A.1	Zitzler-Deb-Thiele test set	75
A.2	Deb-Thiele-Laumanns-Zitzler test set	80
B	COMPLETE RESULTS	87
B.1	ZDT test problems	88
B.2	DTLZ test problems	88
B.3	Scalability results	95
	Bibliography	95

LIST OF ACRONYMS

CMA-ES	covariance matrix adaptation evolution strategy.
DE	differential evolution.
DF	desirability function.
DF-SMS-EMOA	desirability function SMS-EMOA.
DI	desirability index.
DP	number of dominated points.
DTLZ	Deb-Thiele-Laumanns-Zitzler.
EA	evolutionary algorithm.
EP	evolutionary programming.
ES	evolution strategy.
ESP	evolution strategy with probabilistic mutation.
GA	genetic algorithm.
GD	generational distance.
HV	hypervolume indicator.
IBEA	indicator based evolutionary algorithm.
IBMOLS	indicator-based multi-objective local search.
IGD	inverted generational distance.
IHSO	incremental hypervolume by slicing objectives.
LAHC	Lebesgue archiving hill-climber.
MO-CMA-ES	multi-objective covariance matrix adaptation evolution strategy.
MOEA	multi-objective evolutionary algorithm.
MOGA	multi-objective genetic algorithm.
MOP	multi-objective optimization problem.
NDS	non-dominated sorting.
NPGA	niched Pareto genetic algorithm.
NSGA	non-dominated sorting genetic algorithm.

List of Acronyms

NSGA-II	non-dominated sorting genetic algorithm II.
PAES	Pareto archived evolution strategy.
PD	Pareto dominance.
SMS-EMOA	S metric selection evolutionary multi-objective algorithm.
SOP	single-objective optimization problem.
SPAM	set preference algorithm for multi-objective optimization.
SPEA	strength Pareto evolutionary algorithm.
SPEA ₂	strength Pareto evolutionary algorithm 2.
VEGA	vector evaluated genetic algorithm.
ZDT	Zitzler-Deb-Thiele.

LIST OF FIGURES

Figure 2.1	Search spaces in a MOP	4
Figure 2.2	Pareto dominance relation	6
Figure 2.3	Pareto front	7
Figure 2.4	Ideal and nadir objective vectors	7
Figure 3.1	Encoding in an EA	14
Figure 3.2	Population in an EA	15
Figure 3.3	Example of a crossover operator in EA	16
Figure 3.4	Example of a mutation operator in EA	16
Figure 4.1	I_H for a 3-objective MOP	26
Figure 4.2	ϵ -indicators for a bi-objective MOP	31
Figure 5.1	Hypervolume measure used by ESP on a bi-objective MOP	36
Figure 5.2	Binary additive ϵ -indicator used by IBEA for bi-objective MOP	37
Figure 5.3	Binary hypervolume-indicator used by IBEA for bi-objective MOPs	38
Figure 6.1	Individual I_{GD_p} contribution	49
Figure 6.2	Individual I_{IGD_p} contributions	50
Figure 6.3	GD and IGD contributing elements	52
Figure 6.4	Reference set construction for a bi-objective MOP	53
Figure 7.1	Indicators for ZDT3	57
Figure 7.2	Indicators for ZDT4	57
Figure 7.3	Graphical results for ZDT3	58
Figure 7.4	Graphical results for ZDT4	58
Figure 7.5	Indicators for DTLZ1	60
Figure 7.6	Indicators for DTLZ3	60
Figure 7.7	Graphical results for DTLZ1	61
Figure 7.8	Graphical results for DTLZ3	61

List of Figures

Figure 7.9	Indicators for DTLZ5	63
Figure 7.10	Indicators for DTLZ6	63
Figure 7.11	Graphical results for DTLZ5	64
Figure 7.12	Graphical results for DTLZ6	64
Figure 7.13	Indicators for DTLZ7	65
Figure 7.14	Graphical results for DTLZ7	66
Figure 7.15	Indicator values for the DTLZ2 scaled problem	68
Figure 7.16	Running time for the DTLZ2 scaled problem	68
Figure A.1	Pareto front for ZDT1	75
Figure A.2	Pareto front for ZDT2	76
Figure A.3	Pareto front for ZDT3	77
Figure A.4	Pareto front for ZDT4	78
Figure A.5	Pareto front for ZDT6	79
Figure A.6	Pareto front for DTLZ1	80
Figure A.7	Pareto front for DTLZ2	81
Figure A.8	Pareto front for DTLZ3	82
Figure A.9	Pareto front for DTLZ4	83
Figure A.10	Pareto front for DTLZ5	84
Figure A.11	Pareto front for DTLZ6	85
Figure A.12	Pareto front for DTLZ7	86

LIST OF ALGORITHMS

3.1	Basic structure of an EA	14
3.2	Evolution strategy	17
3.3	Evolutionary programming	18
3.4	Genetic algorithm	19
5.1	Basic SMS-EMOA	39
5.2	$(1 + \lambda)$ -CMA-ES	42
5.3	$\lambda_{MO} \times (1 + \lambda)$ -MO-CMA-ES	42
5.4	DF-SMS-EMOA	44
5.5	Basic SPAM	45
5.6	SPAM ⁺	46
6.1	DE/rand/1/bin	48
6.2	Reference set construction	53
6.3	DE/rand/1/bin with Δ_p -based selection	54

List of Algorithms

LIST OF TABLES

Table 7.1	Parameters used for all the test problems	55
Table 7.2	Parameters used for our scalability testing	67
Table 7.3	Resolution values for Δ_p -DE	67
Table B.1	Reference points used for I_H in the ZDT test problems .	88
Table B.2	Reference points used for I_H in the DTLZ test problems	88
Table B.3	Results for the ZDT functions: I_{GD_1} and I_{IGD_1}	89
Table B.4	Results for the ZDT functions: I_H and I_{SS}	90
Table B.5	Results for the ZDT functions: I_{e^+} and I_C	91
Table B.6	Results for DTLZ functions: I_{GD_1} and I_{IGD_1}	92
Table B.7	Results for the DTLZ functions: I_H and I_{SS}	93
Table B.8	Results for the DTLZ functions: I_{e^+} and I_C	94
Table B.9	Results for the DTLZ2 function: I_{GD}	95
Table B.10	Results for the DTLZ2 function: I_H	96
Table B.11	Results for the DTLZ2 function: running time	97

List of Tables

INTRODUCTION

Many engineering problems require the simultaneous optimization of two or more conflicting objectives. Such problems are called *multi-objective optimization problems* (MOPs). In most cases, these problems have several optimal solutions which, in absence of any further information, are all equally good.

Multi-objective evolutionary algorithms (MOEAs) have often been used to solve MOPs because of their simplicity and wide applicability. Most of these MOEAs use the Pareto dominance relation to compare solutions in the population: Pareto-based MOEAs work well when the MOP has a small number of objectives but they struggle to solve MOPs with a large number of objectives.

Since Pareto-based MOEAs scale poorly, alternative approaches using other relations to identify promising solutions have been proposed. Amongst them, approaches using quality indicators have shown promising results. Most indicator-based MOEAs rely on the hypervolume indicator because it is the only known Pareto-compliant *unary* quality indicator. These approaches produce good results but at a very high computational cost.

In this thesis, we propose a MOEA which replaces the Pareto dominance relation with a quality indicator that has some metric properties. Our proposed approach shows competitive results with respect to those obtained by state-of-the-art MOEAs when dealing with MOPs with a small number of objectives and promising results when dealing with a larger number of objectives, while requiring a very reasonable computational cost.

The organization of this thesis is as follows:

- In Chapter 2 we discuss basic concepts of multi-objective optimization which are required to understand the remainder of this document. A short review of the most popular multi-objective optimization algorithms is also provided.
- In Chapter 3 a brief introduction to evolutionary computation is presented. At the beginning of the chapter, the main biological concepts which inspire evolutionary computation are discussed. Then, the main components and the basic structure of evolutionary algorithms are described. Finally, the main characteristics of MOEAs are discussed.
- Chapter 4 introduces quality indicators as a means to evaluate the performance of a MOEA. Some representative unary and binary quality indicators are then briefly described.
- In Chapter 6 we describe a MOEA whose selection mechanism is based on the minimization of the Δ_p quality indicator.
- In Chapter 7 we compare our approach to other state-of-the-art MOEAs using the quality indicators described in Chapter 4.
- In Chapter 8 we provide some final remarks and some possible paths for future research.

MULTI-OBJECTIVE OPTIMIZATION

Contents

2.1	Basic concepts	4
2.2	Multi-objective optimization algorithms	8

Optimization can be defined as the process which seeks the minimum or maximum value of one or more functions representing the objectives of the problem. The optimization problems which involve only one objective are called *single-objective optimization problems (SOPs)* while those which involve two or more objectives are called *multi-criteria* or *multi-objective optimization problems (MOPs)*.

In *SOPs*, determining whether one solution is better than another is always possible and only one optimal value is possible for the objective function. Most *SOPs* have a unique optimal solution.

On the other hand, most *MOPs* don't have a unique solution which optimizes all the objectives at the same time since there is normally some degree of conflict among the objectives. Thus, the solution of a *MOP* is often a set of values representing different trade-offs among all the objectives of the problem.

In this chapter, we present some basic concepts of multi-objective optimization, required to understand the rest of this thesis. Additionally, we also provide a short review of the most popular multi-objective optimization algorithms reported in the mathematical programming literature.

2.1 BASIC CONCEPTS

Informally, a **MOP** consists of searching for some decision variable vectors which optimize a set of objective functions while satisfying the problem's constraints.

Definition 1 (Decision variables). The *decision variables* x_i , $i = 1, \dots, n$ are the n parameters which will be chosen during the optimization process.

The *decision variable vector* is represented as

$$\vec{x} = [x_1, x_2, \dots, x_n]^T \quad (2.1)$$

The set \mathbb{R}^n is known as the *decision variable space* (see Figure 2.1).

Definition 2 (Objective functions). An optimization problem has $k \geq 1$ *objective functions* which determine how good a given solution is. They are expressed in terms of the decision variable vector and are often modeled as a mathematical expression which is represented as $f_i(\vec{x})$, $i \in \{1, \dots, k\}$.

The objective function vector $\vec{f}(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^k$ is defined as:

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \dots, f_k(\vec{x})]^T \quad (2.2)$$

The set \mathbb{R}^k is known as *objective function space* (see Figure 2.1).

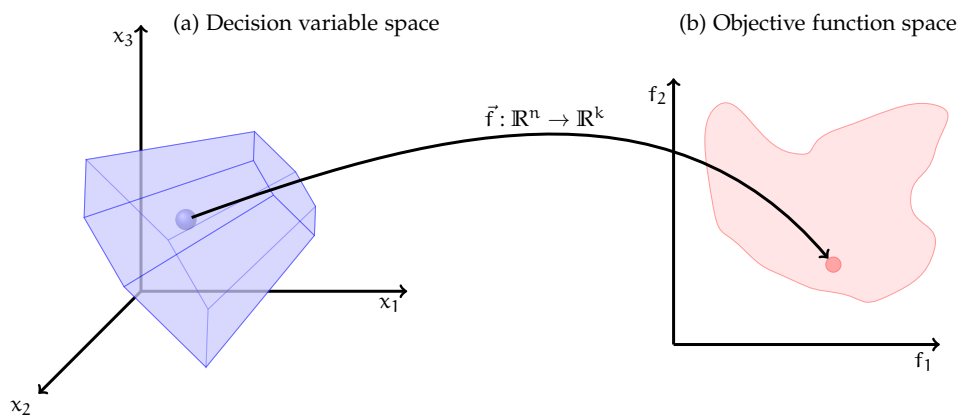


Figure 2.1.: Search spaces in a **MOP**

Definition 3 (Feasible set). The *feasible set* \mathcal{X} is implicitly defined by a vector \vec{g} of $q \geq 0$ inequality constraints and a vector \vec{h} of $r \geq 0$ equality constraints

$$\vec{g}(\vec{x}) = [g_1(\vec{x}), \dots, g_q(\vec{x})]^T \leq 0 \quad (2.3)$$

$$\vec{h}(\vec{x}) = [h_1(\vec{x}), \dots, h_r(\vec{x})]^T = 0 \quad (2.4)$$

Definition 4 (Optimization problems). Formally, every optimization problem can be expressed as:

$$\min_{\vec{x} \in \mathcal{X}} \vec{f}(\vec{x}) \quad (2.5)$$

where $\vec{f}: \mathbb{R}^n \rightarrow \mathbb{R}^k$ is the objective function vector, $\vec{x} \in \mathbb{R}^n$ is the decision variable vector, and $\mathcal{X} \subseteq \mathbb{R}^n$ is the feasible set.

To solve an optimization problem, we must establish what minimizing a function means.

For **SOPs** ($k = 1$), the relation “less than or equal” (\leq) allows the comparison between different solutions while inducing a total order in \mathbb{R} . A solution $\vec{x}^* \in \mathcal{X}$ is optimal if and only if

$$\forall \vec{y} \in \mathcal{X} \quad f(\vec{x}^*) \leq f(\vec{y}) \quad (2.6)$$

Using this relation, there may be several different optimal decision variable vectors but all of them must have the same objective function value.

For **MOPs**, the *Pareto dominance* relation originally proposed by Francis Ysidro Edgeworth [17] and later generalized by Vilfredo Pareto [42] is normally used to compare different solutions.

Definition 5 (Pareto dominance). Given a **MOP** and $\vec{x}_1, \vec{x}_2 \in \mathcal{X}$, \vec{x}_1 *dominates* \vec{x}_2 ($\vec{x}_1 \prec \vec{x}_2$) if and only if

$$\begin{aligned} \forall i \in \{1, \dots, k\} \quad & f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \\ \exists j \in \{1, \dots, k\} \quad & f_j(\vec{x}_1) < f_j(\vec{x}_2) \end{aligned} \quad (2.7)$$

Pareto dominance doesn't induce a total order in \mathbb{R}^k since some solutions may be *incomparable*. Therefore, most **MOPs** don't have a unique solution but a set of incomparable solutions.

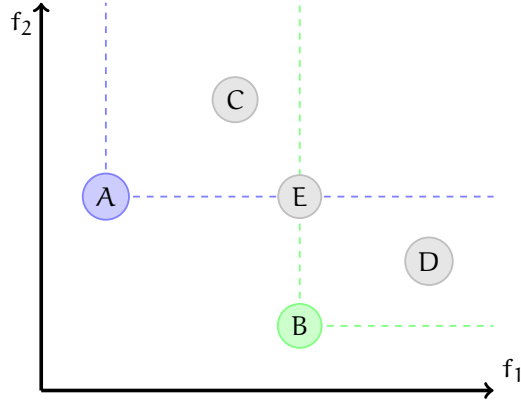


Figure 2.2.: Pareto dominance relation

Figure 2.2 illustrates the Pareto dominance relation for a bi-objective optimization problem: $A \prec C$ since A is better according to both f_1 and f_2 , $A \prec E$ since A is equal according to f_2 but better according to f_1 , $B \prec E$ since B is equal according to f_1 but better according to f_2 , $B \prec D$ since B is better according to both f_1 and f_2 , and A and B are incomparable.

Definition 6 (Pareto optimal set). In a **MOP**, the *Pareto optimal set* P is defined as

$$P = \{\vec{x}^* \in \mathcal{X} \mid \forall \vec{y} \in \mathcal{X} \quad \vec{y} \not\prec \vec{x}^*\} \quad (2.8)$$

Definition 7 (Pareto front). Given a **MOP** and its Pareto optimal set P , the *Pareto front* is defined as:

$$PF = \{\vec{u} = (f_1(\vec{x}), \dots, f_k(\vec{x})) \mid \vec{x} \in P\} \quad (2.9)$$

Figure 2.3 shows the objective function space for a bi-objective optimization problem. The lighter points are all dominated while the darker points form the Pareto front.

Definition 8 (Ideal and nadir objective vectors). Given a **MOP** and its Pareto optimal set P , the ideal objective vector is defined as:

$$f_{\text{ideal}} = (\min_{x \in P} f_1(x), \dots, \min_{x \in P} f_k(x)) \quad (2.10)$$

If the ideal vector objective represents an existing solution, then the objectives aren't conflicting and the solution of the **MOP** is unique. Such problems,

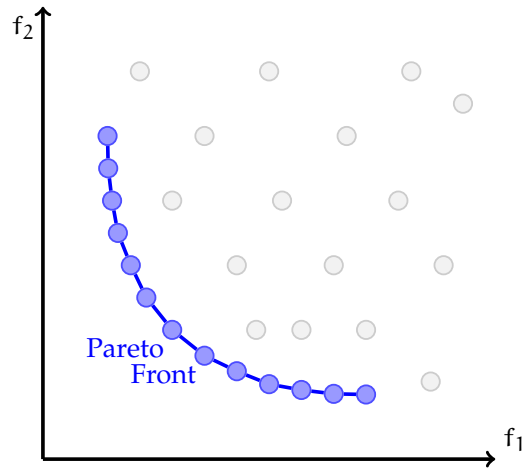


Figure 2.3.: Pareto front

however, are not of interest for the purposes of this thesis, since they can be solved by optimizing separately each of their objectives.

Similarly, the nadir objective vector is defined as:

$$f_{\text{nadir}} = (\max_{x \in P} f_1(x), \dots, \max_{x \in P} f_k(x)) \quad (2.11)$$

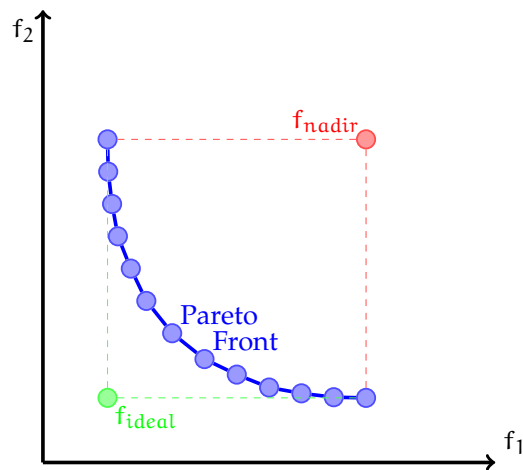


Figure 2.4.: Ideal and nadir objective vectors

The ideal and nadir objective vectors provide bounds (Figure 2.4) for the Pareto front and are often used to normalize objective functions [40]

2.2 MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS

In the mathematical programming literature, two types of optimization techniques are considered:

1. Deterministic methods
2. Stochastic methods

Many deterministic optimization methods have been proposed in the specialized literature such as:

- *Enumeration methods*

This is the simplest optimization algorithm that one can design: all possible solutions are generated and evaluated in order to find the Pareto optimal set. Clearly, this sort of approach is inapplicable in most problems because of its obviously high computational cost, which quickly makes it prohibitive. This sort of approach can only be used when dealing with small (discrete) search spaces.

Land and Doig [38] proposed an improvement to this sort of approach called *branch-and-bound*. This sort of technique is aimed for discrete search spaces and is based on a systematic enumeration procedure in which subsets of solutions are discarded for not being promising for the search.

- *Hill-climbing methods*

These methods take an initial solution and then attempt to improve it through small and systematic changes in one of the decision variables. The process is repeated until no further improvement can be achieved.

- *Classical optimization algorithms*

These methods transform the MOP into a single, parametrized objective function. Then, several optimization runs with different parameter settings must be performed in order to obtain an approximation of the Pareto optimal set. Some examples of these methods are:

– *The weighted sum method:*

The general idea of this method is to associate each objective function with a weighting coefficient $w_i \in [0, 1]$, $i = 1, \dots, k$, $\sum_1^k w_i = 1$ and minimizing the weighted sum of all the objectives. Some properties of this method are [40]:

- * If all weighting coefficients are positive, then the solution of the obtained SOP is a Pareto optimal solution for the MOP.
- * If the solution of the obtained SOP is unique, then it is a Pareto optimal solution for the MOP.
- * This method can generate every Pareto optimal solution for convex MOPs.
- * This approach cannot generate non-convex portions of the Pareto front regardless of the weights used.

– *The weighted metrics method:*

This method scalarizes the objectives by using weighted Minkowski distances from every solution to the ideal objective vector f_{ideal} . The resulting SOP is:

$$\min_{\vec{x} \in \mathcal{X}} \left(\sum_{i=1}^k w_i |f_i(\vec{x}) - f_{\text{ideal}[i]}|^p \right)^{\frac{1}{p}} \quad (2.12)$$

where $f_{\text{ideal}[i]}$ represents the i -th element of the ideal vector; $w_i \in [0, 1]$, $i = 1, \dots, k$, $\sum_{i=1}^k w_i = 1$ are the weights assigned to each function; and the parameter $p \in [1, \infty)$ defines the metric that is used:

- * $p = 1$: the problem is equivalent to the weighted sum approach
- * $p = 2$: minimizes a weighted Euclidean distance
- * $p = \infty$: minimizes the weighted Tchebycheff metric. For this case, the weighted metrics method can generate every Pareto optimal solution [40].

– *The ϵ -constraint method:*

In this method, one of the objectives f_l , $l \in \{1, \dots, k\}$ is chosen to be minimized while the others are transformed into constraints bounded by a ϵ_j , $j \in \{1, \dots, l-1, l+1, \dots, k\}$. Some properties of this method are [40]:

- * The solution of the obtained SOP is weakly Pareto optimal.
- * If the solution of the obtained SOP is unique and $f_i = \epsilon_i$, $i \in \{1, \dots, l-1, l+1, \dots, k\}$ then it is a Pareto optimal solution for the MOP.
- * This method can generate every Pareto optimal solution for the MOPs, but at a high computational cost because of the many optimizations involved.

– *Lexicographic ordering:*

In this method, the objectives are ordered according to their importance by the decision maker. Then, the objectives are optimized sequentially from the best to the worst with the optimal value of each objective being added as a constraint for the following optimizations. The process stops when the k objective functions have been optimized or if a unique optimal solution is found. The solution of this method is Pareto optimal for the original MOP [40], but will normally correspond to a strongly biased solution (i. e. , a solution that lies at one extreme of the Pareto front).

Summarizing, the application of these techniques can be very expensive or even infeasible if the MOP is, for example, multi-frontal, if it has a discontinuous decision variable space, if it has a discontinuous objective function space, if it is non-differentiable, etc. In those cases, stochastic methods such as evolutionary algorithms provide a viable alternative. *Multi-objective evolutionary algorithms (MOEAs)* do not guarantee finding optimal solutions but can often find good approximations at a reasonable computational cost. A more detailed description of evolutionary algorithms is presented in Chapter 3.

EVOLUTIONARY COMPUTATION

Contents

3.1	Neo-Darwinism	12
3.2	Evolutionary algorithms	13
3.2.1	Paradigms	17
3.3	Multi-objective evolutionary algorithms	19
3.3.1	Elitism	19
3.3.2	Fitness assignment	20

Evolutionary computation consists of stochastic techniques, inspired by *neo-Darwinism*, in which evolution is viewed as an optimization process. These techniques are generically called *evolutionary algorithms (EAs)*.

EAs have been very successful solving optimization problems due to their following features:

- They don't need specific knowledge about the problem which implies that they can be used for many types of problems with relatively little problem-specific information.
- They can act as effective global optimizers since they are population-based and are, therefore, less prone to getting trapped in local optima
- They are easy to understand, implement, and parallelize.
- They can be easily hybridized with other optimization techniques (e. g. , with mathematical programming techniques).

In this chapter, we present a brief description of evolutionary algorithms, and how can they be used to solve *MOPs*.

3.1 NEO-DARWINISM

Neo-Darwinism has three core components:

- *Darwin's theory of evolution:*

Charles Darwin introduced his theory of evolution by *natural selection* as an explanation to biological diversity in his book *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* [14]. According to this theory:

- Individuals in a population are significantly different from one another.
- Some characteristics of individuals are inheritable.
- The individuals that are better suited to their environment are more likely to survive and reproduce than those less suited to their environment.
- Populations change to adapt to their environments. The variations can accumulate over time to form new species.

- *Weismann's germ plasm theory:*

Friedrich Leopold August Weismann proposed his germ plasm theory as a mechanism for inheritance in Darwin's theory of evolution. This theory states that individuals consist of two kinds of cells:

- *Germ cells:*

These contain inheritable information that cannot be altered by the abilities acquired during the individual's life

- *Somatic cells:*

These are responsible for bodily functions.

- *Mendel's genetics:*

The laws of inheritance were derived by Gregor Johann Mendel from his hybridization experiments in pea plants. Mendel proposed his laws in the paper *Experiments on Plant Hybridization* [39]:

1. *The Law of Segregation:*
During gamete formation, the two alleles of a gene separate so that each gamete gets only one of them.
2. *The Law of Independent Assortment:*
Also known as the *Inheritance Law*, it states that during gamete formation, alleles of different genes assort independently from one another.
3. *The Law of Dominance:*
When an organism has two different alleles for a trait (one allele being dominant and the other recessive), the dominant allele will be expressed while the recessive allele will remain hidden.

According to neo-Darwinism, four stochastic processes act upon all species:

1. **Reproduction:**
Biological process which allows the creation of new individuals.
2. **Mutation:**
A modification in the genetic information of an organism which can be inherited by its descendants.
3. **Competition:**
Interaction between individuals which belong to a community due to the availability of certain limited resources.
4. **Selection:**
Process by which the individuals who adapt to their environment are more likely to survive and reproduce than those who don't.

3.2 EVOLUTIONARY ALGORITHMS

EAs attempt to mimic evolution by simulating reproduction, mutation, competition, and selection with the aim of solving search and optimization problems. The basic structure of an EA is shown in Algorithm 3.1.

Algorithm 3.1 Basic structure of an EA

Generate initial population *randomly*
 Compute the *fitness* of each individual
repeat
 Select a parent-set P based on their fitness values
 Apply *variation operators* to the elements of P to get the offspring set O
 Compute the *fitness* of each individual in O
 Produce the new population using the best individuals from $P \cup O$
 or only the elements in O .
until a stopping criterion is met

The main components of an EA are [18]:

1. *Encoding*:

There are two levels of encoding in EAs (Figure 3.1):

a) *Genotype*:

The genotype consists of the encoding used to represent to decision variables in the EA. Each individual is represented by a *chromosome* which is a data structure containing a group of *genes*. Each *gene* represents a decision variable of the problem and can be encoded as a binary string, as an integer, as a real number, etc.

b) *Phenotype*:

The phenotype is the decoded value of the chromosomes into the decision variable space of the problem.

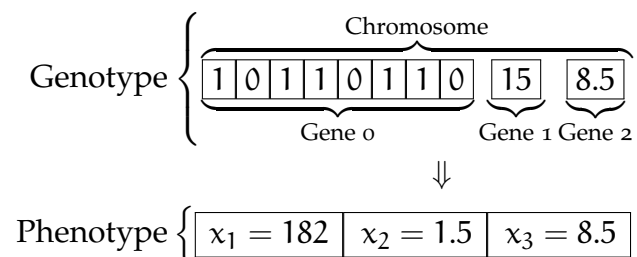


Figure 3.1.: Encoding in an EA

2. *Population:*

A set of individuals representing solutions to a given problem (Figure 3.2). It is desirable to keep a good diversity of solutions in the population.

1	0	0	1	1	0	1	0	Individual 1
0	1	0	1	0	1	0	1	Individual 2
⋮								⋮
1	1	0	0	0	0	1	1	Individual n – 1
0	0	1	0	1	1	1	1	Individual n

Figure 3.2.: Population in an EA

3. *Fitness function:*

The fitness function of an EA assigns a quality measure that relates an individual to the problem's objective function and allows its comparison to the other individuals in the population.

4. *Parent selection mechanism:*

This mechanism defines which individuals in the population will be allowed to reproduce according to their fitness.

5. *Variation operators:*

These operators introduce new individuals into the population. The main variation operators in EAs are:

a) *Crossover operator (Figure 3.3):*

It simulates sexual reproduction in EAs by mixing the genes of two or more selected parents to create the chromosomes of their offspring. These operators exploit the most promising areas of the search space.

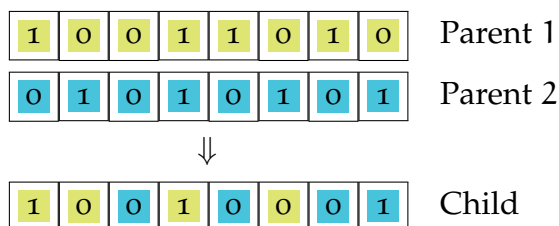


Figure 3.3.: Example of a crossover operator in EA

b) *Mutation operator (Figure 3.4):*

It is a slight random variation produced to the genes of an individual. Such randomness allows for the exploration of unknown areas of the search space.

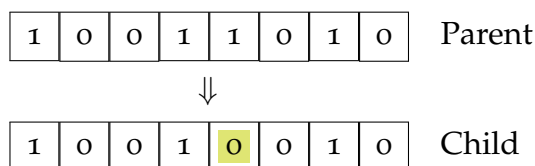


Figure 3.4.: Example of a mutation operator in EA

6. *Survivor selection mechanism:*

This mechanism defines which individuals will survive in the population at a given time. It normally favors individuals with the highest fitness values.

Due to the stochastic nature of EAs, good solutions found during a certain generation could get lost in the next, after applying the variation operators. To avoid this situation, De Jong [33] proposed an additional mechanism called *elitism* which ensures that the best individual in the current population will be kept for the following generation.

3.2.1 Paradigms

There are three main evolutionary algorithms' paradigms:

- *Evolution strategies (ESs)*:
ESs were proposed by Ingo Rechenberg and Hans-Paul Schwefel [44].
The structure of ESs is described in Algorithm 3.2

Algorithm 3.2 Evolution strategy

Generate initial population *randomly*
 Compute the *fitness* of each individual
repeat
 Select a parent-set P
 Apply *crossover* to create offspring set O
 Apply *mutation* to the elements of O
 Compute the *fitness* of each individual in O
 Select the best individuals for the following generation using
 (μ, λ) -selection or $(\mu + \lambda)$ -selection
until a stopping criterion is met

In ESs, individuals are represented as real-valued vectors. The crossover operator creates one child using 2 or more parents that are randomly selected with a uniform distribution from a population of μ individuals. The crossover operator is applied $\lambda \geq \mu$ times to obtain the offspring set.

The mutation operator is applied to all the generated offspring based on a Gaussian distribution where the mean (ζ) and the standard deviation (σ^2) are required as parameters. Generally, σ^2 is modified during the search to allow for a good exploration at the beginning and a better exploitation of solutions by the end of the process.

Once the λ offspring have been generated and mutated, the best μ individuals are deterministically chosen for the following generation. Two selection schemes are possible:

1. (μ, λ) -selection:
The best μ individuals from the children set are deterministically chosen.

2. $(\mu + \lambda)$ -selection:

The best μ individuals from the union of the parent and children sets are deterministically chosen. This scheme is implicitly elitist.

- *Evolutionary programming (EP)*:

EP was proposed by Lawrence J. Fogel [22], who used finite state machines as predictors and then evolved them. The basic structure of EP is described in Algorithm 3.3

Algorithm 3.3 Evolutionary programming

Generate initial population *randomly*

Compute the *fitness* of each individual

repeat

 Apply *mutation* to all elements of P to generate the offspring set O

 Compute the *fitness* of each individual in O

Select the best individuals for the following generation using
 $(\mu + \mu)$ -selection

until a stopping criterion is met

In EP, individuals are represented as real-valued vectors. The parent selection mechanism allows every individual in the population to generate one offspring using a mutation operator; no crossover mechanism is used. Finally, a (stochastic) tournament is used for the survival selection mechanism.

- *Genetic algorithms (GAs)*:

GAs were initially proposed by John Holland [26] for machine learning. The structure of a basic GA is described in Algorithm 3.4.

Traditionally, GAs use a binary encoding and use a fitness proportional selection mechanism. The main genetic operator for GAs is crossover while the mutation operator is considered as a secondary operator (i. e. , it is applied less often than crossover). It has been proved that a simple genetic algorithm requires elitism in order to guarantee convergence [45].

Algorithm 3.4 Genetic algorithm

Generate initial population *randomly*
 Compute the *fitness* of each individual
repeat
 Select a parent-set P
 Apply *crossover* to create offspring set O
 Apply *mutation* to the elements of O
 Compute the *fitness* of each individual in O
 Select the best individuals for the following generation
until a stopping criterion is met

3.3 MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Multi-objective evolutionary algorithms (MOEAs) share the basic structure of EAs. The main differences between them are that the fitness assignment mechanism of MOEAs must be able to deal with more than one objective function and that the elitist mechanism of MOEAs should handle sets of non-dominated solutions instead of only a single solution.

3.3.1 Elitism

Elitism is a mechanism which preserves the best solutions found during the search. However, in multi-objective optimization, the number of non-dominated solutions generated during the search tends to increase very quickly. Therefore, it is normally the case that some filtering mechanism is adopted in order to keep the size of the external file (in which the elitist solutions are maintained) bounded.

There are two main approaches to implement elitism in a MOEA:

1. The parent population and the offspring population are combined and a deterministic survivor selection mechanism is implemented to preserve the best solutions for the next generation.
2. An external population called *archive* is used to store the non-dominated solutions found during the search process.

3.3.2 Fitness assignment

Fitness assignment in MOEAs must deal with $k > 1$ objective functions while the selection mechanism expects a single fitness value. There are three main fitness assignment schemes available for MOEAs [61]:

- *Criterion-based:*

This approach chooses one of the objective functions during the selection stages of the MOEA. The fitness of an individual, or group of individuals, depends only on the value of the chosen objective.

For example, the *vector evaluated genetic algorithm* (VEGA) [46] divides the population into k equally sized subpopulations and each population optimizes one of the k objectives. The individuals are then shuffled after the selection procedure.

- *Aggregation-based:*

In this scheme, the objective functions are combined into a single parametrized objective function which will be optimized. The parameters of the combined objective function must be varied during the optimization process to generate a set of non-dominated solutions.

- *Pareto-based:*

This scheme was proposed by Goldberg [24]. In this approach, individuals in the population are compared according to the Pareto dominance relation.

Most MOEAs use this scheme for fitness assignment. These approaches can be categorized into two generations:

- *First generation:*

In their early days, MOEAs lacked the concept of elitism and normally used fitness sharing to produce well-distributed solutions sets. Some representative examples of first generation MOEAs are the *multi-objective genetic algorithm* (MOGA) [23], the *niched Pareto genetic algorithm* (NPGA) [27], and the *non-dominated sorting genetic algorithm* (NSGA) [49].

– *Second generation:*

These MOEAs incorporated an elitist mechanism for multi-objective optimization and adopted more elaborate diversity estimators (e. g. , based on clustering). Some representative second generation MOEAs are the *strength Pareto evolutionary algorithm* (SPEA) [61], the *Pareto archived evolution strategy* (PAES) [36], the *strength Pareto evolutionary algorithm 2* (SPEA2) [59], and the *non-dominated sorting genetic algorithm II* (NSGA-II) [15].

Recently, proposals for fitness assignment schemes based on quality indicators (see Chapter 4) have been made in the specialized literature. For example, the *indicator based evolutionary algorithm* (IBEA) [58] and the *S metric selection evolutionary multi-objective algorithm* (SMS-EMOA) [19]. Because of the relevance of this type of approach for this thesis, a full chapter (see Chapter 5) will be devoted to it.

QUALITY INDICATORS FOR MULTI-OBJECTIVE
OPTIMIZATION

Contents

4.1	Unary quality indicators	24
4.1.1	Properties of unary quality indicators	24
4.1.2	Discussion of some unary quality indicators	25
4.1.2.1	The hypervolume indicator	25
4.1.2.2	The GD, IGD, and Δ_p indicators	27
4.1.2.3	Spacing indicators	29
4.2	Binary quality indicators	30
4.2.1	Discussion of some binary quality indicators	30
4.2.1.1	The ϵ -indicators	30
4.2.1.2	The coverage indicator	32

Many [EAs](#) have been proposed for multi-objective optimization. Since [EAs](#) do not guarantee the optimality of their solution sets, it is of interest to compare their performance including both the quality of the outcome and the computational resources used by the [MOEAs](#) to generate such outcome.

The computational resources used by an [EA](#) are often measured either by counting the number of fitness evaluations performed or by comparing the CPU time consumed by the algorithm when run in a particular computer.

The quality of the outcome, however, is different for [SOPs](#) and [MOPs](#). In [SOPs](#), the quality of the solution is defined using the objective function: given a minimization problem, smaller values are better. In [MOPs](#), the Pareto dominance relation can be used to compare solutions even though they can be incomparable. However, comparing solution sets is more complicated since

the quality of the solution sets isn't clearly defined in this context [65]: it can consider closeness to the Pareto optimal solutions, closeness to the Pareto front, coverage of the Pareto front, etc.

In this chapter, we will review some quality indicators that have been proposed for assessing the performance of a MOEA.

4.1 UNARY QUALITY INDICATORS

Definition 9 (Unary quality indicator). *A unary quality indicator is a function $I_1 : \Psi \rightarrow \mathbb{R}$ where Ψ is the set of all Pareto set approximations for a given problem.*

The indicator I_1 should induce a total order among the outcome sets. Then, the outcome of two (or more) multi-objective optimizers can be compared through the indicator values.

4.1.1 Properties of unary quality indicators

According to Definition 9, any function from Ψ to \mathbb{R} can be considered an (unary) indicator. However, some properties are desirable for indicators to be useful [57]:

- *Monotonicity:*

An indicator I is strictly monotonic if and only if

$$\forall A, B \in \Psi \quad A \prec B \Rightarrow I(A) > I(B) \quad (4.1)$$

Strict monotonicity guarantees that an indicator I is compatible with the Pareto dominance relation.

- *Scaling invariance:*

An indicator I is scaling invariant if for any strictly monotonic function $s : \mathbb{R}^k \rightarrow \mathbb{R}^k$

$$\forall A \in \Psi \quad I(\vec{f}(A)) = I(s(\vec{f}(A))) \quad (4.2)$$

- *Computational effort:*

This property considers the computational resources needed to compute an indicator depending on the number of solutions in the Pareto set approximation and on the number of objectives in the MOP.

- *Additional problem knowledge:*

Many indicators need information in order to be applied such as: the Pareto optimal set, reference objective vectors, reference sets, etc.

4.1.2 Discussion of some unary quality indicators

4.1.2.1 The hypervolume indicator

The hypervolume indicator was proposed by Zitzler and Thiele [60]. It was originally defined as the size of the *space covered by the Pareto optimal solutions* [60]; equivalent definitions have been proposed based on the Lebesgue measure [21], on polytopes [5], and on the attainment function [55].

Definition 10 (Hypervolume indicator I_H). Let $A = (a_1, \dots, a_l)$ be a non-dominated set of decision vectors, $|a_i| = k$. The hypervolume indicator I_H gives the Lebesgue measure of the union of k -hyperrectangles defined by a_i and a fixed reference point $r = (r_1, \dots, r_l)$.

4.1.2.1.1 PROPERTIES OF I_H

- The hypervolume indicator is the only known unary quality indicator which guarantees strict monotonicity regarding the Pareto dominance relation [65, 21].

A set $A \in \mathbb{R}^l$ achieves maximum hypervolume value if and only if all points $a \in A$ are Pareto optimal [21].

- I_H can be either maximized (see Figure 4.1b) or minimized (see Figure 4.1a) but the reference point must be chosen accordingly:
 - The reference point should be smaller or equal to the ideal vector for hypervolume maximization

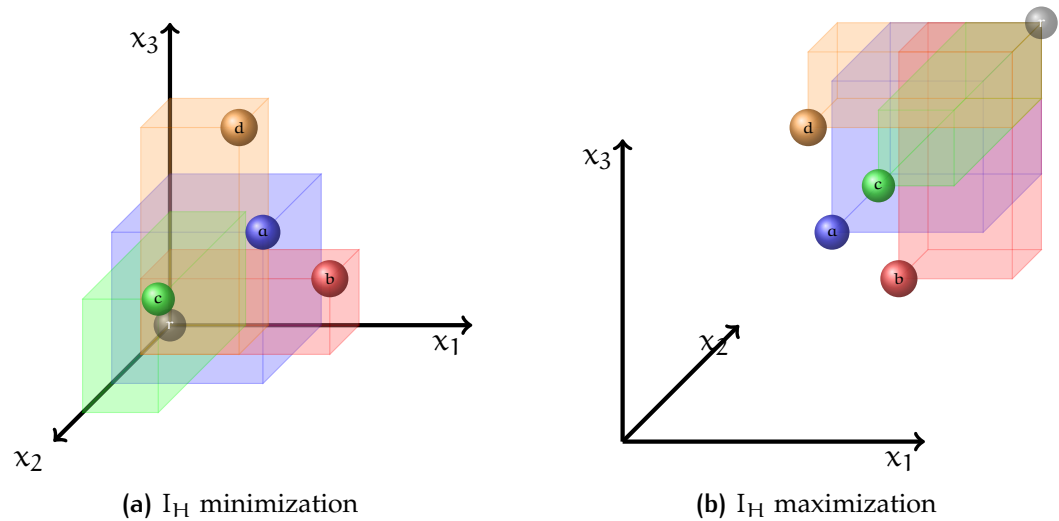


Figure 4.1.: I_H for a 3-objective MOP

- The reference point should be larger or equal to the nadir vector for hypervolume maximization
- This indicator is not scaling invariant [63]: it is sensitive to the choice of the reference point.
- Computation of the hypervolume indicator has been shown to be #P-hard [12]. The running time for calculating the hypervolume indicator is exponential in the number of objectives [54].
- The choice of the reference point can impact the outcome of the hypervolume indicator [35].
- The slope of the Pareto front determines how the points that maximize the hypervolume indicator are distributed [1].
- Preference information can be incorporated into the hypervolume indicator to place more emphasis on certain parts of the Pareto front than on others [55], maintaining the monotonicity property.

4.1.2.2 The GD, IGD, and Δ_p indicators

4.1.2.2.1 GENERATIONAL DISTANCE INDICATOR

Definition 11 (*Generational distance (GD) indicator*). The GD indicator represents how “far” a set A is from the Pareto front P . It is defined as [51]:

$$I_{GD} = \frac{\left(\sum_{i=1}^{|A|} d_i^p \right)^{\frac{1}{p}}}{|A|} \quad (4.3)$$

where d_i is the Euclidean distance (in objective function space) between a_i and the *nearest* member of P . Lower I_{GD} values represent “better” sets.

However, this indicator gets “better” when an element of the set is duplicated or if the set A contains small perturbations of a given element [48].

Definition 12 (*Alternative GD indicator*). An alternative version of GD was proposed by Schütze et al. [48] to avoid the previously described effect. It is defined as:

$$I_{GD_p} = \left(\frac{1}{|A|} \sum_{i=1}^{|A|} d_i^p \right)^{\frac{1}{p}} \quad (4.4)$$

where d_i is the Euclidean distance (in objective function space) between a_i and the *nearest* member of P .

4.1.2.2.2 INVERTED GENERATIONAL DISTANCE INDICATOR

Definition 13 (*Inverted generational distance (IGD) indicator*). The IGD indicator represents how “far” the Pareto front P is from a set A . It is defined as:

$$I_{IGD} = \frac{\left(\sum_{i=1}^{|P|} \tilde{d}_i^p \right)^{\frac{1}{p}}}{|P|} \quad (4.5)$$

where \tilde{d}_i is the Euclidean distance (in objective function space) between p_i and the *nearest* member of A . Lower I_{IGD} values represent “better” sets.

However, the indicator gets “better” when using a finer discretization of the Pareto front [48].

Definition 14 (Alternative IGD indicator). An alternative version of IGD was proposed by Schütze et al. [48] to avoid the previously described effect. It is defined as:

$$I_{IGD_p} = \left(\frac{1}{|P|} \sum_{i=1}^{|P|} \tilde{d}_i^p \right)^{\frac{1}{p}} \quad (4.6)$$

where \tilde{d}_i is the Euclidean distance (in objective function space) between p_i and the *nearest* member of A .

4.1.2.2.3 PROPERTIES OF I_{GD_p} AND I_{IGD_p}

- I_{GD_p} and I_{IGD_p} have (weak) metric properties [48]:
 - I_{GD_p} and I_{IGD_p} are non-negative. They aren't positive since $I_{GD_p} = 0 \iff A \subseteq P$ and $I_{IGD_p} = 0 \iff P \subseteq A$
 - I_{GD_p} and I_{IGD_p} are non-symmetric.
 - I_{GD_p} and I_{IGD_p} don't satisfy the (relaxed) triangle inequality.
- I_{GD_p} and I_{IGD_p} need knowledge of the true Pareto front of the problem [34].

4.1.2.2.4 Δ_p INDICATOR

Definition 15 (Hausdorff distance (d_H)). The Hausdorff distance represents how “far” two sets A, B are from each other. It is defined as:

$$\text{dist}(u, A) = \inf_{v \in A} \|u - v\| \quad (4.7)$$

$$\text{dist}(B, A) = \sup_{u \in B} \text{dist}(u, A) \quad (4.8)$$

$$d_H = \max(\text{dist}(A, B), \text{dist}(B, A)) \quad (4.9)$$

$$(4.10)$$

The Hausdorff distance is a metric [48]: It is positive, symmetric, and satisfies the triangle inequality.

Definition 16 (Δ_p indicator). The Δ_p indicator represents the *averaged Hausdorff distance* between a set A and the Pareto front P . It is defined as [48]:

$$I_{\Delta_p} = \max (I_{GD_p}, I_{IGD_p}) \quad (4.11)$$

4.1.2.2.5 PROPERTIES OF I_{Δ_p}

- I_{Δ_p} has metric properties [48]:
 - It is positive and symmetric: I_{Δ_p} is a semi-metric
 - If the magnitudes of the sets are bounded; the relaxed triangle inequality is satisfied and I_{Δ_p} is a pseudo-metric.
- If $p = \infty$, then I_{Δ_p} is equal to d_H

The proofs for the properties of I_{Δ_p} can be found in [48].

4.1.2.3 Spacing indicators

Several spacing indicators have been proposed which attempt to measure how well the solutions in a non-dominated set are distributed in the objective function space.

Definition 17 (Schott's spacing indicator (I_{SS})). Schott's spacing indicator measures the range variance of neighboring vectors in a non-dominated set A . It is defined as [47]:

$$I_{SS} = \sqrt{\frac{1}{|A|-1} \sum_{i+1}^{|A|} (\bar{d} - d_i)^2} \quad (4.12)$$

where $d_i = \min_{j \in \{1, \dots, |A|\}} \sum_{i=1}^k |f_i(\vec{a}_i) - f_i(\vec{a}_j)|$ and \bar{d} is the mean of all d_i

A similar indicator was proposed by Srinivas and Deb in [49].

4.1.2.3.1 PROPERTIES OF SPACING INDICATORS

- Generally, spacing indicators are not monotonic regarding the Pareto dominance relation.
- Spacing indicators are not scaling invariant
- In most cases, the runtime complexity of the indicators is quadratic in the cardinality of the non-dominated set.

4.2 BINARY QUALITY INDICATORS

Definition 18 (Binary quality indicator). A *binary quality indicator* is a function $I_2 : \Psi \times \Psi \rightarrow \mathbb{R}$ where Ψ is the set of all Pareto set approximations for a given problem.

There is a close relationship between unary and binary quality indicators. On the one hand, every unary indicator I_1 can be transformed into a binary indicator I_2 as follows [62]:

$$I_2(A, B) = I_1(B) - I_1(A) \quad A, B \in \Psi \quad (4.13)$$

On the other hand, every binary indicator I_2 can be transformed into an unary indicator I_1 by using a reference set R [62]:

$$I_1(A) = I_2(A, R) \quad A \in \Psi \quad (4.14)$$

4.2.1 Discussion of some binary quality indicators

4.2.1.1 *The ϵ -indicators*

The ϵ -indicators were introduced in [65] based on multiplicative ϵ -dominance and additive ϵ -dominance.

4.2.1.1.1 MULTIPLICATIVE ϵ -INDICATOR (I_{ϵ^*})

Definition 19 (multiplicative ϵ -dominance). Let $a, b \in \mathbb{R}^n$; a multiplicatively ϵ -dominates b ($a \preceq_{\epsilon^*} B$) if and only if

$$\forall i \in \{1, \dots, k\} \quad f_i(a) \leq \epsilon f_i(b) \quad (4.15)$$

Definition 20 (binary multiplicative ϵ -indicator). Let $A, B \in \mathbb{R}^{1 \times k}$ non-dominated sets of decision vectors. The binary multiplicative ϵ -indicator I_{ϵ^*} is defined as

$$I_{\epsilon^*}(A, B) = \max_{b \in B} \min_{a \in A} \max_{1 \leq i \leq n} \frac{f_i(a)}{f_i(b)} \quad (4.16)$$

 4.2.1.1.2 ADDITIVE ϵ -INDICATOR ($I_{\epsilon+}$)

Definition 21 (additive ϵ -dominance). Let $a, b \in \mathbb{R}^n$; a additively ϵ -dominates b ($a \preceq_{\epsilon+} B$) if and only if

$$\forall i \in \{1, \dots, k\} \quad f_i(a) \leq \epsilon + f_i(b) \quad (4.17)$$

Definition 22 (binary additive ϵ -indicator). Let $A, B \in \mathbb{R}^{1 \times k}$ non-dominated sets of decision vectors. The binary additive ϵ -indicator $I_{\epsilon+}$ is defined as

$$I_{\epsilon+}(A, B) = \max_{b \in B} \min_{a \in A} \max_{1 \leq i \leq n} f_i(a) - f_i(b) \quad (4.18)$$

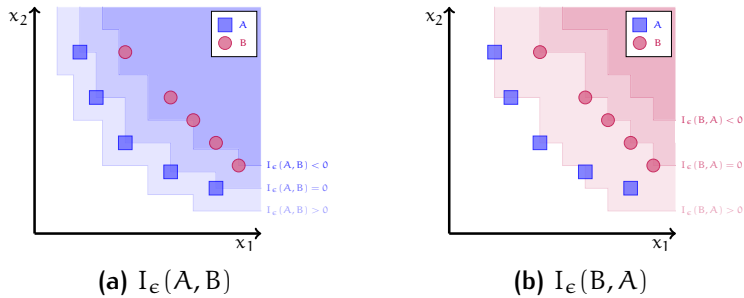


Figure 4.2.: ϵ -indicators for a bi-objective MOP

4.2.1.1.3 PROPERTIES OF THE ϵ INDICATORS

- Both I_{ϵ^*} and I_{ϵ^+} are monotonic but not strictly monotonic regarding the Pareto dominance relation [57].
- The runtime complexity of the indicator is of order $\mathcal{O}(k \cdot |A| \cdot |B|)$ [57].
- I_{ϵ^*} and I_{ϵ^+} are minimized: a set A is better than another set B if $I_{\epsilon}(A, B) < I_{\epsilon}(B, A)$ (see Figure 4.2)

4.2.1.2 *The coverage indicator*

The coverage indicator was proposed by Zitzler and Thiele [60].

Definition 23 (Coverage indicator (I_C)). Let $A, B \in \mathbb{R}^{l \times k}$ non-dominated sets of decision vectors. I_C maps the ordered pair (A, B) to the interval $[0, 1]$ as follows:

$$I_C(A, B) = \frac{|\{b \in B \mid \exists a \in A : a \preceq b\}|}{|B|} \quad (4.19)$$

The unary version of this indicator is called *proportion of Pareto-optimal objective vectors found* and was introduced by Ulungu et al. [50].

4.2.1.2.1 PROPERTIES OF THE COVERAGE INDICATOR

- This indicator is not symmetric: $I_C(A, B)$ may be different from $1 - I_C(B, A)$. Both $I_C(A, B)$ and $I_C(B, A)$ have to be considered to compare the outcome sets of MOEAs [60].
- I_C is strictly monotonic regarding the Pareto dominance relation [57].
- The runtime complexity of the indicator is of order $\mathcal{O}(|A| \cdot |B|)$ [57].
- I_C is scaling invariant [57]

INDICATOR-BASED MULTI-OBJECTIVE SEARCH

Contents

5.1	Indicator-based archiving algorithms	34
5.1.1	LAHC	34
5.1.2	ESP	35
5.2	Indicator-based selection	36
5.2.1	IBEA	36
5.2.2	SMS-EMOA	38
5.2.3	Iterative approach to indicator-based multi-objective optimization	40
5.2.4	MO-CMA-ES	41
5.2.5	DF-SMS-EMOA	43
5.3	Set-oriented multi-objective evolutionary algorithms	44
5.3.1	SPAM	44
5.3.2	SPAM ⁺	45
5.4	Closing remarks	46

Several indicators have been proposed to evaluate the outcome set of different [MOEAs](#). Since the quality of an outcome set can be evaluated by a unary quality indicator, it seems reasonable to recast a given [MOP](#) to the optimization of the chosen indicator.

There have been three main approaches to integrate an indicator into a [MOEA](#):

1. The indicator is integrated into the environmental selection process in an archive (Section [5.1](#))

2. The indicator is integrated directly into the selection mechanism of the MOEA (Section 5.2)
3. The indicator is integrated as a set preference relation for set problems (Section 5.3)

5.1 INDICATOR-BASED ARCHIVING ALGORITHMS

Many MOEAs use an external population, often called archive, to store interesting points found during their search. One possible approach for handling the archive is to store all non-dominated solutions [20] but this is often undesirable since:

- true Pareto fronts can be infinitely large
- there is a large computational overhead for maintaining such archives

An alternative is to store only a subset of the non-dominated points (containing at most N points). However, bounded archiving is non-trivial since several properties of the archive are desirable such as [37]:

- The archive should be full
- The archive should approximate its input sequence (the optimal points generated by an EA) completely

One possible solution to this problem is to use an indicator to define which subset of the (known) non-dominated points to store.

In the following, a brief description of some indicator-based archiving algorithms is presented:

5.1.1 *Lebesgue archiving hill-climber* (LAHC)

Knowles and Corne proposed the LAHC [37] (also referred to as S metric archiving [35]) which was the first indicator-based archiving algorithm. In

the algorithm, the bounded archive is limited to knowing at most $N+1$ solutions at any given time: the solutions currently stored in the archive (at most N) and 1 candidate solution.

The current candidate solution is accepted into the archive if at least one of the following conditions holds:

- The archive isn't full
- The new point dominates at least one point in the archive (the dominated points are then removed from the archive)
- The new point is non-dominated with respect to all other points in the archive and replacing one of the archived points in the archive by the new point improves the hypervolume indicator

This archive works as a steepest-ascent hill-climber in the space of the hypervolumes of the dominated regions by the archive.

5.1.2 *Evolution strategy with probabilistic mutation (ESP)*

ESP was presented by Huband et al. [28]. It is a $(\mu + \lambda)$ *evolution strategy (ES)* with two modifications:

- a genetic algorithm style mutation probability setting:
In *ESP*, individuals are mutated based on a probability p instead of always. The authors report “excellent” results with $p = 1/m$ (where m is the number of variables in the problem)
- a hypervolume-based measure for the truncation of the secondary population:

The *ESP* uses a modified version of the truncation mechanism in the *SPEA2*. The main modification was the replacement of the Euclidean distance-based nearest neighbor measure by a hypervolume-based intra-ranking mechanism.

In this intra-ranking mechanism, equally-ranked (according to Pareto dominance) individuals can be compared using the hypervolume of

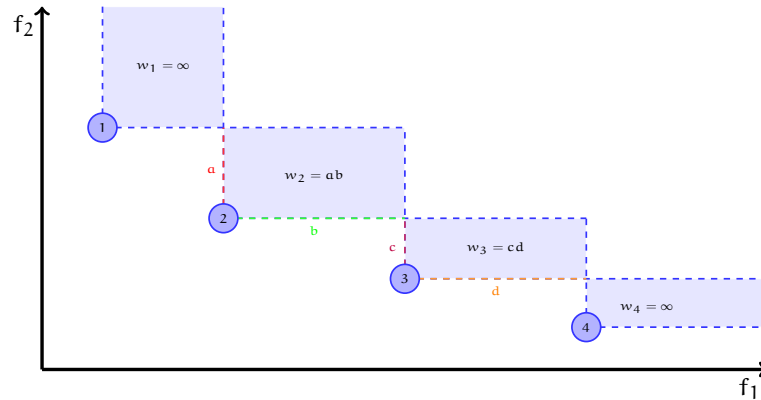


Figure 5.1.: Hypervolume measure used by ESP on a bi-objective MOP

the region dominated with respect to the hypercube formed by the next worse individuals in each dimension. An example of the hypervolume assignment for a bi-objective optimization problem is presented in Figure 5.1. The individuals with the lowest hypervolume values are eliminated in order for the archive bound to be fulfilled.

5.2 INDICATOR-BASED SELECTION

The goal of a MOEA is to find a “good” approximation of the Pareto optimal set. The quality of an approximation set is often evaluated through indicators in order to compare outcomes of different MOEAs. Assuming the existence of an indicator that provides a good ordering amongst the sets of possible Pareto approximations, a MOEA can attempt to maximize the indicator for a given MOP through appropriate selection mechanisms.

In the following, some MOEAs with indicator-based selection mechanisms are briefly described.

5.2.1 *Indicator based evolutionary algorithm (IBEA)*

The first algorithm to use indicators in the selection mechanism was IBEA, proposed by Zitzler and Künzli [58].

IBEA replaces the original MOP by

$$\min I(A, S)$$

where:

I is a dominance preserving binary quality indicator

A is a Pareto set approximation

S is the Pareto optimal set for the original MOP

Since I is assumed to be dominance preserving, $I(A, S)$ should be minimal if $A = S$.

The fitness of an individual x_1 in the population P is assigned as follows:

$$F(x_1) = \sum_{x_2 \in P \setminus \{x_1\}} -e^{-I(x_2, x_1)/\kappa} \quad \text{where } \kappa > 0$$

This fitness assignment scheme is compliant if I is dominance preserving with the influence of individuals with small indicator values to the overall fitness being much larger than the influence of larger values.

The authors presented IBEA (and some variants) with two indicators

- the binary additive ϵ -indicator (see Section 4.2.1.1)

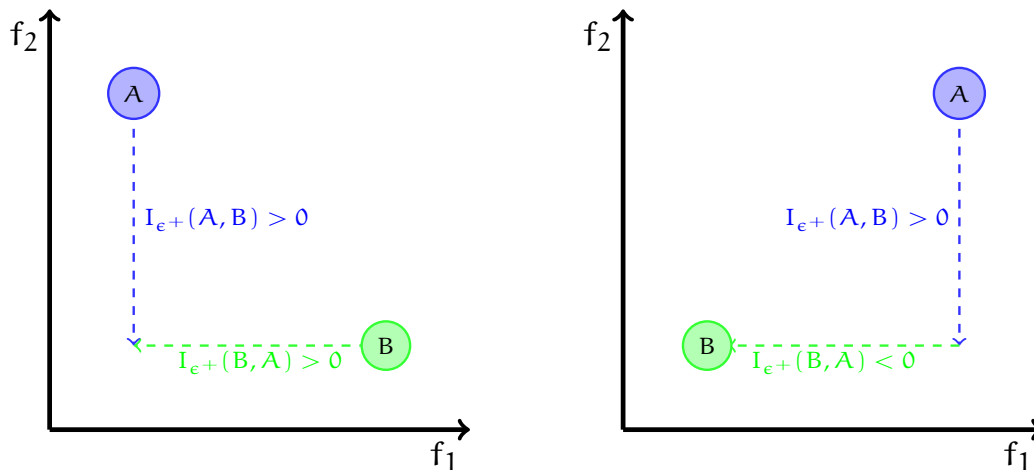


Figure 5.2.: Binary additive ϵ -indicator used by IBEA for bi-objective MOP

- the binary hypervolume indicator

Given two sets A, B , the binary hypervolume indicator $I_H(A, B)$ measures the hypervolume of the space that is dominated by B but not by A with respect to a given reference point. It is defined using the unary hypervolume indicator (see Section 4.1.2.1) as:

$$I_H(A, B) = I_H(A \cup B) - I_H(A) \quad (5.1)$$

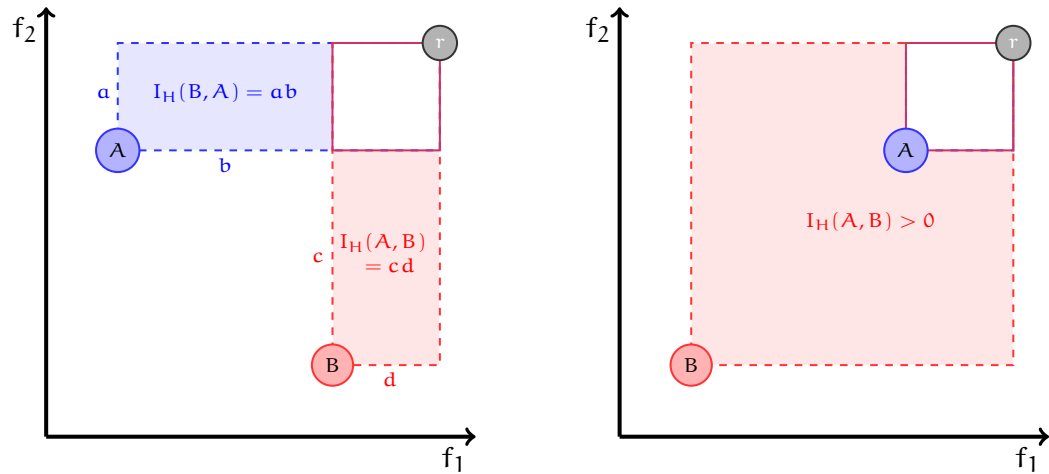


Figure 5.3.: Binary hypervolume-indicator used by *IBEA* for bi-objective MOPs

This algorithm was later extended by Basseur and Burke in [4] to the *indicator-based multi-objective local search (IBMOLS)* by replacing the variation operators of *IBEA* with a local search operator .

5.2.2 *S metric selection evolutionary multi-objective algorithm (SMS-EMOA)*

The *SMS-EMOA* was proposed by Emmerich, Beume, and Naujoks in [19]. This algorithm is a steady state $(\mu + 1)$ -*MOEA* inspired by the *NSGA-II* [15] and the *LAHC* [35, 37]. The *SMS-EMOA* generates only one new individual per iteration which joins the population only if its addition (by replacing another individual) improves the hypervolume covered by the population.

The basic algorithm is described in Algorithm 5.1.

Algorithm 5.1 Basic SMS-EMOA

```

1: Initialize the population  $P_0$  randomly
2:  $i \leftarrow 0$ 
3: repeat
4:   Generate one new individual  $x$ 
5:    $Q \leftarrow P_i \cup \{x\}$ 
6:   Compute the Pareto front rank through non-dominated sorting of the
     individuals in  $Q$ 
7:    $W \leftarrow$  individuals in the worst-ranked front
8:    $r \leftarrow$  individual (in  $W$ ) which least contributes to the hypervolume
     covered by  $W$ .
9:    $P_{i+1} \leftarrow Q \setminus r$ 
10:   $i \leftarrow i + 1$ 
11: until stopping criteria is met
12: return  $P_i$ 

```

The same authors also proposed 3 variants of the SMS-EMOA in [41].

1. *Non-dominated sorting (NDS) + number of dominated points (DP) or hypervolume indicator (HV)*

The basic algorithm attempts to obtain a good distribution of the points for the point on each of the fronts identified when this is only important to the first (non-dominated) front. One alternative is to favor individuals in the dominated fronts that are in sparsely populated areas by removing the individual that is dominated by the largest number of points.

This approach improves the computational cost of the average case since the hypervolume indicator is only computed when all individuals in the population are non-dominated.

2. *Pareto dominance (PD) or HV*

This variant uses a bounded population (instead of it being constant-sized). In each iteration, all dominated individuals are discarded. If the maximum size for the population is exceeded, then the individual which contributes the least to the hypervolume indicator is rejected.

3. 2 phase approach: PD and NDS+HV

This approach combines the approaches from point (2) above and the

basic SMS-EMOA. It starts with variant (2) until the population is full and then continues with the basic algorithm until the stopping criterion is reached.

None of the variants significantly improved the quality of the solution obtained, but approach (1) reduces the computational cost of the average case [6, 8].

In another paper, Beume et al. [7] further analyzed the effects of the τ -greedy S-metric-selection performed by the SMS-EMOA. They presented some “extreme” cases in which this selection scheme fails (often with very small population sizes) but also showed that the scheme can detect the optimal distribution for most connected continuous Pareto front types.

Several improvements have been proposed for the SMS-EMOA such as:

- Bradstreet, Barone and While proposed a greedy generalization of the reduction scheme of the SMS-EMOA to allow the joining of more than τ individual in the population [9]. They further improved their greedy approach in [10] by taking advantage of the *incremental hypervolume by slicing objectives* (IHSO) algorithm [11] to compute the hypervolume contributions of the individuals in the population.
- Ishibuchi, Tsukamoto, Sakane and Nojima proposed to replace the hypervolume computation with a hypervolume approximation by achievement scalarizing functions to decrease (significantly) the computation time of the SMS-EMOA [32].

5.2.3 Iterative approach to indicator-based multi-objective optimization

The iterative approach to indicator-based multi-objective optimization was proposed by Ishibuchi, Tsukamoto, and Nojima in [31]. Unlike most MOEAs, the iterative approach generates only one solution from each run; therefore, several runs are required to find a solution set.

The iterative approach consists of two phases:

1. During the first k runs, each of the k objectives is optimized independently by a standard genetic algorithm (or another optimization strat-

egy) to obtain k extreme solutions which will initialize the solution set.

2. Afterwards, the iterative approach searches for the solution x which most contributes to the hypervolume of the solution set obtained by previous runs. The initial population of the iterative approach is generated from the individuals in the solution set instead of randomly.

The iterative approach was proposed to find only a small number of representative non-dominated solutions of a **MOP** as an alternative to most **MOEAs** which require large populations and output a large number of non-dominated solutions.

5.2.4 *Multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES)*

The **MO-CMA-ES** was proposed by Igel, Hansen, and Roth in [29] as a multi-objective variant of the *covariance matrix adaptation evolution strategy (CMA-ES)* [25]. This approach applies a hypervolume-based selection mechanism to a population of elitist $(1 + \lambda)$ -**CMA-ES**.

The elitist $(1 + \lambda)$ -**CMA-ES** used for the **MO-CMA-ES** samples the λ new candidate solutions and then updates the parent solution p depending on whether any of the new solutions is better than p . Afterwards, the step size is updated according to the success rate (defined as the proportion of the λ new solutions which were better than p). If p was replaced by any new solution, then the covariance matrix is updated as in the $(1, \lambda)$ -**CMA-ES** proposed in [25]. The elitist $(1 + \lambda)$ -**CMA-ES** is described in Algorithm 5.2

The $\lambda_{MO} \times (1 + \lambda)$ -**MO-CMA-ES** keeps a population of λ_{MO} elitist $(1 + \lambda)$ -**CMA-ES** where an individual x is considered better than another individual y if:

- x has a lower rank than y (according to Pareto ranking) or
- x and y have the same rank and x contributes the most to the covered hypervolume

Algorithm 5.2 $(1 + \lambda)$ -CMA-ES

```

1:  $g \leftarrow 0$ 
2: Initialize parent  $p^0$ 
3: repeat
4:   for  $k = 1, \dots, \lambda$  do
5:      $x_k^{g+1} \sim \mathcal{N}(m, C)$  ( $x_k^{g+1}$  is a random variable distributed according to
      a multi-variate normal distribution with mean vector  $m$  and covari-
      ance matrix  $C$ )
6:   end for
7:   Update the step size
8:   if  $f(x_{\text{best}}^{g+1}) \leq f(p^g)$  then
9:      $p^{g+1} \leftarrow x_{\text{best}}^{g+1}$ 
10:    Update the covariance matrix
11:   end if
12:    $g \leftarrow g + 1$ 
13: until stopping criteria is met
14: return  $p_g$ 

```

Algorithm 5.3 $\lambda_{\text{MO}} \times (1 + \lambda)$ -MO-CMA-ES

```

1:  $g \leftarrow 0$ 
2: for  $i = 1, \dots, \lambda_{\text{MO}}$  do
3:   Initialize parent  $p_i^0$ 
4: end for
5: repeat
6:   for  $j = 1, \dots, \lambda_{\text{MO}}$  do
7:     for  $k = 1, \dots, \lambda$  do
8:        $x_{i,k}^{g+1} \sim \mathcal{N}(m, C)$ 
9:     end for
10:   end for
11:    $Q_g = \{p_i^g, x_{i,k}^{g+1} \mid 1 \leq i \leq \lambda_{\text{MO}}, 1 \leq k \leq \lambda\}$ 
12:   for  $j = 1, \dots, \lambda_{\text{MO}}$  do
13:     Update the step size for the parent of the  $j$ -th CMA-ES
14:     Update the step size for the  $\lambda$  offspring of the  $j$ -th CMA-ES
15:     Update the covariance matrix of the  $\lambda$  offspring of the  $j$ -th CMA-ES
16:   end for
17:    $p_{g+1}^{1, \dots, \lambda_{\text{MO}}} \leftarrow$  the best  $\lambda_{\text{MO}}$  individuals in  $Q_g$ 
18:    $g \leftarrow g + 1$ 
19: until stopping criteria is met
20: return  $p_g^{1, \dots, \lambda_{\text{MO}}}$ 

```

The $\lambda_{MO} \times (1 + \lambda)$ -MO-CMA-ES is described in Algorithm 5.3.

Igel, Suttorp, and Hansen proposed some improvements to the update process of the covariance matrices as well as some steady-state selection schemes for the MO-CMA-ES in [30].

Voss, Beume, Rudolph, and Igel compared the outcomes of scalarization methods with the MO-CMA-ES and a hybrid MO-CMA-ES (with a scalarization method) in bi-objective optimization problems [52]. The MO-CMA-ES outperformed all scalarization and hybrid approaches on most standard benchmark functions.

5.2.5 *Desirability function SMS-EMOA (DF-SMS-EMOA)*

Since MOPs have a set of optimal solutions, MOEAs are required to approximate sets instead of a single optimal solution. This implies a higher computational cost for MOEAs which can render them inapplicable for expensive problems.

The DF-SMS-EMOA, proposed by Wagner and Trautmann [53], uses *desirability functions* (DFs), defined by the decision maker, to restrict its outcome set to regions that are of practical interest. In DFs, each objective is transformed into a desirability in $[0, 1]$ which are combined in a *desirability index* (DI) also in $[0, 1]$.

This approach uses Harrington's one-sided DF and maximizes the DFs for the outcome set with the hypervolume-based SMS-EMOA. The DF-SMS-EMOA is described in Algorithm 5.4

The computational complexity of this approach is rather high because of the required hypervolume calculations. Trautmann and Wagner propose an improvement to the approach by replacing the hypervolume indicator with a DI.

Algorithm 5.4 DF-SMS-EMOA

Require: DFs for each objective $\vec{d} : (Y_i^{(1)}, d_i^{(1)}), (Y_i^{(2)}, d_i^{(2)})$, $i = 1, \dots, k$

- 1: Initialize population P randomly and evaluate P
- 2: **repeat**
- 3: Reset DFs according to \vec{d}
- 4: Selection and crossover of the parents.
- 5: Mutation and evaluation of the offspring
- 6: **for** $j = 1, \dots, k$ **do**
- 7: $y_{\text{med},j} = \text{median}(\vec{f}_j)$
- 8: **if** $d_j(y_{\text{med},j}) < \epsilon$ **then**
- 9: $Y_j^{(2)} \leftarrow y_{\text{med},j}$ $d_j^{(2)} \leftarrow \epsilon$
- 10: **end if**
- 11: $f_j \leftarrow -d_j(\vec{f}_j)$
- 12: **end for**
- 13: Identify the last front F using non-dominated sorting
- 14: $HV_i \leftarrow$ hypervolume of the space covered by $\{F \setminus \vec{f}_i\}$, $i = 1, \dots, m$
- 15: Reject individual with the highest HV_i
- 16: **until** stopping criteria is met
- 17: **return** P

5.3 SET-ORIENTED MULTI-OBJECTIVE EVOLUTION-ARY ALGORITHMS

A MOP can be viewed as a set problem where the search space consists of all possible Pareto set approximations. A classical MOEA, from this viewpoint, would be a (1,1)-ES on the set problem:

- The population p represents the current Pareto set approximation
- A new Pareto set approximation p' is generated by applying some genetic operators (such as selection, recombination, and mutation) which, hopefully, is better than p

5.3.1 Set preference algorithm for multi-objective optimization (SPAM)

Zitzler, Thiele and Bader proposed SPAM [63, 62, 3, 64], a (1+1)-ES on the set problem which is independent of the indicator used to guide the search.

SPAM generates two offspring sets from a parent set P :

1. P' is generated using a random mutation operator on P . This operator should be able to generate any Pareto approximation in the search space.
2. P'' is generated using a heuristic mutation on P . This operator attempts to generate a set which is better than P by mimicking genetic operators in a classical **MOEA**.

P is replaced only if any of the offspring sets is weakly preferable (\preceq) to P according to the indicator used. The basic **SPAM** is described in Algorithm 5.5.

Algorithm 5.5 Basic **SPAM**

```

Initialize P randomly
repeat
  P' ← randomMutation(P)
  P'' ← heuristicMutation(P)
  if P''  $\preceq$  P then
    P ← P''
  else if P'  $\preceq$  P then
    P ← P'
  end if
until stopping criteria is met
return P

```

5.3.2 **SPAM**⁺

SPAM⁺ is a set-oriented **MOEA** proposed by Bader, Brockhoff, Welten, and Zitzler [2, 3]. It is a $(\mu + \lambda)$ -**ES** on the set problem. A general description of **SPAM**⁺ is presented in Algorithm 5.6

SPAM⁺ required the definition of the mutation, recombination, and selection operators:

- Set mutation

This corresponds to a run of a normal indicator-based **MOEA**

Algorithm 5.6 SPAM⁺

```

Initialize population S as  $\mu$  sets of N solutions
repeat
   $M \leftarrow \emptyset$ 
  for all  $A \in S$  do
     $M \leftarrow M \cup \{\text{setMutate}(A)\}$ 
  end for
   $M' \leftarrow \text{matingSelection}(M, \lambda)$ 
   $M'' \leftarrow \emptyset$ 
  for all  $(A_p, A_q) \in M'$  do
     $M'' \leftarrow M'' \cup \{\text{setRecombine}(A_p, A_q)\}$ 
  end for
   $S \leftarrow \text{environmentalSelection}(S, M'')$ 
until stopping criteria is met
return P

```

- Set recombination
This operator iteratively replaces the worst solution in the first parent with the best solution in the second parent until no indicator improvement is possible
- Mating and environmental selection
Four variants of mating and environmental selection were presented [2]

5.4 CLOSING REMARKS

To date, most indicator-based MOEAs rely on the hypervolume indicator, even though it has been shown that its computation is #P-complete [12] and its approximation is NP-hard [13], because it has been proven to be strictly monotonic with respect to Pareto dominance [34, 21, 65].

A different approach is to use indicators with metric properties, such as the Δ_p indicator (see Section 4.1.2.2) which, in spite of not being Pareto compliant, has other properties which can be exploited by a MOEA.

OUR PROPOSED APPROACH

Contents

6.1	Differential evolution	48
6.2	Fitness assignment	49
6.2.1	Individual contribution to the GD_p indicator	49
6.2.2	Individual contribution to the IGD_p indicator	50
6.2.3	Individual contribution to the Δ_p indicator	51
6.3	Reference set construction	51
6.4	The algorithm	53

Our aim was to design a [MOEA](#) which minimizes the Δ_p indicator (see Section [4.1.2.2](#)). In our proposed approach, the [MOP](#) is recast as:

$$\min I_{\Delta_p}(P) \quad (6.1)$$

where P is the population of used by the *differential evolution (DE)* algorithm where:

- individual fitness is determined according to an individual's contribution to the indicator
- the survival selection is replaced with a Δ_p -based mechanism
- the outcome of the algorithm is a set instead of an individual

6.1 DIFFERENTIAL EVOLUTION

Differential evolution (DE) was proposed by Rainer Storn and Kenneth Price [43] to solve continuous *SOPs*. The algorithm receives the following inputs: *population size (NP)*, *number of variables of the problem (D)*, *differential weight ($F \in [0, 2]$)*, and *crossover probability ($CR \in [0, 1]$)*.

Different *DE* strategies exist which can be described as *DE/x/y/z* where

- *x* specifies which individual will be mutated (*a*). It can be *rand* (*a* is chosen randomly) or *best* (*a* is the best individual in the population).
- *y* defines the number of difference vectors used.
- *z* defines the crossover scheme. It can be *bin* (for binomial crossover) or *exp* (for exponential crossover).

Our proposed approach uses the *DE/rand/1/bin* strategy (see Algorithm 6.1); however, any other *DE* strategy could be used.

Algorithm 6.1 *DE/rand/1/bin*

Generate initial population of *NP* individuals *randomly*

Evaluate the objective function *G* for each individual

repeat

for each individual \vec{x} in the population **do**

 Pick three individuals $\vec{a}, \vec{b}, \vec{c}$ distinct from each other.

 Pick an index $R \in \{1, \dots, D\}$

for $i = 1, \dots, D$ **do**

 Pick $r_i \in (0, 1)$ uniformly at random

if $i = R$ or $r_i < CR$ **then**

$y_i = a_i + F(b_i - c_i)$

else

$y_i = x_i$

end if

end for

if $G(y) < G(x)$ **then**

 Replace *x* with *y* in the population

end if

end for

until a stopping criterion is met

return the “best” individual in the population

6.2 FITNESS ASSIGNMENT

As previously stated, our aim is to minimize the Δ_p indicator

$$I_{\Delta_p} = \max(I_{GD_p}, I_{IGD_p})$$

which implies the simultaneous minimization of the GD_p and the IGD_p indicators.

Therefore, the fitness assignment of an individual must consider each individual's contribution to I_{GD_p} and I_{IGD_p} to properly reflect its contribution to I_{Δ_p} .

6.2.1 Individual contribution to the GD_p indicator

Since the GD_p indicator takes the power mean of the Euclidean distances between the elements of a set A and the Pareto front P , an individual's contribution must reflect its distance to P .

Definition 24 (Contribution of an individual to I_{GD}). The contribution of an individual α_i to the indicator (I_{GD}^i) can be defined in a straightforward manner as:

$$I_{GD}^i = \text{dist}(\alpha_i, P) \quad (6.2)$$

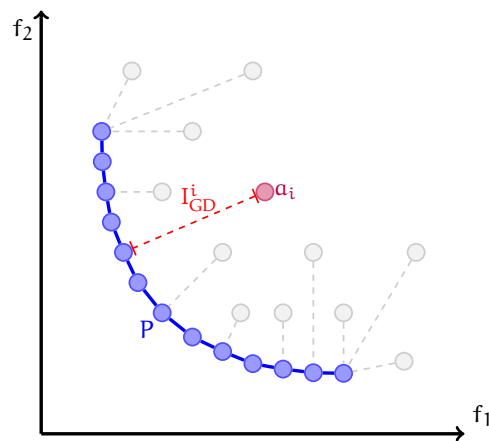


Figure 6.1.: Individual I_{GD_p} contribution

The elements of A with lower I_{GD}^i values are preferable to those with higher I_{GD}^i values (see Figure 6.1).

6.2.2 Individual contribution to the IGD_p indicator

The IGD_p indicator takes the power mean of the Euclidean distances between the elements of the Pareto front P and a set A .

Since I_{IGD_p} considers only the elements of A which are the closest to at least one element of P , some elements of A may not contribute to the value of this indicator (see Figure 6.2a) while other elements may have several contributions (see Figure 6.2b).

Definition 25 (Contribution of an individual to I_{IGD}). Let $a_i \in A$ and Q be the set of all elements of the Pareto front P for which a_i is the closest element in A . The contribution of a_i to the indicator (I_{IGD}^i) is then defined as:

$$I_{IGD}^i = \begin{cases} \sqrt[p]{\sum_{q \in Q} \text{dist}(q, a_i)^p} & Q \neq \emptyset \\ -1 & Q = \emptyset \end{cases} \quad (6.3)$$

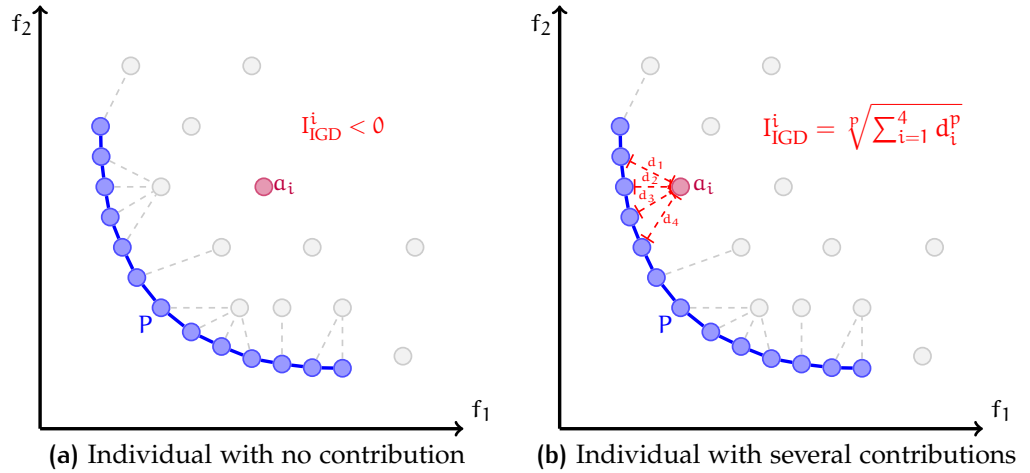


Figure 6.2.: Individual I_{IGD_p} contributions

Clearly, the elements that contribute to I_{IGD_p} are preferable to those that do not (see Figure 6.2). Also, elements with higher I_{IGD}^i values are preferable

to those with lower I_{IGD}^i values since they often cover larger sections of the Pareto front P .

6.2.3 Individual contribution to the Δ_p indicator

An individual's contribution to the Δ_p indicator must consider its contributions to both the GD and the IGD indicators.

Definition 26 (Contribution of an individual to I_{Δ_p}). Let $a_i, a_j \in A$ and the Pareto front P . Then, a_i contributes less to I_{Δ_p} than a_j if one of the following conditions holds:

1. a_i contributes to I_{IGD} and a_j doesn't

$$I_{IGD}^i > 0 \quad \text{and} \quad I_{IGD}^j < 0 \quad (6.4)$$

2. a_i contributes more to I_{IGD} than a_j

$$I_{IGD}^i > I_{IGD}^j \quad (6.5)$$

3. a_i and a_j contribute equally to I_{IGD} but a_i is closer to the Pareto front

$$I_{IGD}^i = I_{IGD}^j \quad \text{and} \quad I_{GD}^i < I_{GD}^j \quad (6.6)$$

In our proposed approach, an individual's contribution to the IGD indicator is given more importance than its contribution to the GD indicator since I_{IGD} reflects an individual's importance in terms of both convergence and coverage of the Pareto front (see Figure 6.3a) while I_{GD} only reflects its contribution to convergence (see Figure 6.3b).

6.3 REFERENCE SET CONSTRUCTION

The Δ_p indicator needs two sets:

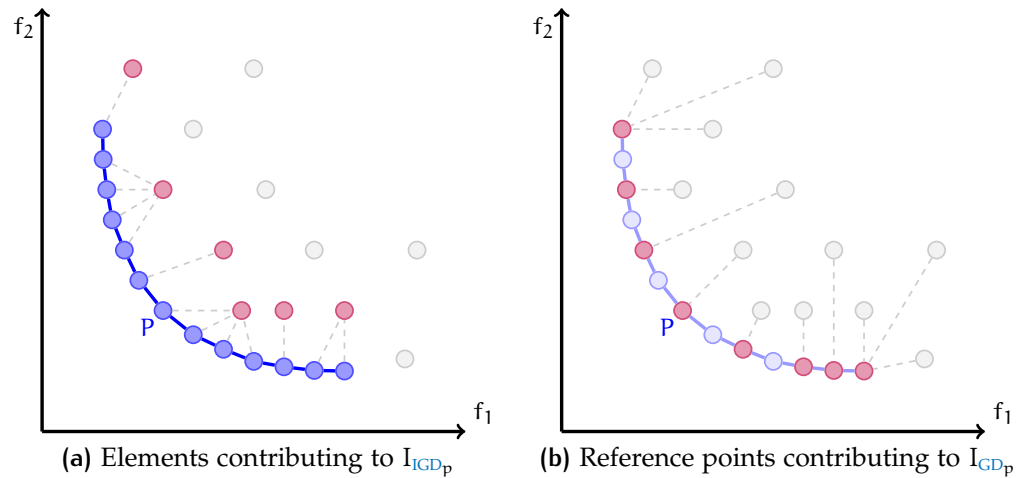


Figure 6.3.: GD and IGD contributing elements

- the *approximation set* A :
given by the population of the MOEA
- the *reference set* R :
a (discrete) representation of the Pareto front for the MOP

Since the Pareto front is generally unknown, an approximation must be constructed to be used as the reference set in the MOEA. Some properties are desirable for the reference set:

1. All the elements in A should be dominated by elements of R .
2. R can be (easily) updated if A is slightly modified.
3. Any element in R cannot (strongly) dominate other elements in the reference set.
4. The elements of R should be evenly spaced.

Our proposal for building the reference set is described in Algorithm 6.2.

Algorithm 6.2 Reference set construction**Require:** Set of vectors A **Require:** Resolution r for the reference SetApproximate the ideal and nadir objective vectors from A (Figure 6.4a)**if** the ideal or nadir objective vector changed **then**Build an upper k -dimensional frame (Figure 6.4b)

- bounded by the ideal and nadir objective vectors
- with the given resolution r

end ifFit frame to the non-dominated points in A (Figure 6.4c)

Remove duplicated points

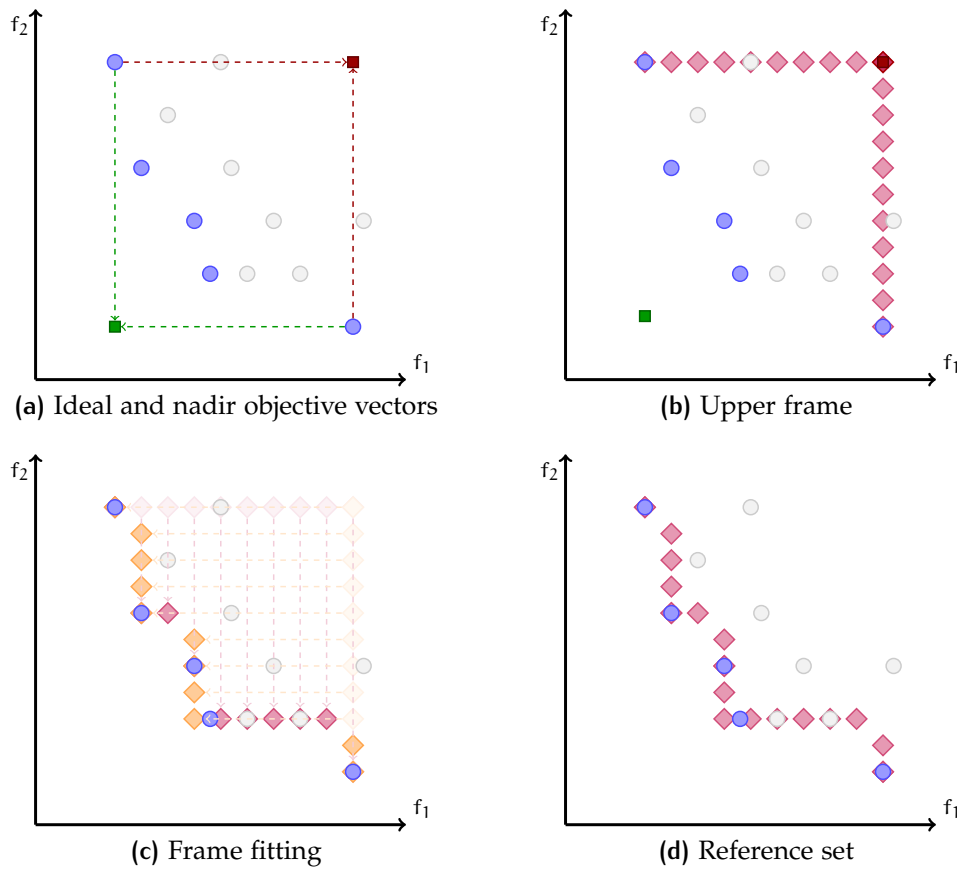
return the built reference set R (Figure 6.4d)

Figure 6.4.: Reference set construction for a bi-objective MOP

6.4 THE ALGORITHM

The DE with Δ_p -based selection is presented in Algorithm 6.3. Some modifications were done to *DE/rand/1/bin* to adapt the EA for multi-objective optimization using the proposed Δ_p -based selection mechanism:

- the creation and update of the reference set (lines 3 and 18) was included
- Δ_p contributions must be computed considering the objective function values of the population and the reference set (lines 4 and 19)
- The $(1+1)$ -selection was replaced by a (NP+NP)-selection (lines 17-20)
- The outcome of DE is its final population instead of a single individual (line 22)

Algorithm 6.3 *DE/rand/1/bin* with Δ_p -based selection

```

1: Generate initial population (P) of NP individuals randomly
2: Evaluate the objective function G for each individual
3: Initialize the reference set S
4: Compute the  $I_{\Delta_p}$  contributions of each individual according to S
5: repeat
6:    $\{O\} \leftarrow \emptyset$ 
7:   for each individual  $\vec{x} \in P$  do
8:     Pick three individuals  $\vec{a}, \vec{b}, \vec{c} \in P$ 
       distinct from  $\vec{x}$  and from each other.
9:     Pick an index  $R \in \{1, \dots, D\}$ 
10:    for  $i = 1, \dots, D$  do
11:      Pick  $r_i \in (0, 1)$  uniformly at random
12:      if ( $i = R$  or  $r_i < CR$ ) then  $y_i = a_i + F(b_i - c_i)$ 
13:      else  $y_i = x_i$ 
14:    end for
15:     $\{O\} \leftarrow \{O\} + \vec{y}$ 
16:  end for
17:  Evaluate the objective function G for all the generated offspring O
18:  Update S
19:  Compute the  $I_{\Delta_p}$  contributions of  $P \cup O$  according to the updated S
20:   $P \leftarrow$  the NP individuals which the lowest  $I_{\Delta_p}$  contributions
21: until a stopping criterion is met
22: return the final population

```

Our proposed approach assumes the real Pareto front of the problem is $(k - 1)$ -dimensional and continuous. Therefore, we don't expect our proposed approach to work well with functions where the Pareto front lost dimensionality or where it is discontinuous.

 PERFORMANCE ASSESSMENT

Contents

7.1	ZDT test problems	56
7.2	DTLZ test problems	59
7.3	Scalability of our approach	67
7.4	Closing remarks	69

In this chapter, we present the results obtained using the algorithm discussed in Chapter 6. Our proposed approach (Δ_p -DE) was validated using 5 bi-objective and 7 tri-objective problems from the *Zitzler-Deb-Thiele* (ZDT) [56] and the *Deb-Thiele-Laumanns-Zitzler* (DTLZ) [16] sets. The definitions of these test functions are provided in Appendix A.

Our approach is compared to NSGA-II and SMS-EMOA using the following quality indicators (see Section 4):

- I_{GD_p}
- I_{IGD_p}
- I_{Δ_p}
- I_{SS}
- I_{e^+}
- I_C

30 independent runs of each algorithm were performed with a population of 100 individuals running for 200 generations with the parameters presented in Table 7.1

Δ_p -DE	F	1.0	NSGA-II	p_c	0.9	SMS-EMOA	p_m	$1/ \vec{x} $
	Cr	0.4		p_m	$1/ \vec{x} $		η_c	15
	r	$\lceil k^{-1}\sqrt{100} \rceil$		η_c	15		η_m	20
	p	1.0		η_m	20		η_m	20

Table 7.1.: Parameters used for all the test problems

7.1 ZDT TEST PROBLEMS

The results obtained for most ZDT problems were very similar amongst the three algorithms. Therefore, only two special cases (ZDT₃ and ZDT₄) will be discussed in this section. The complete results for the ZDT test problems are provided in Appendix B.1

ZDT3

ZDT₃ has a discontinuous Pareto front consisting of five non-contiguous convex parts [56]. Since our approach always generates a continuous reference set, it is expected that the solutions cluster together nearby the discontinuities in the Pareto front.

The results for this function are, however, only slightly worse than those obtained by NSGA-II and SMS-EMOA regarding both the Δ_p and the hypervolume indicator (see Figure 7.1).

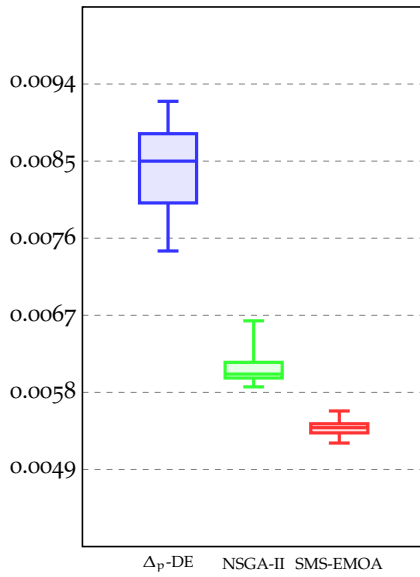
Figure 7.3 shows an example of the outcome sets generated (in objective function space) by the three MOEAs. The expected effect appears in the results from our approach: the solutions have occupied the areas close to the discontinuities while leaving larger gaps in the central parts of the sections of the Pareto front.

ZDT4

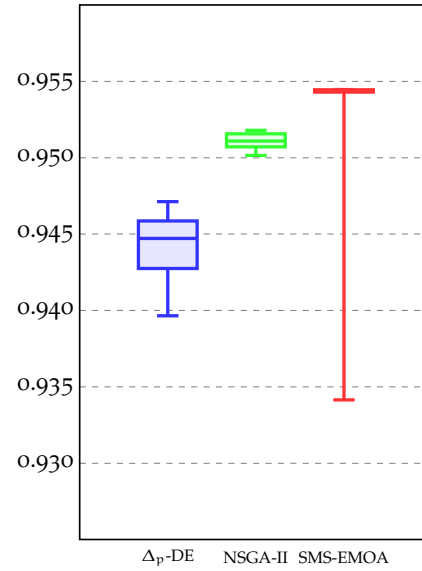
ZDT₄ is a highly multi-frontal problem: it contains 21^9 local Pareto-optimal fronts [56].

Our approach, in general, obtains better results than SMS-EMOA and marginally worse than NSGA-II, but exhibits a more stable behavior (see Figure 7.2).

Figure 7.4 shows an example of the outcome sets generated (in objective function space) by the three MOEAs. All three MOEAs reached the global Pareto front; however, SMS-EMOA didn't cover it completely.

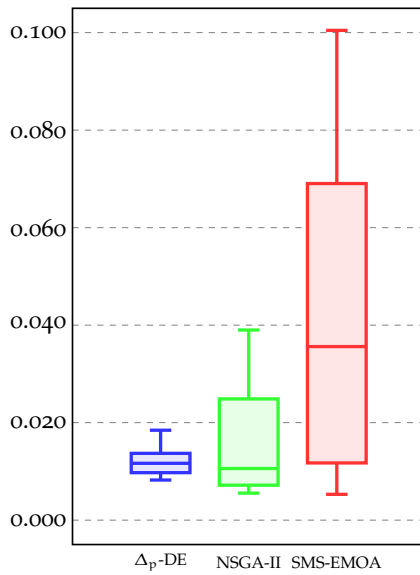


(a) Δ_1 indicator

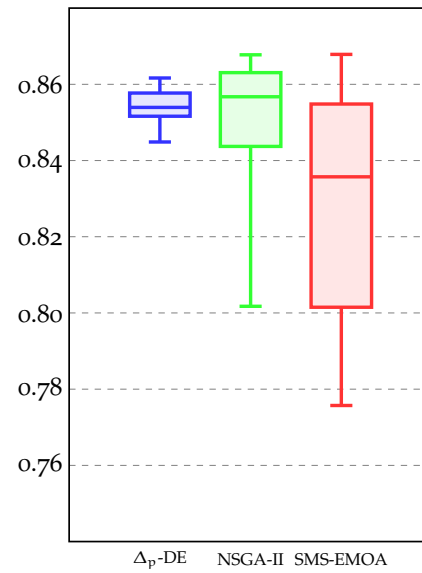


(b) Hypervolume indicator

Figure 7.1.: Indicators for ZDT3



(a) Δ_1 indicator



(b) Hypervolume indicator

Figure 7.2.: Indicators for ZDT4

PERFORMANCE ASSESSMENT

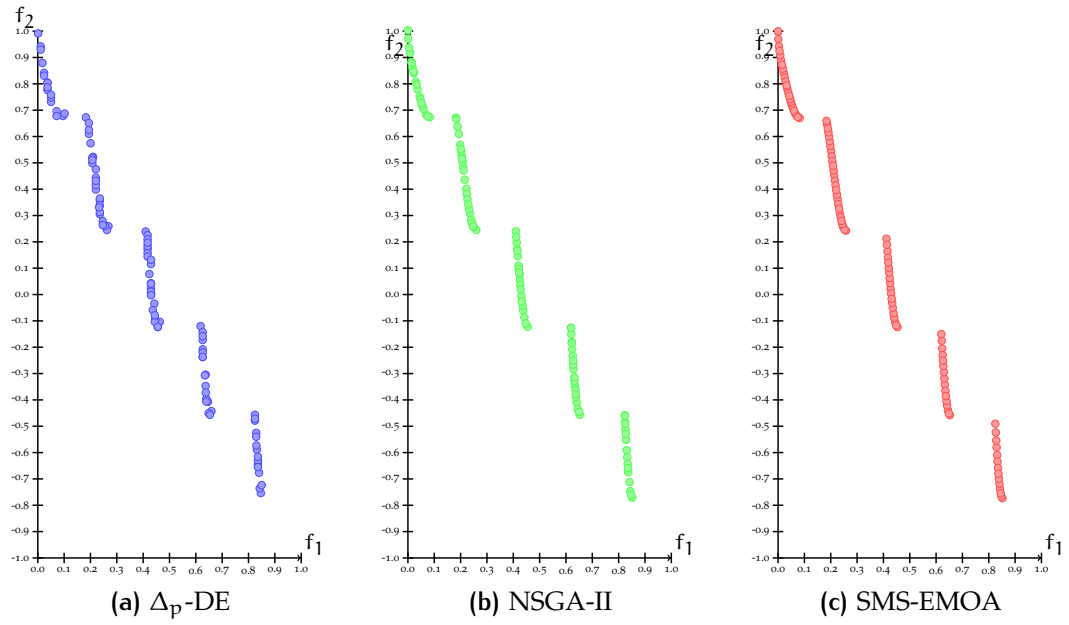


Figure 7.3.: Graphical results for ZDT3

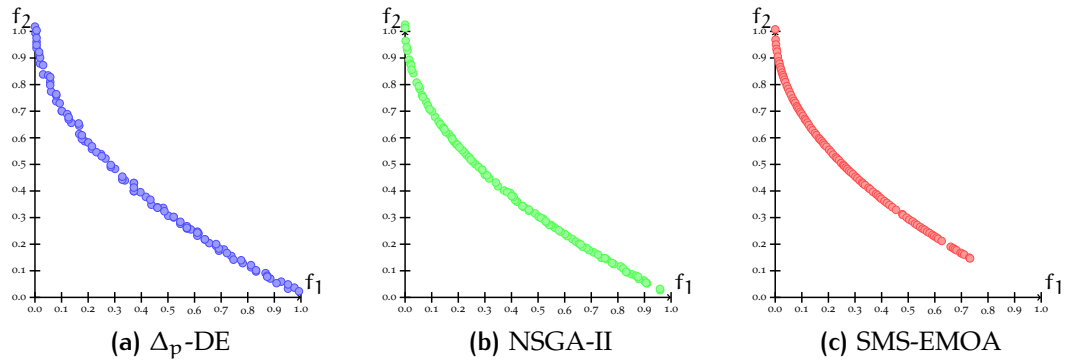


Figure 7.4.: Graphical results for ZDT4

7.2 DTLZ TEST PROBLEMS

The problems for which the three algorithms generated very similar results won't be discussed in this section. The complete results for the [DTLZ](#) test problems are provided in [Appendix B.2](#)

DTLZ1

[DTLZ1](#) is a multi-frontal problem with $(11^n - 1)$ local Pareto optimal fronts. The global Pareto front is a linear hyperplane: $\sum_{m=1}^k f_m = 0.5$ [16].

The results obtained for this problem using our approach are considerably better than those obtained with both [SMS-EMOA](#) and [NSGA-II](#) (see [Figure 7.5](#)).

[Figure 7.7](#) shows an example of the outcome sets generated (in objective function space) by the three MOEAs. The outcome set of [NSGA-II](#) doesn't appear in the figure since it is too far from the Pareto front while the outcome set of [SMS-EMOA](#) is shown only partially for a similar reason.

DTLZ3

[DTLZ3](#) is also a multi-frontal problem: it has $(3^n - 1)$ local Pareto fronts which are all parallel to the global Pareto front.

The results obtained using our approach are significantly better than those obtained by [SMS-EMOA](#) and [NSGA-II](#) (see [Figure 7.6](#)).

[Figure 7.8](#) shows an example of the outcome sets generated (in objective function space) by the three MOEAs. The outcome set of [NSGA-II](#) doesn't appear in the figure since it is too far from the Pareto front; the outcome set of [SMS-EMOA](#) is shown only partially for a similar reason. Our proposed approach was able to reach the true Pareto front of this problem.

PERFORMANCE ASSESSMENT

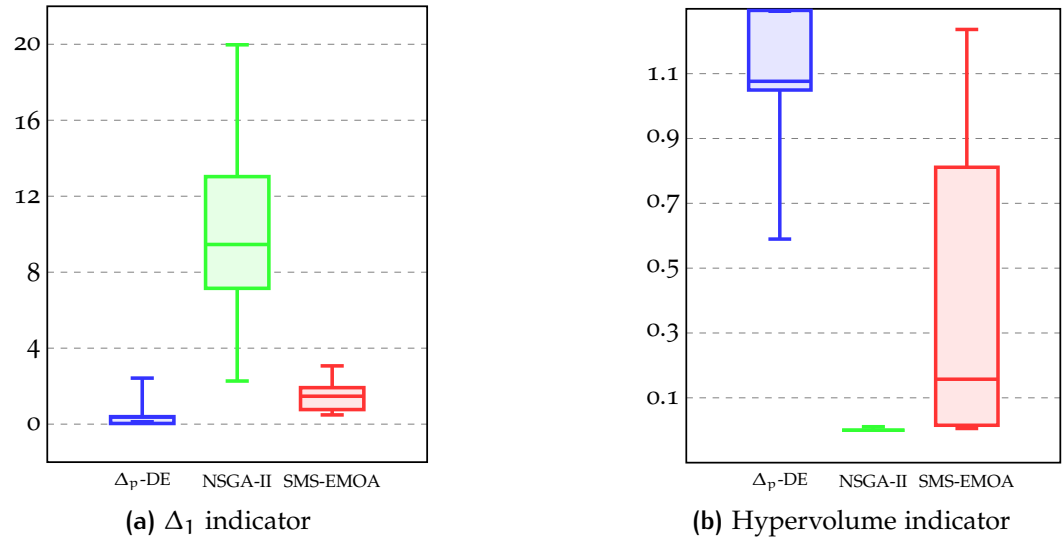


Figure 7.5.: Indicators for [DTLZ1](#)

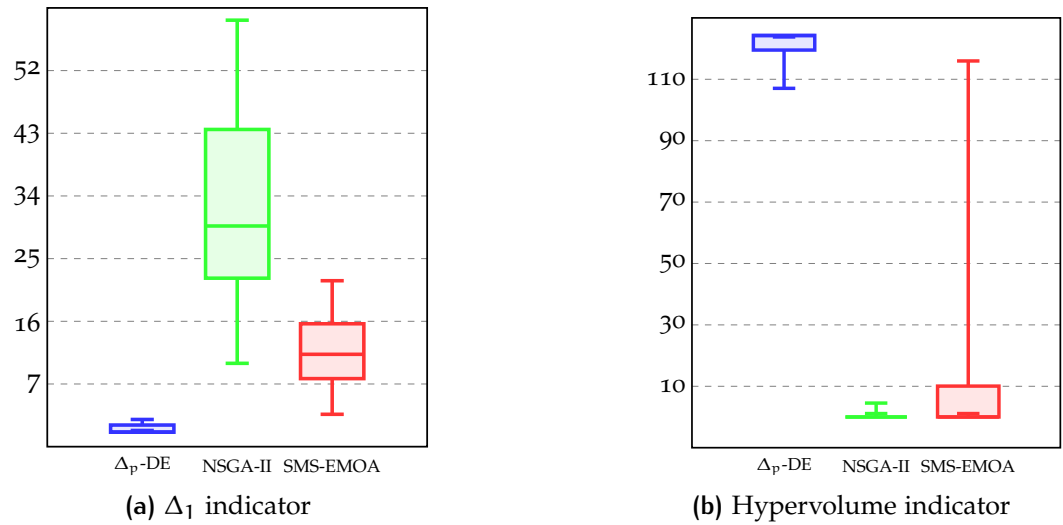


Figure 7.6.: Indicators for [DTLZ3](#)

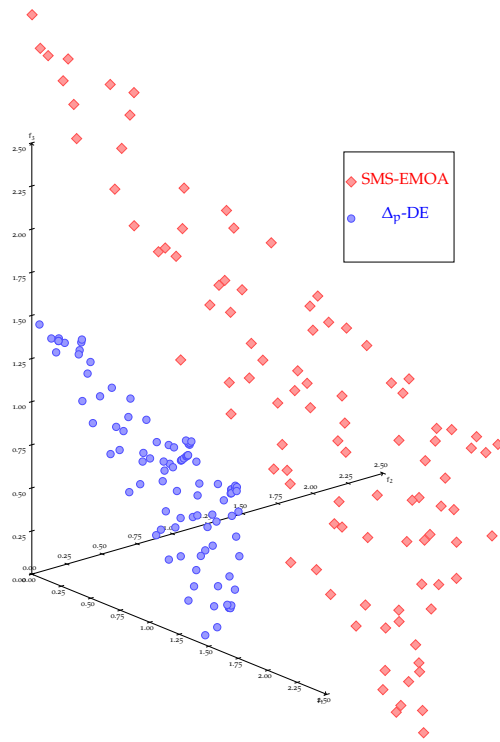


Figure 7.7.: Graphical results for $DTLZ_1$

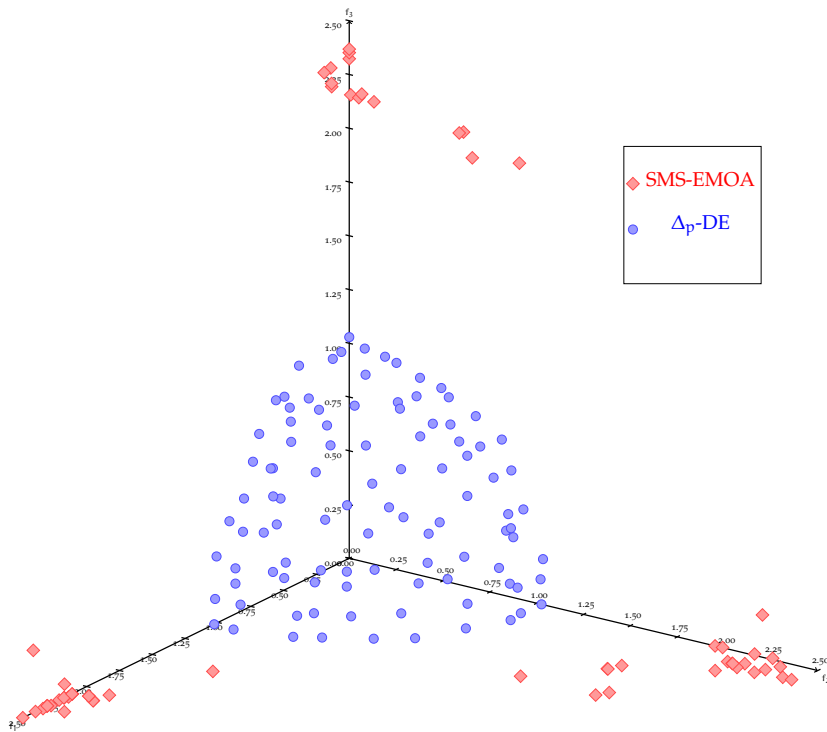


Figure 7.8.: Graphical results for $DTLZ_3$

DTLZ5

The true Pareto front of [DTLZ5](#) is a curve instead of a surface. [DTLZ5](#) is considered an easy problem since the search spaces contains a bias towards solutions close to the true Pareto front. However, our approach has difficulties for solving this problem since it uses a surface as a reference set (see [Figure 7.9](#)).

[Figure 7.11](#) shows an example of the outcome sets generated (in objective function space) by the three [MOEAs](#). As expected, the outcome set of our proposed approach is a surface instead of the line obtained by [NSGA-II](#) and [SMS-EMOA](#).

DTLZ6

[DTLZ6](#) has the same Pareto front as [DTLZ5](#). However, [DTLZ6](#) is considered a harder problem. We expected our approach to obtain similar results to those obtained for [DTLZ5](#); however, the results obtained using our approach were significantly better than those obtained by the other [MOEAs](#) (see [Figure 7.10](#)).

[Figure 7.12](#) shows an example of the outcome sets generated (in objective function space) by the three [MOEAs](#). Our algorithm reached the true Pareto front while both [NSGA-II](#) and [SMS-EMOA](#) found dominated surfaces “close” to the true Pareto front.

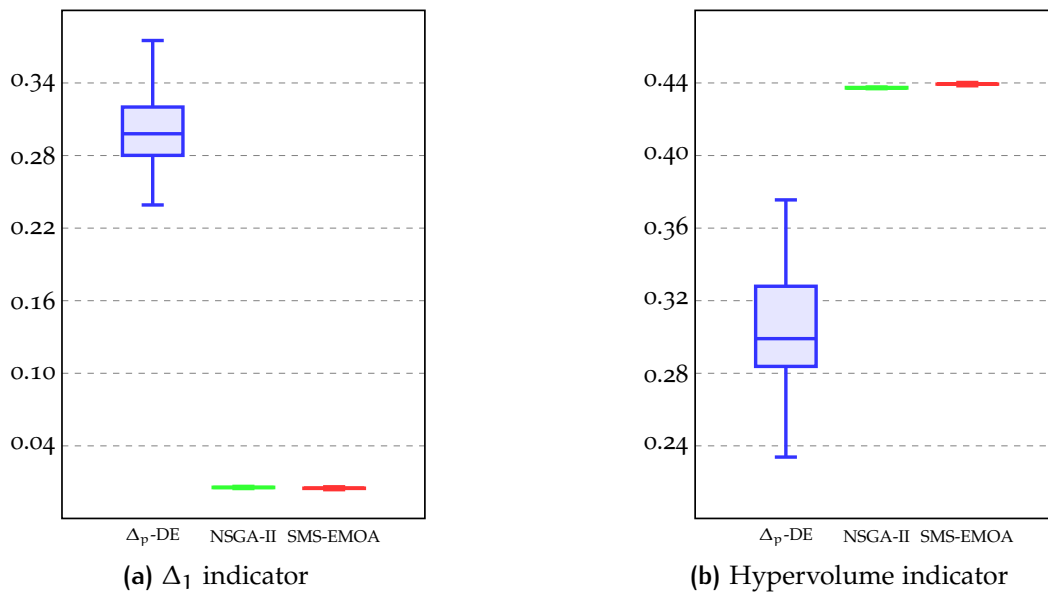


Figure 7.9.: Indicators for DTLZ5

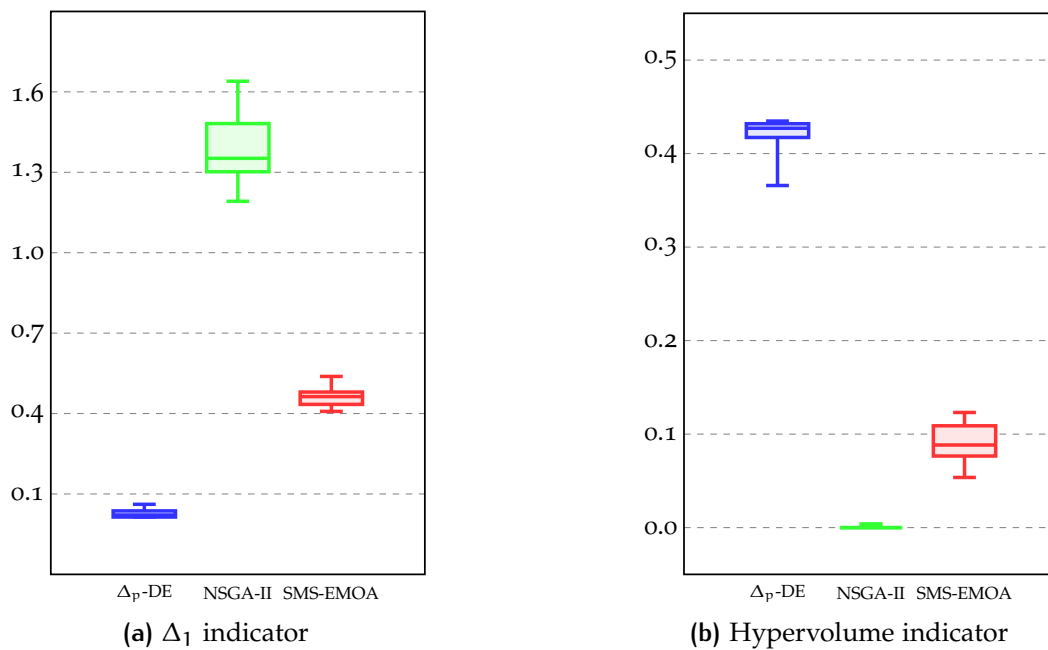


Figure 7.10.: Indicators for DTLZ6

PERFORMANCE ASSESSMENT

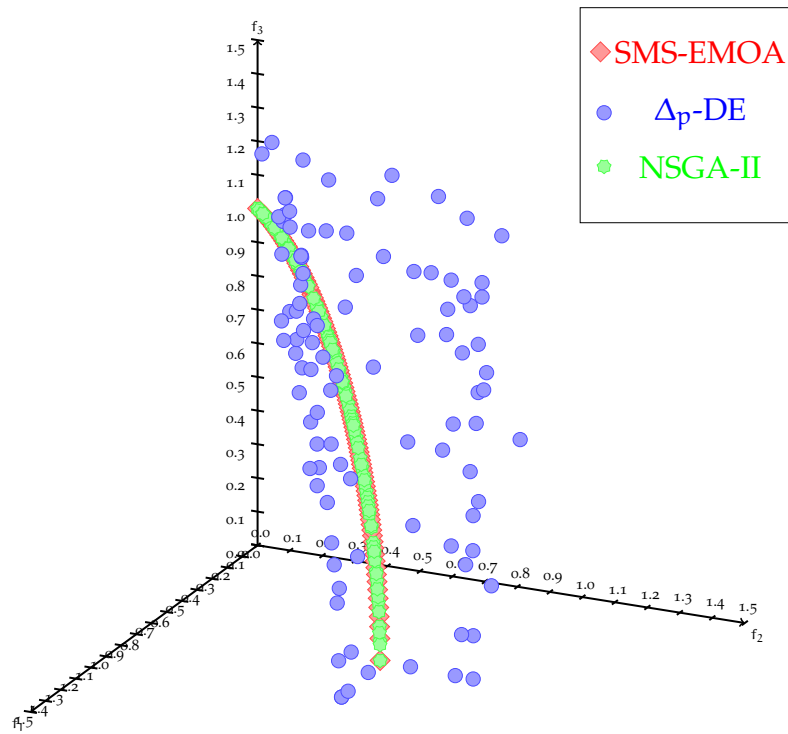


Figure 7.11.: Graphical results for *DTLZ5*

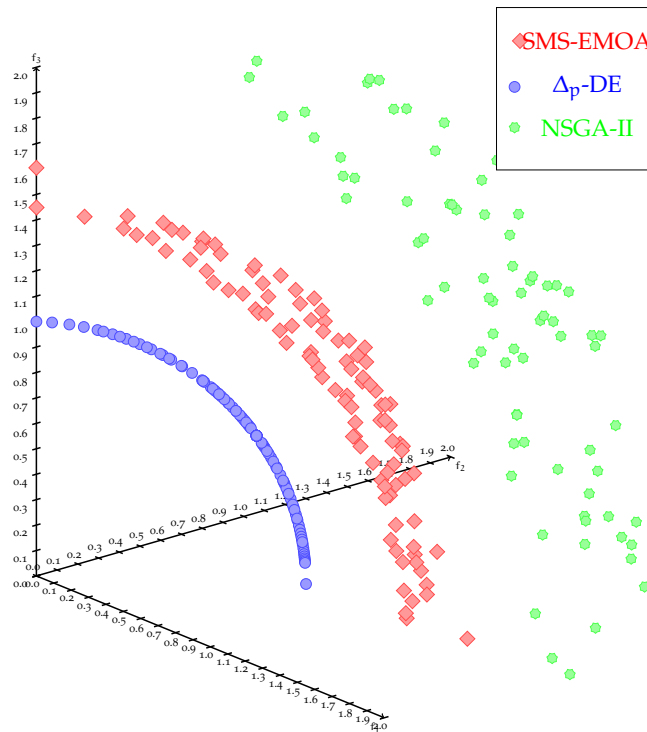


Figure 7.12.: Graphical results for *DTLZ6*

DTLZ7

This problem has several discontinuous Pareto optimal regions. The results obtained using our approach are similar to those obtained by [SMS-EMOA](#) and [NSGA-II](#) (see Figure 7.13). As with [ZDT3](#), we expected the solutions to cluster together close to the discontinuities in the Pareto front.

The results for this function are similar to those obtained by [NSGA-II](#) and [SMS-EMOA](#) regarding both the Δ_p and the hypervolume indicators (see Figure 7.13).

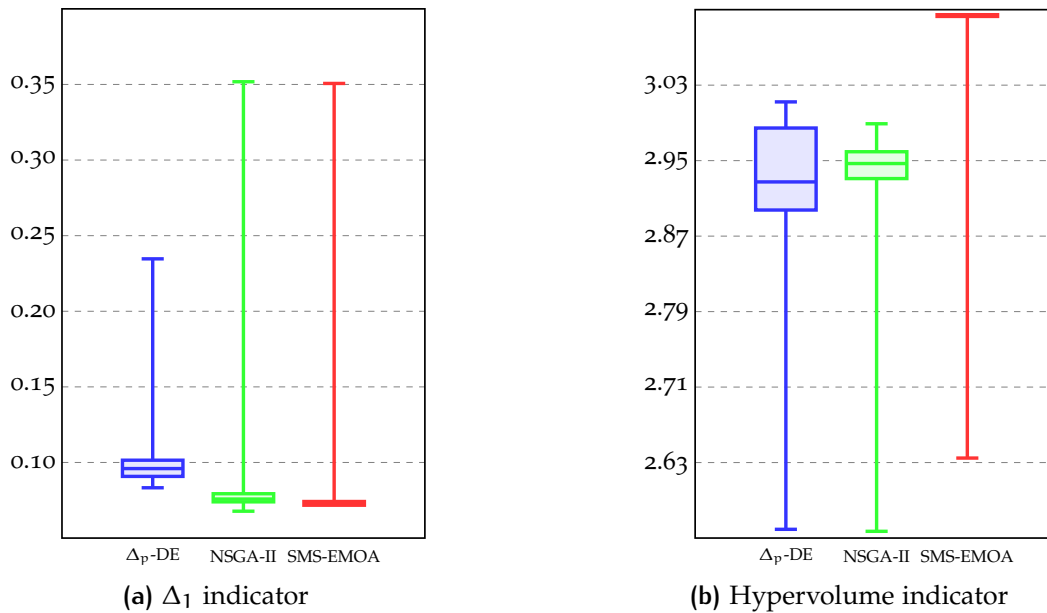


Figure 7.13.: Indicators for [DTLZ7](#)

Figure 7.14 shows an example of the outcome sets generated (in objective function space) by the three [MOEAs](#). The expected effect is clear in the outcome: solutions are clustered towards the bottom border of the Pareto optimal sections with few solutions in the middle regions.

PERFORMANCE ASSESSMENT

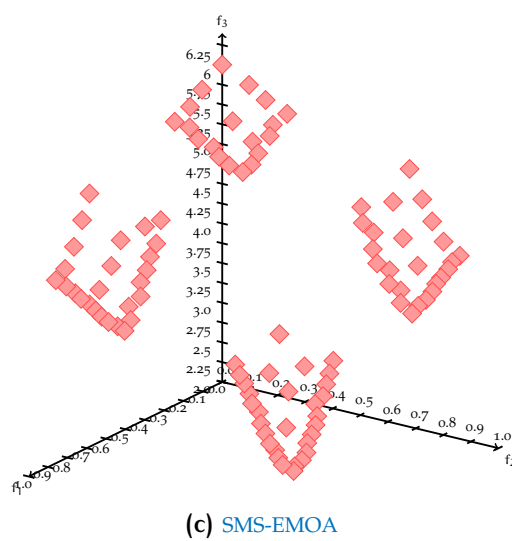
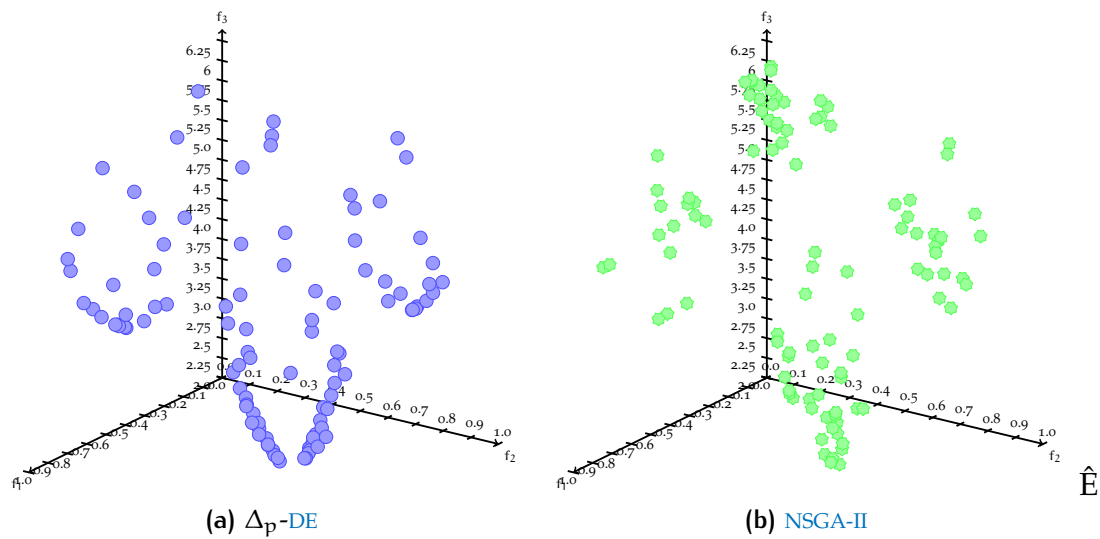


Figure 7.14.: Graphical results for DTLZ7

7.3 SCALABILITY OF OUR APPROACH

To evaluate the scalability of our approach, we scaled the [DTLZ2](#) function from 2 to 10 objectives. We compared our approach to both [NSGA-II](#) and [SMS-EMOA](#). 10 independent runs of each algorithm were performed with a population of 200 individuals running for 200 generations using the parameters shown in [Tables 7.2 and 7.3](#)

Δ_p -DE	F	0.5	NSGA-II	p_c	0.9	SMS-EMOA	p_m	$1/ \bar{x} $
	Cr	0.3		p_m	$1/ \bar{x} $		η_c	15
	p	1.0		η_c	15		η_m	20
				η_m	20			

Table 7.2.: Parameters used for our scalability testing

The outcome sets of the algorithms were compared using the exact generational distance indicator (which, due to the geometry of the Pareto front for [DTLZ2](#), can be computed as $\sum_{i=1}^k f_k^2 - 1$), the hypervolume indicator (using $\vec{1.1}$ as reference point), and the running time of each [MOEA](#).

Objectives	2	3	4	5	6	7	8	9	10
r	150	15	6	4				3	

Table 7.3.: Resolution values for Δ_p -DE

In most cases, the outcome produced by the [SMS-EMOA](#) was better according to the quality indicators used (see [Figure 7.15](#)). However, our approach reached only slightly worse results (obtaining, on average, over 95% of the best hypervolume value found by [SMS-EMOA](#)) while requiring a very reasonable computational time (under 0.003% of the time used by [SMS-EMOA](#))

PERFORMANCE ASSESSMENT

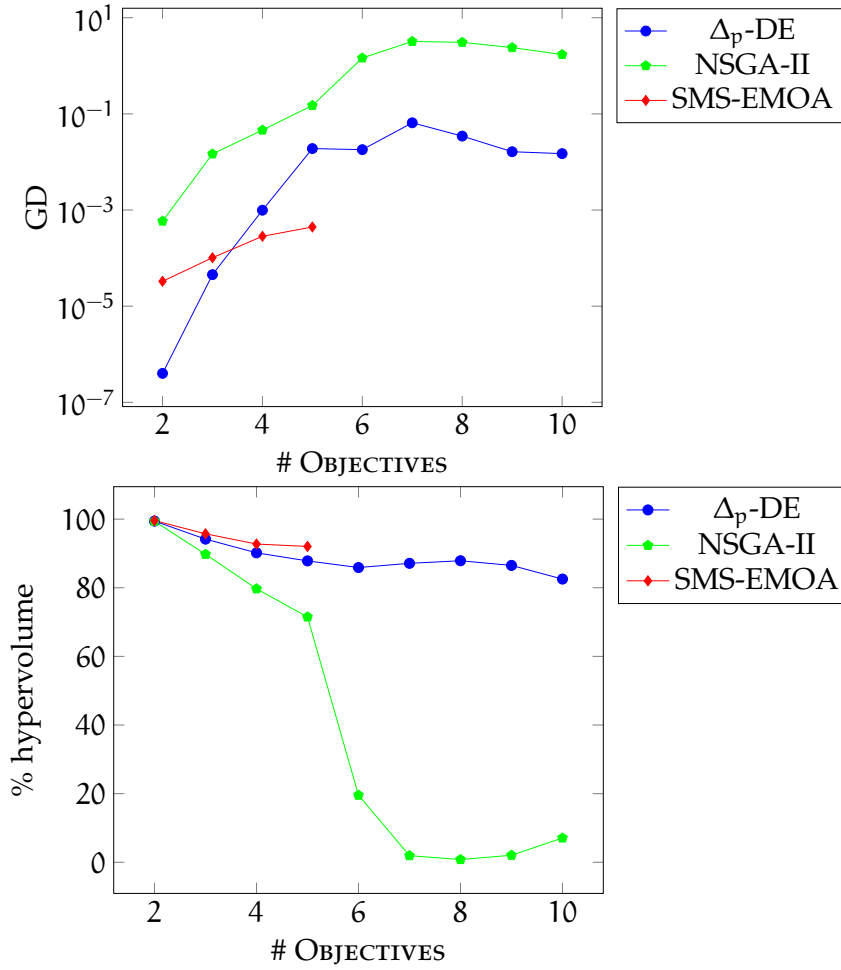


Figure 7.15.: Indicator values for the DTLZ2 scaled problem

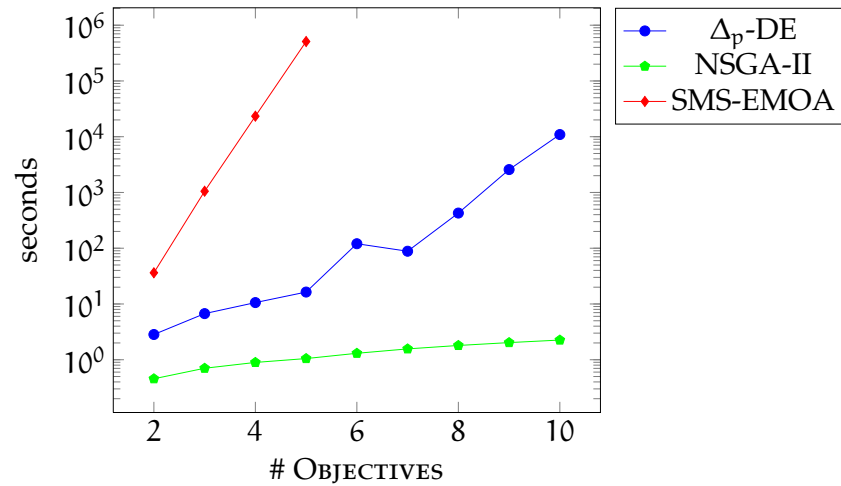


Figure 7.16.: Running time for the DTLZ2 scaled problem

7.4 CLOSING REMARKS

These results seem to indicate that the Δ_p -DE is a suitable MOEA for solving MOPs. When dealing with MOPs with few (2 or 3) objectives, the results obtained were competitive with respect to both NSGA-II and SMS-EMOA. For MOPs with many (4 or more) objectives, NSGA-II showed a deterioration in the quality of its outcome as the number of objective functions increased while maintaining a very low computational cost. For such problems, SMS-EMOA was shown to obtain good results for many-objective problems; however, its computational cost was prohibitively high. Our proposed approach was shown to obtain competitive results with respect to SMS-EMOA while keeping a reasonable computational cost.

PERFORMANCE ASSESSMENT

CONCLUSIONS AND FUTURE WORK

Contents

8.1	Conclusions	71
8.2	Future work	73

The main goal of this thesis was to design an indicator-based *multi-objective evolutionary algorithm* (MOEA) that was competitive with respect to state-of-the-art MOEAs.

We have proposed an approach based on the minimization of the Δ_p quality indicator. Our proposed approach was shown to obtain competitive results when compared to state-of-the-art MOEAs in several test problems from the specialized literature. We also showed that our proposed approach was able to find promising results for test problems with many (at least 4) objective functions.

8.1 CONCLUSIONS

The following conclusions were obtained during the development of the approach and its experimental validation:

- The use of quality indicators with metric properties (such as I_{Δ_p}) can be successfully used by MOEAs to guide the optimization process even when these indicators are not Pareto-compliant.

- We proposed a new MOEA based on a DE strategy with a Δ_p -based selection mechanism (see Chapter 6). The proposed approach provided competitive results with respect to NSGA-II and SMS-EMOA in several test problems taken from the specialized literature (see Chapter 7).

Our proposed approach, in most cases, outperformed NSGA-II but was slightly outperformed by SMS-EMOA. The difference in the quality of the outcomes of our proposed approach and those produced by SMS-EMOA is very small. However, the computational cost of SMS-EMOA is much higher than that of our proposed approach because of the hypervolume computations involved.

- Our proposed approach seems to be less prone to getting trapped in local optima than NSGA-II and SMS-EMOA. The results obtained by our proposed approach in highly multi-frontal problems (such as ZDT4, DTLZ1, and DTLZ3) are significantly better than those obtained by the other approaches with respect to which it was compared.
- We used the test problem DTLZ2 to test the scalability of our proposed approach, as well as that of NSGA-II and SMS-EMOA. NSGA-II showed as expected a deterioration in the quality of its outcome when dealing with a high number of objectives while our proposed approach and the SMS-EMOA produced good results for problems with many objectives.

SMS-EMOA consistently outperformed our approach (by a small amount) albeit at a very high computational cost (this MOEA could only be tested up to 5 objectives while both NSGA-II and our approach were tested up to 10 objectives). In contrast, the computational cost of our approach is significantly lower than that of SMS-EMOA (see Section 7.3).

8.2 FUTURE WORK

Currently, our proposed approach assumes that the Pareto front of every problem is continuous and $(k - 1)$ -dimensional. Then, the results obtained for problems that do not fulfill these requirements (such as ZDT3, DTLZ5, DTLZ6, and DTLZ7) are often worse than those obtained by other MOEAs. Therefore, further research is required to generalize the construction of the reference sets to problems with different characteristics.

Also, the running time of the construction of the reference sets used by our proposed approach is currently exponential on the number of objectives. Therefore, the running time of the approach can be improved by using alternative procedures to build reference sets (such as linear interpolation techniques for bi-objective MOPs).

Finally, a more detailed analysis on the scalability of the approach is required since the results obtained in this thesis are promising but involve the use of only one scalable test problem.

CONCLUSIONS AND FUTURE WORK

TEST PROBLEMS

A.1 ZITZLER-DEB-THIELE TEST SET**ZDT1**

- Objective functions
- Bounds

$$g = 1 + 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

$$n = 30$$
$$0 \leq x_i \leq 1$$

$$f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}, g) = 1 - \sqrt{\frac{f_1}{g(\vec{x})}}$$

- Pareto front: convex, connected

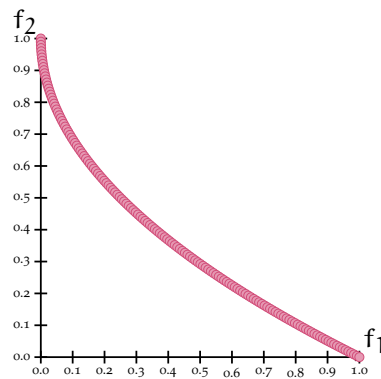


Figure A.1.: Pareto front for [ZDT1](#)

TEST PROBLEMS

ZDT2

- Objective functions

$$g = 1 + 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

$$f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}, g) = 1 - \left(\frac{f_1}{g(\vec{x})} \right)^2$$

- Bounds

$$n = 30$$

$$0 \leq x_i \leq 1$$

- Pareto front: non-convex, connected

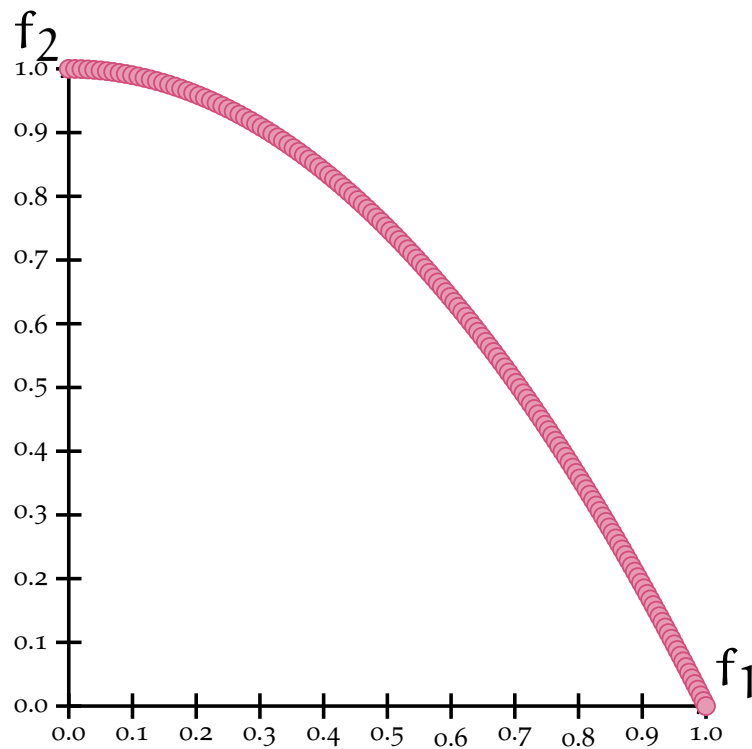


Figure A.2.: Pareto front for ZDT2

ZDT3

- Objective functions

$$g = 1 + 1 + \frac{9}{n-1} \sum_{i=2}^n x_i$$

$$f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}, g) = 1 - \sqrt{\frac{f_1}{g(\vec{x})}} - \frac{f_1}{g(\vec{x})} \sin(10\pi f_1)$$

- Bounds

$$n = 30$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 30\}$$

- Pareto front: discontinuous

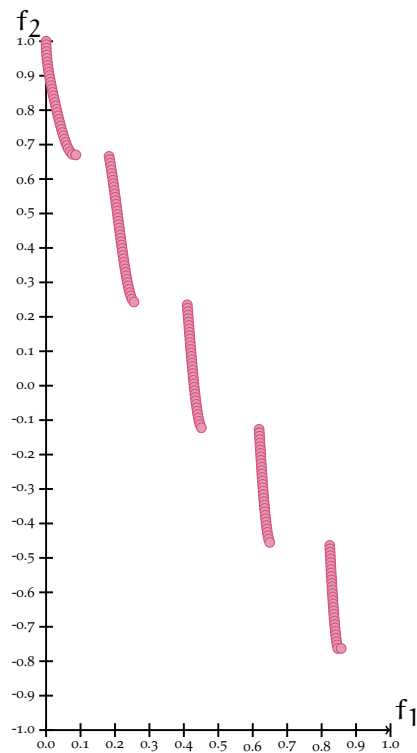


Figure A.3.: Pareto front for ZDT3

TEST PROBLEMS

ZDT4

- Objective functions

$$g = 1 + 10(n - 1) + \sum_{i=1}^n (x_i^2 - 10 \cos(4\pi x_i))$$

$$f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}, g) = 1 - \sqrt{\frac{f_1}{g(\vec{x})}} - \frac{f_1}{g(\vec{x})} \sin(10\pi f_1)$$

- Bounds

$$n = 10$$

$$0 \leq x_1 \leq 1$$

$$-5 \leq x_i \leq 5 \quad i \in \{2, \dots, 10\}$$

- Pareto front: non-convex, multi-frontal

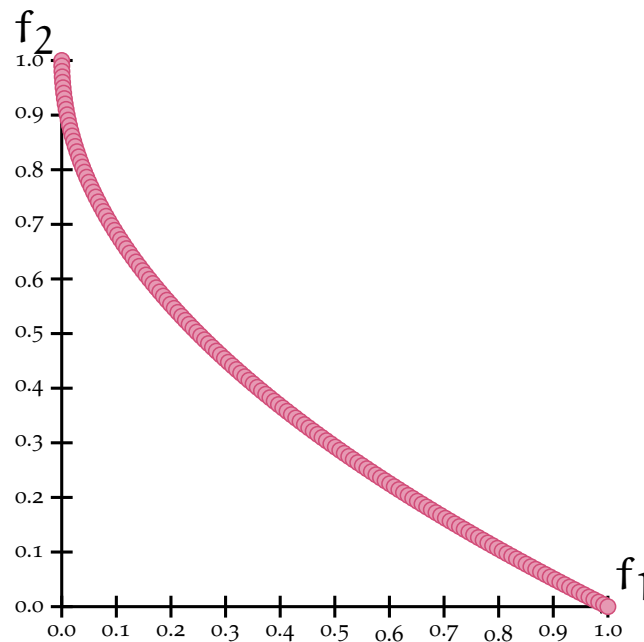


Figure A.4.: Pareto front for ZDT4

ZDT6

- Objective functions

$$g = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{9} \right)^{0.25}$$

$$f_1(\vec{x}) = 1 - e^{-4x_1} \sin^6(6\pi x_1)$$

$$f_2(\vec{x}, g) = 1 - \left(\frac{f_1}{g(\vec{x})} \right)^2$$

- Bounds

$$n = 10$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 10\}$$

- Pareto front: non-convex

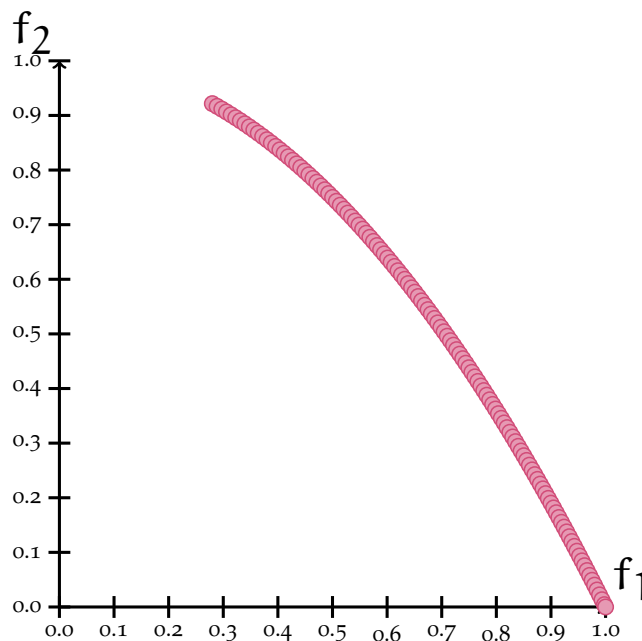


Figure A.5.: Pareto front for ZDT6

A.2 DEB-THIELE-LAUMANN-S-ZITZLER TEST SET

DTLZ1

- Objective functions

$$g = 100 \left(5 + \sum_{i=3}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right)$$

$$f_1(\vec{x}) = \frac{1}{2} x_1 x_2 (1 + g(\vec{x}))$$

$$f_2(\vec{x}) = \frac{1}{2} x_1 (1 - x_2) (1 + g(\vec{x}))$$

$$f_3(\vec{x}) = \frac{1}{2} (1 - x_1) (1 + g(\vec{x}))$$

- Bounds

$$n = 12$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 12\}$$

- Pareto front: linear, multi-frontal

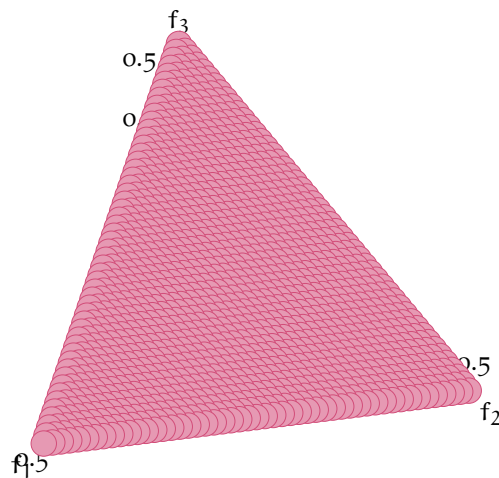


Figure A.6.: Pareto front for [DTLZ1](#)

DTLZ2

- Objective functions

$$g = \sum_{i=3}^n (x_i - 0.5)^2$$

$$f_1(\vec{x}) = \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) (1 + g(\vec{x}))$$

$$f_2(\vec{x}) = \cos\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) (1 + g(\vec{x}))$$

$$f_3(\vec{x}) = \sin\left(\frac{\pi}{2}x_1\right) (1 + g(\vec{x}))$$

- Bounds

$$n = 12$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 12\}$$

- Pareto front: non-convex, connected

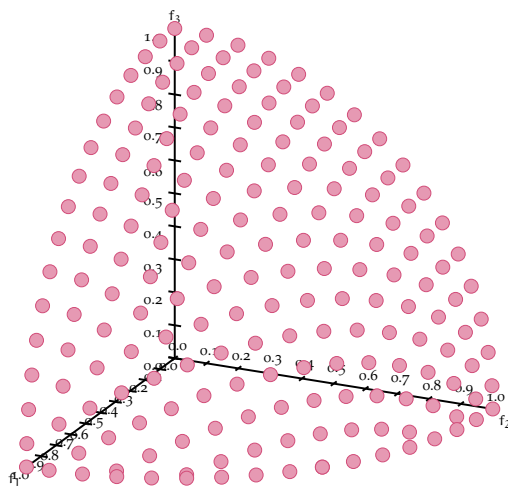


Figure A.7.: Pareto front for [DTLZ2](#)

TEST PROBLEMS

DTLZ3

- Objective functions

$$g = 100 \left(5 + \sum_{i=3}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 5)) \right)$$

$$f_1(\vec{x}) = \cos\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) (1 + g(\vec{x}))$$

$$f_2(\vec{x}) = \cos\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) (1 + g(\vec{x}))$$

$$f_3(\vec{x}) = \sin\left(\frac{\pi}{2}x_1\right) (1 + g(\vec{x}))$$

- Bounds

$$n = 12$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 12\}$$

- Pareto front: non-convex, connected, multi-frontal

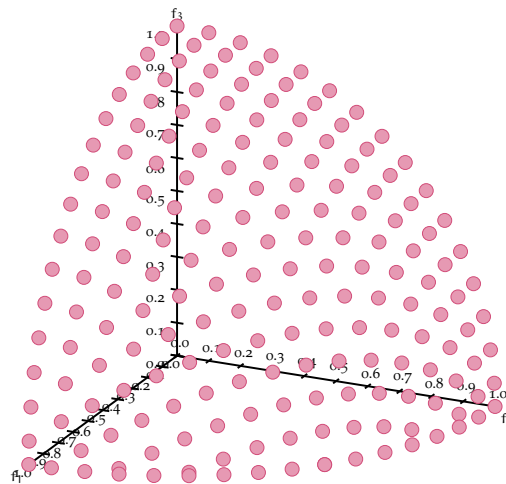


Figure A.8.: Pareto front for [DTLZ3](#)

DTLZ4

- Objective functions

$$g = 100 \left(5 + \sum_{i=3}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 5)) \right)$$

$$f_1(\vec{x}) = \cos\left(\frac{\pi}{2}x_1^\alpha\right) \cos\left(\frac{\pi}{2}x_2^\alpha\right) (1 + g(\vec{x}))$$

$$f_2(\vec{x}) = \cos\left(\frac{\pi}{2}x_1^\alpha\right) \sin\left(\frac{\pi}{2}x_2^\alpha\right) (1 + g(\vec{x}))$$

$$f_3(\vec{x}) = \sin\left(\frac{\pi}{2}x_1^\alpha\right) (1 + g(\vec{x}))$$

- Bounds

$$n = 12$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 12\}$$

$$\alpha = 100$$

- Pareto front: non-convex, connected

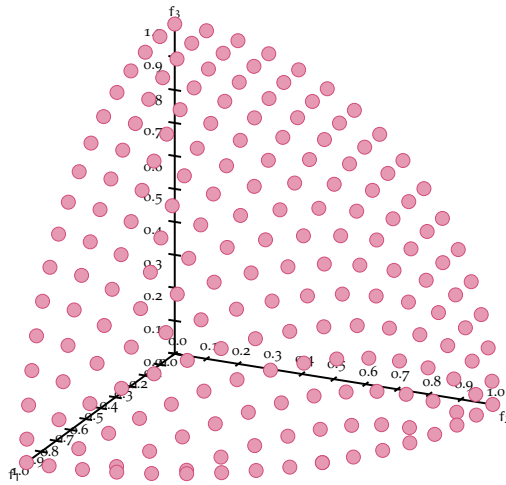


Figure A.9.: Pareto front for [DTLZ4](#)

TEST PROBLEMS

DTLZ5

- Objective functions

$$g = \sum_{i=3}^n (x_i - 0.5)^2$$

$$\theta_1 = \frac{\pi}{2} x_1$$

$$\theta_2 = \frac{\pi}{4(1 + g(\vec{x}))} (1 + 2x_2 g(\vec{x}))$$

$$f_1(\vec{x}) = \cos(\theta_1) \cos(\theta_2) (1 + g(\vec{x}))$$

$$f_2(\vec{x}) = \cos(\theta_1) \sin(\theta_2) (1 + g(\vec{x}))$$

$$f_3(\vec{x}) = \sin(\theta_1) (1 + g(\vec{x}))$$

- Bounds

$$n = 12$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 12\}$$

- Pareto front: curve, connected

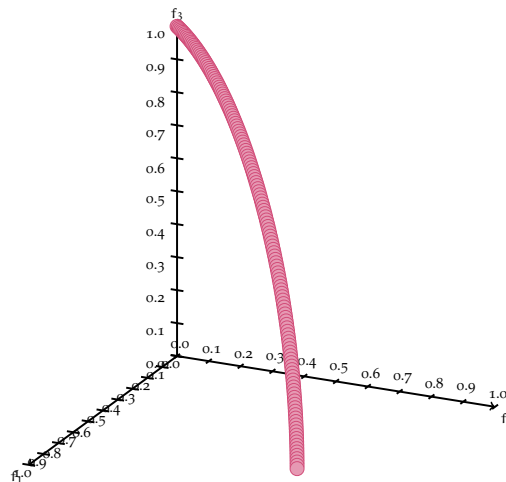


Figure A.10.: Pareto front for DTLZ5

DTLZ6

- Objective functions

$$g = \sum_{i=3}^n (x_i)^{0.1}$$

$$\theta_1 = \frac{\pi}{2} x_1$$

$$\theta_2 = \frac{\pi}{4(1+g(\vec{x}))} (1 + 2x_2 g(\vec{x}))$$

$$f_1(\vec{x}) = \cos(\theta_1) \cos(\theta_2) (1 + g(\vec{x}))$$

$$f_2(\vec{x}) = \cos(\theta_1) \sin(\theta_2) (1 + g(\vec{x}))$$

$$f_3(\vec{x}) = \sin(\theta_1) (1 + g(\vec{x}))$$

- Bounds

$$n = 12$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 12\}$$

- Pareto front: curve, connected

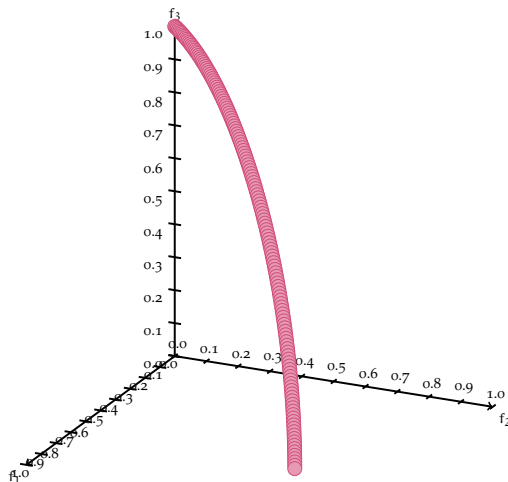


Figure A.11.: Pareto front for [DTLZ6](#)

TEST PROBLEMS

DTLZ7

- Objective functions

$$g = 1 + \frac{9}{22} \sum_{i=3}^n x_i$$

$$h = 3 - \sum_{i=1}^2 \left(\frac{f_i}{1+g} (1 + \sin(3\pi f_i)) \right)$$

$$f_1(\vec{x}) = x_1$$

$$f_2(\vec{x}) = x_2$$

$$f_3(\vec{x}) = (1 + g(\vec{x}))h(f_1, f_2, g(\vec{x}))$$

- Bounds

$$n = 22$$

$$0 \leq x_i \leq 1 \quad i \in \{1, \dots, 22\}$$

- Pareto front: disconnected

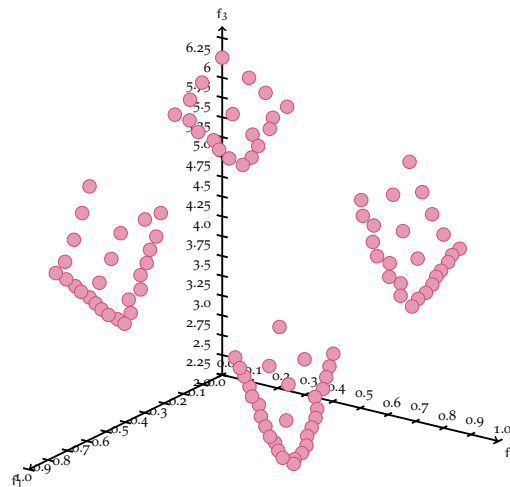


Figure A.12.: Pareto front for [DTLZ7](#)

B

COMPLETE RESULTS

Contents

B.1	ZDT test problems	88
B.2	DTLZ test problems	88
B.3	Scalability results	95

This appendix contains the complete results obtained for Chapter 7 with our proposed approach, [NSGA-II](#) and [SMS-EMOA](#).

Regarding unary quality indicators, the results reported are the mean values for each indicator (μ), the standard deviation (σ), and the best and worst values obtained from the runs performed. For binary indicators, only the mean values for each indicator are reported. The algorithm with the best mean value for each case is emphasized.

For the I_{GD} and I_{IGD} indicators, a uniformly distributed discretization of the real Pareto front was used as the reference set.

B.1 ZDT TEST PROBLEMS

Tables B.3, B.4, and B.5 summarize our results for the ZDT problems (except ZDT5, which was excluded for being a binary problem) using several quality indicators (see Chapter 4).

The reference points used for the hypervolume indicator (see Section 4.1.2.1) are provided in Table B.1.

Problem	Reference Point
ZDT1	(1.1 , 1.1)
ZDT2	
ZDT3	(0.9 , 1.1)
ZDT4	(1.1 , 1.1)
ZDT6	

Table B.1.: Reference points used for I_H in the ZDT test problems

B.2 DTLZ TEST PROBLEMS

Tables B.6, B.7, and B.8 summarize our results for the DTLZ test problems using several quality indicators (see Chapter 4).

The reference points used for the hypervolume indicator (see Section 4.1.2.1) are provided in Table B.2.

Problem	Reference Point
DTLZ1	(1.1 , 1.1 , 1.1)
DTLZ2	
DTLZ3	(5.0 , 5.0 , 5.0)
DTLZ4	(1.1 , 1.1 , 1.1)
DTLZ5	
DTLZ6	
DTLZ7	(1.1 , 1.1 , 7.0)

Table B.2.: Reference points used for I_H in the DTLZ test problems

Problem	Algorithm	GD indicator			IGD indicator				
		μ	σ	min	max	μ	σ	min	max
ZDT1	Δ_p -DE	0.002496	0.000224	0.001818	0.003176	0.011476	0.020331	0.004271	0.111430
	NSGA-II	0.003484	0.000308	0.002858	0.004430	0.005314	0.000219	0.004876	0.005859
	SMS-EMOA	0.002485	0.000103	0.002361	0.002847	0.003912	0.000071	0.003794	0.004178
ZDT2	Δ_p -DE	0.002515	0.000122	0.002146	0.002784	0.011305	0.012246	0.004384	0.055820
	NSGA-II	0.003730	0.000286	0.003218	0.004354	0.0054881	0.000265	0.004874	0.005954
	SMS-EMOA	0.002495	0.000064	0.002369	0.002613	0.004093	0.000077	0.003924	0.004263
ZDT3	Δ_p -DE	0.005114	0.000551	0.004012	0.006202	0.008455	0.000464	0.007438	0.009220
	NSGA-II	0.002991	0.000125	0.002643	0.003248	0.006065	0.000158	0.005843	0.006653
	SMS-EMOA	0.003091	0.002390	0.002368	0.015942	0.007855	0.009367	0.005191	0.047567
ZDT4	Δ_p -DE	0.015388	0.021458	0.006410	0.129818	0.015517	0.020882	0.007892	0.127303
	NSGA-II	0.013385	0.023693	0.003130	0.136242	0.0196718	0.022900	0.005204	0.117220
	SMS-EMOA	0.023356	0.062188	0.002964	0.308379	0.044153	0.038494	0.004953	0.155810
ZDT6	Δ_p -DE	0.002393	0.000082	0.002196	0.002523	0.003563	0.000171	0.003223	0.003973
	NSGA-II	0.016327	0.002081	0.012344	0.021931	0.016444	0.002000	0.012678	0.021888
	SMS-EMOA	0.007893	0.000912	0.006400	0.010305	0.008221	0.000899	0.006801	0.010660

Table B.3.: Results for the ZDT functions: I_{GD1} and I_{IGD1}

Problem	Algorithm	Hypervolume indicator					Spacing indicator				
		μ	σ	min	max		μ	σ	min	max	
ZDT1	Δ_p -DE	0.865146	0.019405	0.766833	0.871306	0.008415	0.001338	0.006567	0.011963		
	NSGA-II	0.868009	0.000549	0.866712	0.869206	0.006978	0.000653	0.005553	0.008308		
	SMS-EMOA	0.871766	0.000056	0.871660	0.871881	0.003099	0.000530	0.002303	0.005319		
ZDT2	Δ_p -DE	0.528088	0.013564	0.486930	0.537924	0.007435	0.000847	0.005811	0.009379		
	NSGA-II	0.533828	0.000646	0.532678	0.535358	0.007356	0.000545	0.006122	0.008877		
	SMS-EMOA	0.538295	0.000109	0.538073	0.538444	0.004452	0.000550	0.003206	0.005596		
ZDT3	Δ_p -DE	0.944345	0.001815	0.939544	0.947240	0.010145	0.001206	0.007241	0.012112		
	NSGA-II	0.951098	0.000489	0.950037	0.951900	0.007879	0.000709	0.006737	0.009128		
	SMS-EMOA	0.951629	0.011584	0.892422	0.954580	0.005276	0.000434	0.004271	0.006392		
ZDT4	Δ_p -DE	0.848062	0.033164	0.670988	0.862099	0.008295	0.000574	0.007263	0.009934		
	NSGA-II	0.845837	0.033015	0.689414	0.868200	0.007698	0.003500	0.004357	0.024240		
	SMS-EMOA	0.825166	0.037723	0.726366	0.868299	0.063341	0.233652	0.003403	1.285944		
ZDT6	Δ_p -DE	0.502570	0.000707	0.501341	0.503668	0.005680	0.000539	0.004472	0.006862		
	NSGA-II	0.4823269	0.002835	0.474724	0.487819	0.005182	0.000493	0.004307	0.006153		
	SMS-EMOA	0.494348	0.001341	0.490871	0.496709	0.003790	0.000456	0.002963	0.004630		

Table B.4.: Results for the ZDT functions: I_H and I_{SS}

Problem	Algorithm A	$I_{e^+}(A, B)$			$I_C(A, B)$		
		Algorithm B					
		Δ_p -DE	NSGA-II	SMS-EMOA	Δ_p -DE	NSGA-II	SMS-EMOA
ZDT1	Δ_p -DE	—	0.022041	0.022795	—	0.298333	0.028666
	NSGA-II	0.011942	—	0.011608	0.000000	—	0.000666
	SMS-EMOA	0.005351	0.0041833	—	0.000666	0.295333	—
ZDT2	Δ_p -DE	—	0.086318	0.086961	—	0.359000	0.044333
	NSGA-II	0.012642	—	0.012685	0.000000	—	0.000000
	SMS-EMOA	0.005493	0.003877	—	0.000000	0.365333	—
ZDT3	Δ_p -DE	—	0.015407	0.017324	—	0.069000	0.008666
	NSGA-II	0.006514	—	0.007984	0.546666	—	0.004000
	SMS-EMOA	0.0186566	0.0185636	—	0.664333	0.287000	—
ZDT4	Δ_p -DE	—	0.018946	0.019906	—	0.154000	0.132000
	NSGA-II	0.040917	—	0.035103	0.499000	—	0.181666
	SMS-EMOA	0.118952	0.109213	—	0.507333	0.341666	—
ZDT6	Δ_p -DE	—	0.006141	0.007059	—	0.982333	0.893000
	NSGA-II	0.028039	—	0.017023	0.000000	—	0.000000
	SMS-EMOA	0.012488	0.001267	—	0.000000	0.949666	—

Table B.5.: Results for the ZDT functions: I_{e^+} and I_C

Problem	Algorithm	GD indicator					IGD indicator				
		μ	σ	min	max	μ	σ	min	max		
DTLZ1	Δ_p -DE	0.323884	0.462850	0.017093	2.516848	0.276645	0.142848	0.031439	2.069703		
	NSGA-II	10.204714	4.398068	2.176226	20.066518	3.911751	1.438019	1.621596	7.568594		
	SMS-EMOA	1.4261365	0.689273	0.393214	3.161340	0.861782	0.460328	0.128425	2.093355		
DTLZ2	Δ_p -DE	0.036068	0.002250	0.030211	0.040984	0.074523	0.029538	0.054866	0.186947		
	NSGA-II	0.038937	0.001065	0.037127	0.041966	0.074047	0.003411	0.069283	0.084555		
	SMS-EMOA	0.029501	0.000946	0.027559	0.031133	0.063697	0.000645	0.061957	0.064884		
DTLZ3	Δ_p -DE	0.657818	0.744609	0.050107	2.160001	0.653419	0.723754	0.069364	2.143787		
	NSGA-II	32.001436	13.401155	9.709050	59.493846	13.160544	4.955068	5.120912	26.040332		
	SMS-EMOA	11.754814	4.750187	2.391411	22.082736	5.525270	1.728886	1.260405	9.276372		
DTLZ4	Δ_p -DE	0.036425	0.002705	0.030526	0.044288	0.081691	0.071770	0.055801	0.451335		
	NSGA-II	0.036245	0.005930	0.023933	0.040727	0.165438	0.188414	0.065258	0.550433		
	SMS-EMOA	0.029985	0.000846	0.028222	0.031707	0.063790	0.000767	0.062314	0.065204		
DTLZ5	Δ_p -DE	0.297979	0.030226	0.237744	0.376499	0.120641	0.020485	0.096230	0.179033		
	NSGA-II	0.003007	0.000225	0.002602	0.003578	0.005608	0.000234	0.005115	0.006125		
	SMS-EMOA	0.002466	0.000102	0.002283	0.002630	0.004923	0.000111	0.004764	0.005166		
DTLZ6	Δ_p -DE	0.0023499	0.000615	0.001497	0.003162	0.027452	0.017328	0.007655	0.067947		
	NSGA-II	1.352182	0.120997	1.185159	1.646268	1.195632	0.099224	0.985204	1.396974		
	SMS-EMOA	0.460537	0.031437	0.401485	0.544976	0.403205	0.029532	0.354244	0.486935		
DTLZ7	Δ_p -DE	0.027085	0.005062	0.008915	0.036357	0.102124	0.028188	0.082157	0.235796		
	NSGA-II	0.025068	0.003382	0.018028	0.034535	0.085428	0.049937	0.066644	0.352927		
	SMS-EMOA	0.005445	0.000350	0.004724	0.006345	0.110085	0.094605	0.071378	0.351797		

Table B.6.: Results for DTLZ functions: I_{GD_1} and I_{IGD_1}

Problem	Algorithm	Hypervolume indicator					Spacing indicator				
		μ	σ	min	max		μ	σ	min	max	
DTLZ1	Δ_p -DE	1.095670	0.277145	0.000000	1.297963		0.063178	0.041419	0.030983	0.248544	
	NSGA-II	0.000000	0.000000	0.000000	0.000000		3.712455	4.157084	0.155584	14.373336	
	SMS-EMOA	0.399135	0.44120	0.000000	1.241754		0.909329	2.549488	0.046284	13.527007	
DTLZ2	Δ_p -DE	0.701161	0.045387	0.538906	0.734356		0.057959	0.022581	0.030352	0.118945	
	NSGA-II	0.690814	0.007487	0.668839	0.700561		0.055372	0.004692	0.048047	0.064445	
	SMS-EMOA	0.757763	0.000071	0.757640	0.757883		0.041697	0.002087	0.036532	0.046268	
DTLZ3	Δ_p -DE	120.4821	5.729501	106.4878	124.3497		0.079759	0.039443	0.041049	0.183004	
	NSGA-II	0.169759	0.914182	0.000000	5.092783		12.30287	8.778941	0.606102	39.12390	
	SMS-EMOA	11.16606	25.67012	0.000000	116.5334		10.44783	11.70420	0.250234	38.52293	
DTLZ4	Δ_p -DE	0.700258	0.059847	0.422175	0.734088		0.058061	0.017011	0.037069	0.101273	
	NSGA-II	0.650300	0.095096	0.459927	0.708710		0.045600	0.018925	0.007515	0.063220	
	SMS-EMOA	0.757853	0.000058	0.757726	0.757978		0.040868	0.001912	0.036925	0.044869	
DTLZ5	Δ_p -DE	0.29337	0.02251	0.232877	0.327952		0.051516	0.051516	0.040141	0.067209	
	NSGA-II	0.437321	0.000231	0.436823	0.437769		0.009530	0.000704	0.008106	0.010800	
	SMS-EMOA	0.439354	0.000010	0.439333	0.439376		0.008762	0.000514	0.007877	0.009600	
DTLZ6	Δ_p -DE	0.416980	0.023451	0.363971	0.436641		0.020400	0.003478	0.015581	0.032215	
	NSGA-II	0.000000	0.000000	0.000000	0.000000		0.121257	0.027979	0.081400	0.238123	
	SMS-EMOA	0.090737	0.017132	0.051771	0.125026		0.050669	0.010143	0.037919	0.096115	
DTLZ7	Δ_p -DE	2.921157	0.088830	2.557450	3.014050		0.101388	0.011420	0.080391	0.123283	
	NSGA-II	2.933973	0.072878	2.555335	2.990868		0.069763	0.010317	0.031138	0.089563	
	SMS-EMOA	3.041450	0.000924	2.632969	3.105699		0.059117	0.006228	0.042009	0.067219	

Table B.7.: Results for the DTLZ functions: I_H and I_{SS}

Problem	Algorithm A	$I_{e^+}(A, B)$			$I_c(A, B)$		
		Algorithm B					
		Δ_p -DE	NSGA-II	SMS-EMOA	Δ_p -DE	NSGA-II	SMS-EMOA
DTLZ1	Δ_p -DE	—	0.043678	0.086826	—	0.809333	0.825000
	NSGA-II	3.069412	—	2.952188	0.000000	—	0.020666
	SMS-EMOA	0.60607	0.022639	—	0.779666	0.708333	—
DTLZ2	Δ_p -DE	—	0.142578	0.169552	—	0.037666	0.000000
	NSGA-II	0.178579	—	0.182433	0.008666	—	0.000000
	SMS-EMOA	0.039952	0.040808	—	0.048666	0.589666	—
DTLZ3	Δ_p -DE	—	0.046414	0.046414	—	0.856666	0.876666
	NSGA-II	11.78089	—	9.015845	0.000000	—	0.028000
	SMS-EMOA	5.128037	0.350753	—	0.000000	0.848666	—
DTLZ4	Δ_p -DE	—	0.153341	0.179317	—	0.035000	0.000000
	NSGA-II	0.244583	—	0.263707	0.032000	—	0.001333
	SMS-EMOA	0.038625	0.036893	—	0.087666	0.086666	—
DTLZ5	Δ_p -DE	—	0.165856	0.166575	—	0.000000	0.000000
	NSGA-II	0.000001	—	0.010253	0.942000	—	0.000333
	SMS-EMOA	0.000012	0.003437	—	0.957333	0.121666	—
DTLZ6	Δ_p -DE	—	0.053993	0.053976	—	0.979333	0.982000
	NSGA-II	0.947063	—	0.719401	0.000000	—	0.000000
	SMS-EMOA	0.349647	0.000139	—	0.000000	0.942000	—
DTLZ7	Δ_p -DE	—	0.153956	0.169901	—	0.107333	0.003000
	NSGA-II	0.202031	—	0.209930	0.053000	—	0.000000
	SMS-EMOA	0.205410	0.210882	—	0.213333	0.256333	—

Table B.8.: Results for the DTLZ functions: I_{e^+} and I_c

B.3 SCALABILITY RESULTS

Tables B.9 and B.10 summarize our results for DTLZ2 problem with up to 10 objectives using the GD and hypervolume indicators (see Chapter 4). The running time for each algorithm is presented in table B.11

The reference point used for the hypervolume indicator was $(\vec{1}, \vec{1})$.

# objectives	Algorithm	GD indicator			
		μ	σ	min	max
2	Δ_p -DE	0.0000004	0.0000012	0.000000	0.000004
	NSGA-II	0.0005877	0.000130	0.000481	0.000952
	SMS-EMOA	0.0000329	0.000049	0.000004	0.000136
3	Δ_p -DE	0.000045	0.000013	0.000026	0.000069
	NSGA-II	0.0146852	0.001583	0.011695	0.016491
	SMS-EMOA	0.000102	0.000014	0.000073	0.000132
4	Δ_p -DE	0.000991	0.001589	0.000053	0.005141
	NSGA-II	0.0460704	0.006683	0.038534	0.059021
	SMS-EMOA	0.000284	0.000031	0.000233	0.000325
5	Δ_p -DE	0.018994	0.020466	0.001042	0.067794
	NSGA-II	0.1497307	0.021936	0.116644	0.195061
	SMS-EMOA	0.000444	0.000061	0.000354	0.000553
6	Δ_p -DE	0.018027	0.005456	0.011671	0.026870
	NSGA-II	1.455277	0.179236	1.072336	1.677652
	SMS-EMOA	—	—	—	—
7	Δ_p -DE	0.065257	0.050900	0.008182	0.190008
	NSGA-II	3.214524	0.191547	2.885911	3.479824
	SMS-EMOA	—	—	—	—
8	Δ_p -DE	0.034512	0.027262	0.001571	0.097806
	NSGA-II	3.081084	0.097669	2.885206	3.233306
	SMS-EMOA	—	—	—	—
9	Δ_p -DE	0.016333	0.004926	0.010517	0.028624
	NSGA-II	2.393702	0.090556	2.260907	2.581568
	SMS-EMOA	—	—	—	—
10	Δ_p -DE	0.014894	0.014259	0.010517	0.028624
	NSGA-II	1.7166391	0.061353	1.599913	1.786154
	SMS-EMOA	—	—	—	—

Table B.9.: Results for the DTLZ2 function: I_{GD}

COMPLETE RESULTS

# objectives	Algorithm	Hypervolume indicator			
		μ	σ	min	max
2	Δ_p -DE	0.422147	0.000261	0.421522	0.422310
	NSGA-II	0.421928	0.000071	0.421792	0.422029
	SMS-EMOA	0.422791	0.000007	0.422776	0.422800
3	Δ_p -DE	0.760359	0.001291	0.757998	0.762954
	NSGA-II	0.724661	0.003624	0.718817	0.730217
	SMS-EMOA	0.7728841	0.000037	0.772829	0.772935
4	Δ_p -DE	1.042166	0.003703	1.035251	1.048438
	NSGA-II	0.921006	0.008616	0.907570	0.930972
	SMS-EMOA	1.071800	0.000034	1.071761	1.071854
5	Δ_p -DE	1.269987	0.016897	1.240539	1.288388
	NSGA-II	1.033873	0.015736	0.999701	1.053882
	SMS-EMOA	1.330860	0.000110	1.330661	1.331089
6	Δ_p -DE	1.452230	0.0110470	1.437443	1.473233
	NSGA-II	0.330570	0.101391	0.178405	0.511004
	SMS-EMOA	—	—	—	—
7	Δ_p -DE	1.665510	0.058751	1.530023	1.742427
	NSGA-II	0.036647	0.031180	0.000302	0.095066
	SMS-EMOA	—	—	—	—
8	Δ_p -DE	1.8696093	0.053191	1.768818	1.942596
	NSGA-II	0.0170008	0.019864	0.001130	0.070408
	SMS-EMOA	—	—	—	—
9	Δ_p -DE	2.034205	0.041145	1.945704	2.086265
	NSGA-II	0.0481003	0.021721	0.018862	0.088790
	SMS-EMOA	—	—	—	—
10	Δ_p -DE	2.138645	0.014621	2.118968	2.153989
	NSGA-II	0.183653	0.036656	0.140166	0.248981
	SMS-EMOA	—	—	—	—

Table B.10.: Results for the DTLZ2 function: I_H

B.3 SCALABILITY RESULTS

# obj.	Algorithm	Running time (seconds)			
		μ	σ	min	max
2	Δ_p -DE	2.8348855	0.052186	2.684505	2.866449
	NSGA-II	0.4555398	0.002258	0.450883	0.458300
	SMS-EMOA	36.2628864	0.108132	36.144054	36.531175
3	Δ_p -DE	6.716288	0.455965	5.987896	7.243083
	NSGA-II	0.703396	0.004185	0.696468	0.710417
	SMS-EMOA	1052.4071	17.68631	1036.664953	1091.894354
4	Δ_p -DE	10.593524	1.032320	8.753984	11.874017
	NSGA-II	0.894355	0.0026259	0.891668	0.900697
	SMS-EMOA	23338.339	263.5205	23038.191662	23884.258040
5	Δ_p -DE	16.361607	1.433969	13.765632	18.676352
	NSGA-II	1.049597	0.004649	1.039976	1.056847
	SMS-EMOA	509808.04	15598.21	480578.229004	541586.485339
6	Δ_p -DE	120.4086241	16.142403	93.437694	146.156669
	NSGA-II	1.3048848	0.008647	1.296380	1.327472
	SMS-EMOA	—	—	—	—
7	Δ_p -DE	88.316456	11.686119	72.004666	107.063486
	NSGA-II	1.5645563	0.010703	1.541526	1.581247
	SMS-EMOA	—	—	—	—
8	Δ_p -DE	427.0854794	69.953813	292.025018	535.883238
	NSGA-II	1.8024511	0.0143106	1.776279	1.825986
	SMS-EMOA	—	—	—	—
9	Δ_p -DE	2574.167783	677.9899	1462.103560	3430.646654
	NSGA-II	2.0293899	0.020010	1.998924	2.066521
	SMS-EMOA	—	—	—	—
10	Δ_p -DE	10936.5007	3297.1294	7010.875822	18202.064041
	NSGA-II	2.2535963	0.0118404	2.229482	2.269007
	SMS-EMOA	—	—	—	—

Table B.11.: Results for the DTLZ2 function: running time

COMPLETE RESULTS

BIBLIOGRAPHY

- [1] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the Hypervolume Indicator: Optimal $\{\mu\}$ -Distributions and the Choice Of The Reference Point. In *FOGA '09: Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102, Orlando, Florida, USA, January 2009. ACM. (Cited on page 26.)
- [2] Johannes Bader, Dimo Brockhoff, Samuel Welten, and Eckart Zitzler. On Using Populations of Sets in Multiobjective Optimization. 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 140–154. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009. (Cited on pages 45 and 46.)
- [3] Johannes M. Bader. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. PhD thesis, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology Zürich, Zürich, Switzerland, 2009. (Cited on pages 44 and 45.)
- [4] Matthieu Basseur and Edmund K. Burke. Indicator-Based Multi-Objective Local Search. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3100–3107, Singapore, September 2007. IEEE Press. (Cited on page 38.)
- [5] Nicola Beume, Carlos M. Fonseca, Manuel Lopez-Ibanez, Luis Paquete, and Jan Vahrenhold. On the Complexity of Computing the Hypervolume Indicator. *IEEE Transactions on Evolutionary Computation*, 13(5):1075–1082, October 2009. (Cited on page 25.)

Bibliography

- [6] Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007. (Cited on page 40.)
- [7] Nicola Beume, Boris Naujoks, Mike Preuss, Günter Rudolph, and Tobias Wagner. Effects of 1-Greedy S-Metric-Selection on Innumerably Large Pareto Fronts. 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 21–35. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009. (Cited on page 40.)
- [8] Nicola Beume, Boris Naujoks, and Guenter Rudolph. SMS-EMOA - Effective Evolutionary Multiobjective Optimization. *AT-Automatisierungstechnik*, 56(7):357–364, 2008. (Cited on page 40.)
- [9] Lucas Bradstreet, Luigi Barone, and Lyndon While. Maximising Hypervolume for Selection in Multi-objective Evolutionary Algorithms. In *2006 IEEE Congress on Evolutionary Computation (CEC'2006)*, pages 6208–6215, Vancouver, BC, Canada, July 2006. IEEE. (Cited on page 40.)
- [10] Lucas Bradstreet, Lyndon While, and Luigi Barone. Incrementally Maximising Hypervolume for Selection in Multi-Objective Evolutionary Algorithms. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3203–3210, Singapore, September 2007. IEEE Press. (Cited on page 40.)
- [11] Lucas Bradstreet, Lyndon While, and Luigi Barone. A Fast Incremental Hypervolume Algorithm. *IEEE Transactions on Evolutionary Computation*, 12(6):714–723, December 2008. (Cited on page 40.)
- [12] Karl Bringmann and Tobias Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *Proceedings of the 19th International Symposium on Algorithms and Computation*, pages 436–447, Berlin, Heidelberg, 2008. Springer-Verlag. (Cited on pages 26 and 46.)

- [13] Karl Bringmann and Tobias Friedrich. Approximating the Least Hypervolume Contributor: NP-Hard in General, But Fast in Practice. 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 6–20. Springer. Lecture Notes in Computer Science Vol. 5467, Nantes, France, April 2009. (Cited on page 46.)
- [14] Charles R. Darwin. *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*. Penguin Books Ltd., New York, USA, 1859. (Cited on page 12.)
- [15] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. KanGAL report 200001, Indian Institute of Technology, Kanpur, India, 2000. (Cited on pages 21 and 38.)
- [16] 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. (Cited on pages 55 and 59.)
- [17] Francis Ysidro Edgeworth. *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. C. Kegan Paul & Co., 1881. (Cited on page 5.)
- [18] Agoston E. Eiben and James E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, 2003. (Cited on page 14.)
- [19] Michael Emmerich, Nicola Beume, and Boris Naujoks. An EMO Algorithm Using the Hypervolume Measure as Selection Criterion. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 62–76, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410. (Cited on pages 21 and 38.)

Bibliography

- [20] Jonathan E. Fieldsend, Richard M. Everson, and Sameer Singh. Using Unconstrained Elite Archives for Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 7(3):305–323, June 2003. (Cited on page 34.)
- [21] Mark Fleischer. The Measure of Pareto Optima. Applications to Multiobjective Metaheuristics. 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 519–533, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632. (Cited on pages 25 and 46.)
- [22] Lawrence J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley, New York, USA, 1966. (Cited on page 18.)
- [23] 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. (Cited on page 20.)
- [24] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1989. (Cited on page 20.)
- [25] Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, Summer 2001. (Cited on page 41.)
- [26] John H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, 1975. (Cited on page 18.)
- [27] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg. A Niche Pareto Genetic Algorithm for Multiobjective Optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World*

- Congress on Computational Intelligence*, volume 1, pages 82–87, Piscataway, New Jersey, June 1994. IEEE Service Center. (Cited on page 20.)
- [28] Simon Huband, Phil Hingston, Lyndon White, and Luigi Barone. An Evolution Strategy with Probabilistic Mutation for Multi-Objective Optimisation. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 3, pages 2284–2291, Canberra, Australia, December 2003. IEEE Press. (Cited on page 35.)
- [29] Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, Spring 2007. (Cited on page 41.)
- [30] Christian Igel, Thorsten Suttrop, and Nikolaus Hansen. Steady-State Selection and Efficient Covariance Matrix Update in the Multi-objective CMA-ES. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 171–185, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403. (Cited on page 43.)
- [31] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Iterative Approach to Indicator-Based Multiobjective Optimization. In *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, pages 3967–3974, Singapore, September 2007. IEEE Press. (Cited on page 40.)
- [32] Hisao Ishibuchi, Noritaka Tsukamoto, Yuji Sakane, and Yusuke Nojima. Indicator-Based Evolutionary Algorithm with Hypervolume Approximation by Achievement Scalarizing Functions. In *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation (GECCO'2010)*, pages 527–534, Portland, Oregon, USA, July 7–11 2010. ACM Press. ISBN 978-1-4503-0072-8. (Cited on page 40.)
- [33] Kenneth Alan De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, Ann Arbor, MI, USA, 1975. (Cited on page 16.)

Bibliography

- [34] Joshua Knowles and David Corne. On Metrics for Comparing Nondominated Sets. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716, Piscataway, New Jersey, May 2002. IEEE Service Center. (Cited on pages 28 and 46.)
- [35] Joshua Knowles and David Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003. (Cited on pages 26, 34, and 38.)
- [36] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000. (Cited on page 21.)
- [37] Joshua D. Knowles, David W. Corne, and Mark Fleischer. Bounded Archiving using the Lebesgue Measure. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2490–2497, Canberra, Australia, December 2003. IEEE Press. (Cited on pages 34 and 38.)
- [38] Ailsa H. Land and Alison G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960. (Cited on page 8.)
- [39] Gregor J. Mendel. Versuche über pflanzen-hybriden. *Proceedings of the Natural History Society of Brünn*, 1866. (Cited on page 12.)
- [40] Kaisa M. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, Massachusetts, EE.UU., 1998. (Cited on pages 7, 9, and 10.)
- [41] Boris Naujoks, Nicola Beume, and Michael Emmerich. Multiobjective Optimization using S-metric Selection: Application to three-dimensional Solution Spaces. In *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, volume 2, pages 1282–1289, Edinburgh, Scotland, September 2005. IEEE Service Center. (Cited on page 39.)

- [42] Vilfredo Pareto. *Cours d'économie politique*. F. Rouge, 1896. (Cited on page 5.)
- [43] Kenneth Price, Rainer Storn, and Jouni Lampinen. *Differential Evolution A Practical Approach to Global Optimization*. Natural Computing Series. Springer-Verlag, Berlin, Germany, 2005. (Cited on page 48.)
- [44] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. PhD thesis, TU Berlin, 1971. (Cited on page 17.)
- [45] Günter Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks*, 5(1):96–101, january 1994. (Cited on page 18.)
- [46] J. David Schaffer. *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, 1984. (Cited on page 20.)
- [47] Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995. (Cited on page 29.)
- [48] Oliver Schütze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Measuring the Averaged Hausdorff Distance to the Pareto Front of a Multi-objective Optimization Problem. Technical Report TR-OS-2010-02, CINVESTAV, 2010. (Cited on pages 27, 28, and 29.)
- [49] N. Srinivas and Kalyanmoy Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, Fall 1994. (Cited on pages 20 and 29.)
- [50] E.L. Ulungu, J. Teghem, Ph. Fortemps, and D. Tuyttens. MOSA Method: A Tool for Solving Multiobjective Combinatorial Optimization Problems. *Journal of Multi-Criteria Decision Analysis*, 8(4):221–236, 1999. (Cited on page 32.)

Bibliography

- [51] 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. (Cited on page 27.)
- [52] Thomas Voß, Nicola Beume, Günter Rudolph, and Christian Igel. Scalarization Versus Indicator-Based Selection in Multi-Objective CMA Evolution Strategies. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 3041–3048, Hong Kong, June 2008. IEEE Service Center. (Cited on page 43.)
- [53] Tobias Wagner and Heike Trautmann. Integration of Preferences in Hypervolume-Based Multiobjective Evolutionary Algorithms by Means of Desirability Functions. *IEEE Transactions on Evolutionary Computation*, 14(5):688–701, October 2010. (Cited on page 43.)
- [54] Lyndon While. A New Analysis of the LebMeasure Algorithm for Calculating Hypervolume. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 326–340, Guanajuato, México, March 2005. Springer. Lecture Notes in Computer Science Vol. 3410. (Cited on page 26.)
- [55] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicator Via Weighted Integration. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization, 4th International Conference, EMO 2007*, pages 862–876, Matshushima, Japan, March 2007. Springer. Lecture Notes in Computer Science Vol. 4403. (Cited on pages 25 and 26.)
- [56] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000. (Cited on pages 55 and 56.)

- [57] Eckart Zitzler, Joshua Knowles, and Lothar Thiele. Quality Assessment of Pareto Set Approximations. In Jürgen Branke, Kalyanmoy Deb, Kaisa Miettinen, and Roman Slowinski, editors, *Multiobjective Optimization. Interactive and Evolutionary Approaches*, pages 373–404. Springer. Lecture Notes in Computer Science Vol. 5252, Berlin, Germany, 2008. (Cited on pages 24 and 32.)
- [58] Eckart Zitzler and Simon Künzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, volume LNCS 3242, pages 832–842, Birmingham, UK, September 2004. Springer-Verlag. (Cited on pages 21 and 36.)
- [59] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001. (Cited on page 21.)
- [60] Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag. (Cited on pages 25 and 32.)
- [61] 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, November 1999. (Cited on pages 20 and 21.)
- [62] Eckart Zitzler, Lothar Thiele, and Johannes Bader. On Set-Based Multiobjective Optimization. Technical Report 300, Computer Engineering and Networks Laboratory, ETH Zurich, February 2008. (Cited on pages 30 and 44.)
- [63] Eckart Zitzler, Lothar Thiele, and Johannes Bader. SPAM: Set Preference Algorithm for Multiobjective Optimization. In Günter Rudolph, Thomas Jansen, Simon Lucas, Carlo Poloni, and Nicola Beume, editors, *Parallel Problem Solving from Nature—PPSN X*, pages 847–858. Springer.

Bibliography

Lecture Notes in Computer Science Vol. 5199, Dortmund, Germany, September 2008. (Cited on pages [26](#) and [44](#).)

- [64] Eckart Zitzler, Lothar Thiele, and Johannes Bader. On Set-Based Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 14(1):58–79, February 2010. (Cited on page [44](#).)
- [65] 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. (Cited on pages [24](#), [25](#), [30](#), and [46](#).)

Bibliography