On the Evaluation of the Quality of Non–dominated Sets

Giovanni Lizárraga Lizárraga

Doctorado en Ciencias con Orientación en Ciencias de la Computación

Center for Research in Mathematics A.C.

April 2009

Abstract

There are many published multi–objective evolutionary algorithms (also known as MOEAs). A natural question is what algorithm performs better? If we ignore other factors such as computational complexity, the evaluation of the performance of a MOEA only depends on the output of the algorithm. The output of a MOEA is a set of vectors (usually known as non–dominated sets) with some special properties derived from the Pareto Optimality Criteria (POC). Unfortunately, evaluating and comparing these non–dominated sets is not an easy task and is an open research problem.

Many performance measures have been proposed in the past, but they are sensitive to misleading cases and sometimes, hard to use. Some theoretical studies have been developed in order to determine what we want from a good performance measure. Unfortunately, many of the performance measures derived from those studies are too conservative and have a very limited capacity to distinguish between good and bad sets.

The goal of this thesis is to analyze the problem and introduce a new method for the evaluation of non–dominated sets, named G–Ranker. The G–Ranker is designed having in mind most of the desired properties of a non–dominated set, needs no extra information about the multi–objective problem and is more robust to misleading cases compared to other methods.

Also, we introduce a set of test cases to evaluate the effectiveness of a performance measure. The results of the experiments demonstrate the superiority of the G–Ranker with respect to state–of–the–art approaches reported in specialized literature.

A mi madre, Evangelina Lizárraga Lizárraga.

Agradecimientos

Hay tanta gente a quien agradecer que no quisiera no excluir a nadie, pero si alguien me faltó, espero me disculpe.

Primero agradezco a mis padres por todo lo que hicieron por mí, empezando por haberme hecho a mí. También a mis hermanos por su apoyo.

Segundo a mi asesor Arturo Hernández por la guía y la oportunidad.

A todo el personal de CIMAT, tanto académicos como administrativos por su soporte y el excelente trabajo que hacen. Gracias a eso, uno solo tiene que preocuparse por hacer ciencia y no por otras cosas.

A mis compañeros de cubículo Alonso, Angel, Justino, Gerardo, Rogelio y Karen. Hicieron que el trabajo de investigación la cosa más divertida del mundo.

A Alejandra, que es una de las mejores cosas que me han pasado en la vida.

A Judith e Ivette por ser una inspiración.

A todas mis amistades en Guanajuato, que hicieron mi vida más amena fuera del ambiente en CIMAT. Si leen esto sabrán que me refiero a ustedes.

Al CIMAT como institución, por toda la ayuda y por ser el mejor lugar para hacer investigación.

Un agradecimiento especial al Doctor Miguel Angel Moreles por todas sus asesorías respecto a matemáticas y en especial la revisión de una de mis demostraciones referente a teoría de conjuntos.

Un enorme agradecimiento al Consejo Nacional de Ciencia y Tecnología (CONACYT), por todo su apoyo para mis estudios de maestría y doctorado bajo el número de becario 159170. Sin este apoyo hubiera sido imposible la realización de esta investigación.

Y finalmente, a todos los que me faltaron.

Contents

List of Figures

Chapter 1

Introduction

In this document we study the problem of how to evaluate the quality of a non–dominated set, or how to compare the relative quality of several non–dominated sets. This problem has become an important topic of research in multi–objective optimization using the Pareto Optimality Criteria (POC). In multi–objective optimization using POC, we approximate a special set of vectors known as the Pareto Set. This set is important because, under a few assumptions, it contains all possible optimal solutions that a Decision Maker may want. Unfortunately, it is not possible to obtain the complete Pareto Set in the general case, so it is necessary to make an approximation using a non–dominated set. This way, it is more likely for a Decision Maker, to find an acceptable solution.

Many special algorithms have been created to approximate the Pareto Set. It is natural to wonder which algorithm is better than the others in the range of problems they are supposed to solve[1]. As a consequence, a new area of research has arisen: the comparison of the performance of two or more multi–objective algorithms. One of the most important aspects to consider when comparing two algorithms is the quality of their outputs. In our case,

[1]It has been demonstrated that the expected performance of all optimization algorithms are equal [74], but if we are interested only in a subset of problems, some algorithms may perform better than others.

the output of an algorithm is a non–dominated set, so, an important part in the comparison of two algorithms is the comparison of non–dominated sets.

Unfortunately, the evaluation of the quality of non–dominated sets is a difficult task. Many methodologies, proposed to this end, in some cases contradict the intuitive notion of when a non–dominated set is better than others. Part of the problem is the lack of a proper definition of quality regarding non–dominated sets. Several theoretical studies have been developed to try to identify the desirable properties of both a non–dominated set and a methodology to evaluate non–dominated sets. But, still there is no definitive method that is universally accepted, so an active area of research is to find more accurate methodologies to evaluate non–dominated sets.

We are interested in a special class of algorithms that approximate the Pareto Set, known as multi–objective evolutionary algorithms. MOEAs are stochastic algorithms, therefore, repeated runs of the algorithms may result in different non–dominated sets. As a consequence, a single run is not enough to compare two algorithms. So, the comparison of two MOEAs must be based on several runs from both algorithms. In [43], a methodology to make this kind of analysis is proposed. The basic steps of this methodology are:

1. Run each MOEA several times.
2. Combine the outputs of all MOEAs in a set of m non–dominated sets.
3. Rank the m sets.
4. Decide which MOEA is the best based on a non–parametric analysis of the ranked sets.

Non–parametric analysis is a well developed area in statistics, so the main area of opportunity here is how to rank m non–dominated sets. The goal of this work is to develop a method to evaluate the elements of a list of non–dominated sets, that models our intuitive notion of when one non–dominated set is better than another. In order to attain this goal, we review

some of the most important articles related to the evaluation of the quality of non–dominated sets and make a revision of several basic concepts related to optimization and operation research in order to identify what makes a non–dominated set useful.

Some of the most important results from this work are the following:

1. We create a method to compare non–dominated sets that combines diversity and convergence and has most of the desirable properties for a quality indicator.

2. We introduce a benchmark that is useful to study the behavior of quality indicators.

3. We show that, contrary to what is widely believed, unary quality indicators are not inferior to binary ones, and that being both compatible and complete is not desirable for a quality indicator.

The rest of the document is organized in the following way: in Chapter 2, we present some basic concepts of problem solving. In Chapter 3, we review the main concepts about performance measures for non–dominated sets and some of the most popular quality indicators. In Chapter 4, we state the problem we address in this work. In Chapter 5, we present our quality indicator. In Chapter 6, we present some test cases to evaluate the robustness of a comparison method and the result of the experimentation. In Chapter 7, we explore some details and ideas for future research based on this work. Finally in Chapter 8, we state our conclusions.

Chapter 2

Basic Concepts

2.1 Introduction

In daily life, we make decisions all the time. Some of these decisions are easily made and do not require a lot of time or other resources, for example: choosing a pair of shoes. Other decisions may need more time and effort, like buying a new car. The goal of making decisions is to choose good solutions to a problem, for instance, what pair of shoes combines better with my clothes? or where am I going to live?

Sometimes, we are satisfied with any solution that fulfills some requirements, for example, any working car that cost less then $5000.00 is good enough to satisfy our needs. Sometimes, we want the "best" solution to the problem, for example:

Example 1. *We want the cheapest apartment in our city whose distance from our office is less than 10 kilometers.*

When we are looking for a solution, it is common to have a set of possible candidates to choose from. We call the set of candidate solutions "the search space". These candidate solutions have some attributes of interest, depending on the problem we are solving. These attributes of interest can be divided

into constraints and objectives. In Example 1, the set of candidate solutions is all the houses for rent in the city and the attributes of interest are "distance from our workplace" and "cost of the rent". The attribute "distance from the workplace" is a constraint while the attribute "cost of the rent" is an objective.

The procedure we use to solve a problem may depend on how clear our objective is. If we are looking for a birthday present but we are not sure of what we want, an option is to enter a gift store and look for the options. In this case, we first explore the options and then decide what we want. In other cases, like Example 1, we know in advance exactly what solution we want, "the cheapest apartment in the city no more than 10 kilometers from our workplace", the problem is to find that apartment.

The examples given above, like finding an apartment or a pair of shoes, are useful to introduce some of the basic concepts of optimization, but in this thesis we are interested in more complex problems that arise in science, engineering, economy and other areas of human activity. Problems that can be represented in a "mathematical way", so the search space, constraints and other concepts can be represented by numbers, vectors, functions and other mathematical objects.

In the rest of this chapter we describe some basic concepts related to some areas of applied mathematics and engineering, such as numerical optimization and operations research. Operations research deals with the complete process of problem solving, considering some steps like:

List 1.

1. *Identifying the problem*
2. *Modeling*
3. *Mathematical optimization*
4. *Decision making*

5. Implementation

Modeling (item 2 in List 1) is sometimes the most important step in problem solving, according to [53], page 1. Modeling is the construction of a mathematical representation of the system we want to work on. We review some basic concepts about modeling because they are very relevant in problem solving. Mathematical optimization (item 3) consists of applying powerful mathematical tools on a model to find the optimal solution for a problem. When solving a problem, we have constraints in time and other resources, so choosing the best tool to solve a problem is very important, because some tools may work better than others. This issue is related to the problem we are dealing with. There is a class of algorithms known as multi–objective evolutionary algorithms (MOEAs), designed to solve a special class of problems. Choosing the best MOEAs is, itself, a problem related to decision making (item 4). When the candidate solutions have several, equally important, attributes of interest, we are dealing with multi–criteria decision making, also known as multi–objective decision making. We claim that the key to creating a criteria to evaluate MOEAs must be based on a good understanding of the basic concepts of multi–criteria decision making. Item 5, implementation of the solution, is beyond the scope of this work and will not be discussed here.

2.2 Modeling

When designing an airplane, it is common for aeronautic engineers to construct a maquette of the airplane in order to do experiments and make adjustments to the design. There are several reasons to construct maquettes, for example:

List 2.

1. *It is cheaper and faster to construct a maquette and do experiments on it than constructing the real plane for the experimentation.*

2. *A maquette is a lot easier to manipulate than a real plane.*

3. *A well constructed maquette contains the attributes of interest for the engineers, or at least has a behavior similar enough to that of the real plane. So, we can make valid inferences about the design based on the maquette.*

Note that item 3 in List 2, implies that a maquette is not a perfect representation of the airplane, there may be differences between the behavior of the real airplane and the maquette. There is a trade off between simplicity and accuracy. In this example, the maquette is a model of the plane. "Models are a representations of reality"([1], page 60), and they are very helpful in problem solving to obtain solutions in less time and at a lower cost.

In operations research we also have models, they are not "physical" models, like a maquette, but abstract models. We represent the system under study as a combination of real numbers and functions. We use these mathematical models for the same reasons mentioned in List 2. For example, when we want to evaluate if a steel structure is strong enough to resist some load conditions, it is cheaper and easier to use the finite element method to represent and evaluate the characteristics of the structure than constructing the building and testing it.

A typical example of a mathematical model is parabolic movement. Consider the following example:

Example 2. *Consider a cannon on flat terrain that can be rotated vertically at angle θ, with values from θ_0 to θ_f (Figure 2.1). The cannon shoots a cannonball with an initial speed V_0. We want to find the angle of rotation for the cannon so the cannonball reaches the maximum horizontal distance from the cannon.*

In Example 2, the physical system under study is composed by the cannon, cannonball, gravity, etc. The attribute of interest is the position of the cannonball, and the search space are the different angles of rotation of the

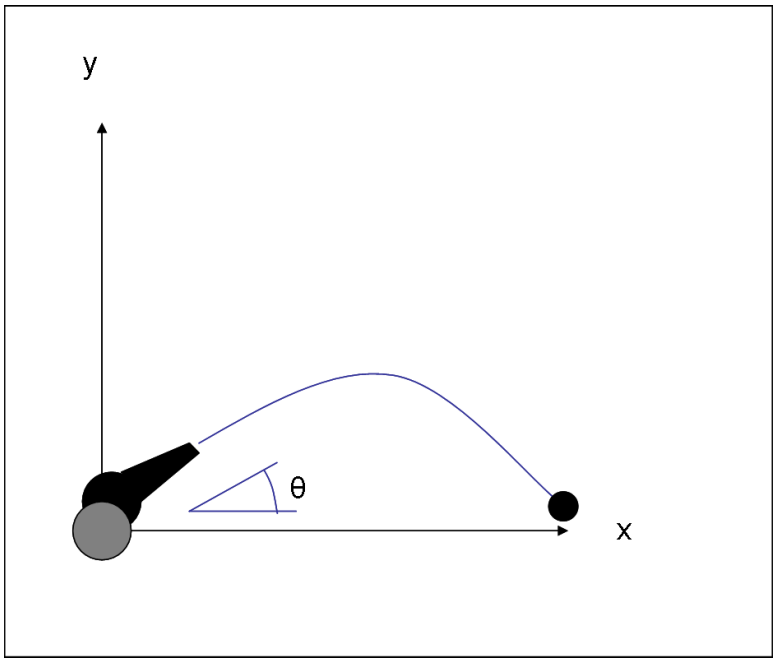

Figure 2.1: Parabolic Movement.

cannon. The angle of rotation can be represented with real numbers in the interval $[\theta_0, \theta_f]$. In order to simplify the problem, assume that the trajectory of the cannonball is contained in a vertical plane P, and define a coordinate system with the origin in the position of the cannon and the vertical and horizontal axis contained in P. Using the principles of universal gravitation we can model the position (x, y) of the cannonball with respect to the time t passed after the cannonball is shot as:

$$x(t, \theta) = V_0 \cdot t \cdot \cos\theta \tag{2.1}$$

$$y(t, \theta) = V_0 \cdot t \cdot \sin\theta - K \cdot t^2 \tag{2.2}$$

where K is a constant related to the force of gravity. Also, we have some

constraints related to the system:

$$t \geq 0 \tag{2.3}$$

$$y(t, \theta) \geq 0 \tag{2.4}$$

$$\theta_0 \leq \theta \leq \theta_f \tag{2.5}$$

Constraint 2.3 is necessary because we are only interested in what happens after the cannonball is shot. Constraint 2.4 is necessary because the ball can not go under ground level. Constraint 2.5 comes from the statement of Example 2.

Formulas 2.1–2.5 represent an acceptable mathematical model of the system under study. In order to represent the problem we want to solve, we need to create an objective function which is at the same time, a model of the desirability of a solution. In this case, the objective function is $x(t, \theta)$. In this case the definition of an objective function is trivial, but in other problems the construction of the objective function may be more difficult. So, the mathematical formulation of the problem in Example 2 is to maximize Formula 2.1 subject to the constraints represented in Formulas 2.3–2.5. This means, we need to find the value of θ[1] so the cannonball reaches the maximum horizontal distance.

Modeling is a very important issue in operations research, because we look for the optimal solution for the problem in the model. If the model is not accurate enough, the solution found in the model may have no relation with the real solution of the system.

For simplicity, in Example 2 we make some simplifications and assumptions that are not completely true. For example, we assumed that the trajectory of the cannonball is contained in a plane and we ignored other factors like the friction of the air. If we consider such factors we obtain a more

[1]The value of t that maximizes x is implicitly determined by the value of θ.

accurate, and complicated model. For instance, there are several kinds of models, for example Aris ([3], page 27) describes six types of models: verbal models, finite models, fuzzy subsets, statistical models, differential equations and stochastic models. Ackoff and Sasieni ([1], page 60) describe three kinds of models frequently used in operations research: iconic, analogue and symbolic.

Another refinement that can be done to a model is to consider the stochastic nature of the physical world. The model constructed for Example 2 was deterministic, this means that for a given value of the search variable, we obtain a unique value of the objective function. This is not true in the real world, where there is variability in the results of several experiments realized under the same conditions. There are models where the value of the objective function is affected by a distribution of probability. In this work, we only consider deterministic models to represent optimization problems.

Modeling is very important, but we have only given an overview. The concept of modeling is fundamental in the process of problem solving, we refer the reader to sources of detailed information as [11], [1], [3] and [50]

2.3 Optimization

Once a good model is obtained, the following step to solve a problem is to optimize the model to find the best solution. We call this step optimization.

There are many kinds of problems, and for different problems there are different models. We work with optimization problems that can be stated in the following way:

Optimization Problem 1.

$$Minimize\ f(x) \tag{2.6}$$

$$subject\ to$$

$$g_i(x) \leq 0,\ i = 1, \ldots, I \tag{2.7}$$

$$h_j(x) = 0,\ j = 1, \ldots, J \tag{2.8}$$

$$x \in \boldsymbol{R}^n \tag{2.9}$$

Where $f(x)$ is the objective function that represents the desirability of a solution, I is the number of inequality constraints $g_i(x)$ and J is the number of equality constraints $h_j(x)$. We state the problem so the objective function must be minimized, but in many problems the objective function must be maximized. Fortunately, any problem in the form "maximize $f(x)$" can be converted into the equivalent form "minimize $-f(x)$".

When a candidate solution x does not violate any constraint of the problem, we say that x is feasible. We denote by X the set of all feasible candidate solutions of the search space. A candidate solution that violates one or more constraints is unfeasible. So, the optimization problem can also be stated as:

Optimization Problem 2.

$$Minimize\ f(x) \tag{2.10}$$

$$subject\ to$$

$$x \in X \tag{2.11}$$

The "best" solution for an optimization problem is known as a *global optimum* and its definition is the following:

Definition 1. *A global optimum of a model in the form of Optimization Problem 1, is a point $x^* \in X$ such that $f(x^*) \leq f(x)$, for all $x \in X$.*

Note that x^* may not be unique, more than one element of X may be a

global optimum. Another important definition is local optimum, but before defining a local optimum we present some other definitions:

Definition 2. *Let r be a real number greater than zero and x a point in $\boldsymbol{R}^n$. An open ball of radius r and center $x \in \boldsymbol{R}^n$, is a subset of $\boldsymbol{R}^n$ given by $\{y \in \boldsymbol{R}^n | d(x, y) < r\}$, where $d(x, y)$ denotes the Euclidean distance between x and y.*

Definition 3. *Let Y be a subset of $\boldsymbol{R}^n$. Y is an open set in $\boldsymbol{R}^n$ if for each $y \in Y$, Y contains some open ball with center y.*

Definition 4. *A neighborhood of x is any open set that contains x.*

In this work we only use subsets of $\mathbf{R}^n$. When we mention open sets and open balls, we assume that these objects are in $\mathbf{R}^n$. Finally, we define local optimum:

Definition 5. *A local optimum of an optimization problem, is a point $x^* \in X$ such that for a neighborhood N of x^*, $f(x^*) \leq f(x)$, for all $x \in N \cap X$.*

We also call global minimum and local minimum to the global optimum and local optimum, respectively. With these definitions, we introduce a definition of optimization:

Definition 6. *Given a model of a problem, like Optimization Problem 1, optimization is the process to explore the search space in order to find or at least approximate the global optimum of the model.*

Many optimization algorithms can only guarantee to find a local optimum, not the global optimum. Other algorithms do not guarantee to find neither the global or local optimum, but are less likely to be trapped in a local optimum, thus improving the potential of finding a global optimum.

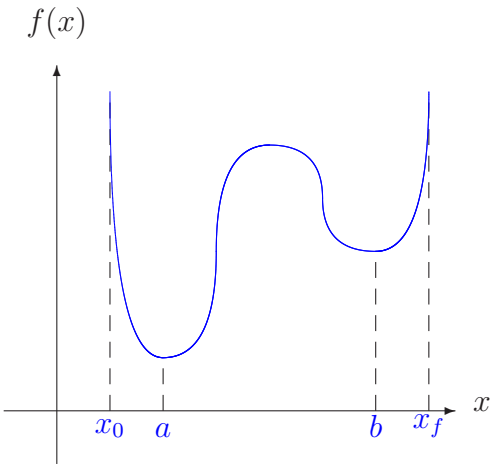

Figure 2.2: The graph of an objective function. The search space is the interval $[x_0, x_f]$. Point a is a global optimum, point b is a local optimum.

2.3.1 Types of Optimization Problems

Depending on the kind of problem, more specific information can be introduced to the model, and more efficient and effective search algorithms can be created. Optimization problems can be divided into several kinds depending on the model to optimize. Next, we present some examples.

Continuous versus Discrete Optimization

In many problems, the search space consists of variables that can take continuous values at least in some intervals. But, for other problems, the search space is composed by integer values. For these cases it is necessary to introduce the constraint:

$$x \in \mathbb{Z} \tag{2.12}$$

Constrained versus Unconstrained Optimization

When there are no constraints in the optimization problem, we have a case of unconstrained optimization. It is possible to have no constraints in some special problems, or when the model of Optimization Problem 1 is reformulated so the constraints are added to the objective function as a penalty. Constrained optimization occurs when we have one or more constraints.

Linear versus Non–linear Optimization

When the objective function and the constraints are all linear functions, we have a case of linear programming. There are many effective and efficient algorithms to solve linear programming problems. When the objective function and/or at least one of the constraints is not linear, we have a case of non–linear optimization, which usually is more complicated. A special case of non–linear optimization occurs when all constraints are linear and the objective function is quadratic. This case is known as quadratic programming.

Optimization Algorithms

There is a wide variety of algorithms that compute an optimal solution of a problem. One of the main differences between these algorithms is their search mechanism. Many of these algorithms are designed for specific kinds of problems and exploit special information depending on the model. For example, in continuous, unconstrained non–linear problems where the objective function is twice continuously differentiable, there are several methods that guarantee to find a local optimum, such as line search methods, trust region methods, Newton methods, etc.

For constrained, continuous, non–linear problems, where the objective function is twice continuously differentiable, we can use *Penalty methods* [10], [33], *Barrier methods* [26], [20], *Augmented Lagrangian methods* [36], [56], *Sequential Quadratic programming* [27], [34], [73], [57], and others.

For linear problems, we have the *Simplex Method* [11], and the *Interior Point Methods* [75]. For quadratic programming there are some algorithms such as the *Active Set Method* and the *Gradient Projection Method* [53].

For a detailed description of the algorithms mentioned here we recommend reading Nocedal and Wright [53]. One characteristic present in most search mechanisms is that they are iterative algorithms. This means that the algorithms repeat a cycle in which one or more new candidate solutions are obtained. Many search algorithms are designed in such a way that the candidate solution obtained in one iteration is better than the one in the previous iteration. Other algorithms, like genetic algorithms [32], need to compare new solutions with previous ones in order to choose the best solutions and continue the search based on the information contained in the best solutions.

Most of the algorithms just mentioned, and many others, can only be guaranteed to find a local optimum. Once they reach certain neighborhood of a local optimum, they are hopelessly attracted to that local optimum. A special case is linear optimization, where there are algorithms that are guaranteed to find a global optimum.

In the next section, we describe a special family of optimization algorithms, known as evolutionary algorithms. They are called evolutionary algorithms because their search mechanism is inspired by concepts of the theory of evolution.

2.4 Evolutionary Computation

Evolutionary algorithms (EAs) [21], [37], [63] are part of a wider kind of optimization algorithms that use a very strong stochastic element in their search mechanism. Other examples of this family of algorithms are tabu search (TS) [31], [29], [30] and simulated annealing (SA) [41]. Another common characteristic of this family of algorithms is that they are "general purpose algorithms", this means that they can work with practically any optimization

problem, without the necessity to make any assumption about the nature of the problem. It does not matter if the optimization problem is continuous or discontinuous or the objective function is twice continuously differentiable, etc, EAs, SA and TS can be easily adapted to work with the problem.

Evolutionary Algorithms are inspired, as the name suggests, in the theory of evolution [12]. EAs use some procedures based on concepts like "population", "selection of the fittest", "mutation", "crossover", "fitness" and others.

The idea in evolutionary algorithms, is that each candidate solution $x \in \mathbf{R}^n$ is a member of a population P of size N. The first population is chosen randomly from X. There is a fitness function that assigns to each member of the population a quantity that represents how "well adapted" is the solution to the "environment". So, based on the fitness of each individual, a subset $P_{parents}$ of P, is chosen. From $P_{parents}$, a new population of N offsprings is created through a process of crossover and mutation of the $P_{parents}$. The process is repeated by using the new population until a stop condition is met. At the end, the best individual of the last population is presented as the solution of the problem. These are the general ideas of a genetic algorithm, but there are many variations.

The implicit assumption in genetic algorithms is that from the best individuals of a population, we can create new individuals that are better than the parents. There are many important concepts related to the implementation of EAs. Concepts such as the representation of an individual, crossover procedure, mutation procedure, search space, fitness function, selection function, and many others. These concepts can be implemented in several ways, and the different implementations of these concepts result in different EAs. Next, we explain briefly some of these concepts.

Search Space. Evolutionary Algorithms need to define a box in $\mathbf{R}^n$ that will be the whole search space. This box is defined as a set of intervals $[a_i, b_i]$ for $i \in \{1, \ldots, n\}$. If the model of the problem does not define such a box as

part of the constraints, it is necessary to define a box in such a way that X, or at least a promising zone of it, is included inside that box.

Fitness Function. An important step of an evolutionary algorithm is to select the fittest individuals in the population. So, a fitness function, that measures the utility of a candidate solution, is necessary. The obvious option is to use the objective function $f(x)$ of the optimization problem, but sometimes another function based on $f(x)$ is used. The fitness function may also include information about the constraints, but usually the constraints are considered in the selection procedure.

Representation of the Individuals. The most straightforward way to represent an individual in a population is by an element of $\mathbf{R}^n$, just like the candidate solutions in the model of the optimization problem. But in some cases, the individuals must be encoded differently in order to use some special operators. For some classes of EAs, the individuals are represented as strings of binary digits.

Selection Procedure. Another important operator of an Evolutionary Algorithms is the selection of the best individuals in the population which will generate the next population. This operator is inspired by natural selection, where the fittest individuals have a higher chance to survive and to reproduce, passing their characteristics to their descendants. Deterministic selection, one of the most basic selection procedures, consists on choosing the n individuals with the highest fitness from a population of N individuals, $0 < n < N$. There are many other techniques that introduce an stochastic component in the selection. For example, in a *binary tournament*, two individuals from the population are chosen randomly and the one with the highest fitness value is selected. The tournament is repeated until n individuals are chosen. If the optimization problem has constraints, these are considered in the selection process, usually giving preference to feasible solutions.

Crossover Procedure. The individuals selected in the selection procedure are recombined to create new individuals. Usually, two parents are chosen (there are several procedures to choose these two parents), and they are recombined to create one (or more) offspring(s). When the individuals are encoded with a string of K binary numbers, an example of crossover is to generate a random integer k, between zero and K, and to generate an offspring whose first k binary numbers are identical to the ones in one parent, and the last $K - k$ binary numbers are equal to the ones in the other parent. When the parents are represented with a vector of real numbers, an example of crossover is to generate an offspring equal to a random point on the segment of the line that goes from one parent to the other.

Mutation Procedure. Mutation is another important component in the search mechanism of an Evolutionary Algorithm. A mutation is a random perturbation of the elements of an individual, just as in nature random changes in the genes result in mutated individuals. When the individual is encoded as a string of binary numbers, an example of mutation mechanism is to choose randomly an element of the string and to flip its value. When the individual is represented as a vector of real numbers, an example of mutation is simply to add to the individual a vector of random noise.

Stopping Criterion. In every iterative algorithm, a stopping criterion must be established. In Evolutionary Algorithms, a cycle is composed by: the evaluation of the fitness of the current population, the selection of a set of parents from the current population, and the creation of the population for the next cycle through the crossover and mutation of the parents in the actual cycle. This cycle is commonly known as "a generation". The most common stopping criterion in EAs is to set a maximum number of generations. Another criterion is to stop when the mean value of the fitness of the population has not changed significantly in several generations.

There are three dominant paradigms of EAs: Genetic Algorithms, Evolu-

tionary Strategies and Evolutionary Programming. In Genetic Algorithms, the individuals are typically represented as a string of binary numbers and the main search mechanism is the crossover, with the mutation as a secondary mechanism. In Evolutionary Strategies, the individuals are represented as a vector of real numbers and the main search mechanism is mutation, with the crossover as a secondary operator. In Genetic Programming, the individuals are represented as a tree, this uses more complex procedures of crossover and mutation.

More recently, new classes of Evolutionary Algorithms have been proposed, such as Particle Swarm [40], [18], Ant Colonies [17], Estimation of Distribution Algorithms [46], Differential Evolution [58], etc., each one with their own search mechanism. It is common to use the term "Biological Inspired Algorithms" to refer to all of these algorithms because some of them are not inspired in evolution but in other natural processes, such as bird migration and the behavior of ant colonies.

This is only an overview of Evolutionary Algorithms. For a more complete study of EAs we recommend reviewing the wide bibliography on the subject, for example [32], [72], [59], [5], [2].

2.5 Multi–Objective Optimization

There is a branch of optimization known as multi–objective optimization that deals with problems whose model is in the form:

Optimization Problem 3.

$$Minimize\ F(x) = \langle f_1(x), f_2(x), \ldots, f_m(x) \rangle \tag{2.13}$$

$$subject\ to$$

$$g_i(x) \leq 0,\ i = 1, \ldots, I \tag{2.14}$$

$$h_j(x) = 0,\ j = 1, \ldots, J \tag{2.15}$$

$$x \in \boldsymbol{R}^n \tag{2.16}$$

From now on, when we speak of a multi–objective problem we refer to an optimization problem whose model is stated in the form of Optimization Problem 3.

As is evident, the main difference between Optimization Problem 3 with respect to Optimization Problem 1 is that we have several objective functions instead of one. Sometimes the optimization of a single objective function is called mono–objective optimization or single objective optimization.

There is more than one point of view with respect to the relationship between mono–objective and multi–objective optimization. One point of view considers Optimization Problem 3 as the more general model for optimization, and Optimization Problem 1 is a degenerated case ([14], page 1), where the number of objective functions m is equal to one.

Here, we consider another, more traditional, point of view where the model of Optimization Problem 1 is the basic model of an optimization problem, and Optimization Problem 3 is a degenerated case in which it is not possible to create a unique function of merit, but the attributes of interest of the system are identified. In other words, the model of the optimization problem can not be completed.

One may wonder if it makes sense to optimize when the user preferences have not been established clearly. The answer is that we do know a lot about what the user wants. First, the attributes of interest have been identified and they are modeled by the objective functions $f_1, \ldots, f_m$. Second, we know

what we want to do with these functions: we want to minimize them [2]. Based on these information, we can discard many candidate solutions that are not optimal and concentrate only on potential optimal solutions.

Like in mono–objective optimization, a multi–objective problem is solved when we find a single solution that gives the most benefit. Except for special cases, it is not possible to choose a single solution based on the information in Optimization Problem 3. In order to find the final solution we need the help of a Decision Maker (DM), the person who has the final word when comparing candidate solutions. In some cases, the preferences of the DM are not clear "a priori". So, we need to present the DM with some information about the optimization problem in order to clarify preferences. One of the goals of multi–objective optimization is to generate a set of promising solutions in order to "make up" the DM's mind.

In multi–objective optimization, there is another very important concept: "objective function space". We call the "objective functions space" to the space where we locate the image of the search space under $F(x)$ (Figure 2.3). Moreover, the image of X under $F(x)$ is called Z, in other words:

Definition 7. *For a multi–objective optimization problem, Z is the set of vectors in objective function space, that are generated by the feasible vectors in the search space. In mathematical notation $Z = \{z \mid z = F(x), x \in X\}$*

We mentioned above that in order to find the best solution for a multi–objective problem we need the help of a Decision Maker and that in some cases the preferences of the DM are not clear "a priori". In this cases, where the preferences are not clear, we present the DM with some elements of X (and its corresponding image in Z), so he/she can define their preferences based these attainable candidate solutions. As the final decision of the DM

[2]It is possible to have both functions to be minimized and functions to be maximized in the same multi–objective problem. Like in mono–objective optimization, we can transform any multi–objective problem with functions to be maximized, into a multi–objective problem with functions to be minimized only. For this, we only need to substitute every function $f_i(x)$ to be maximized by $-f_i(x)$.

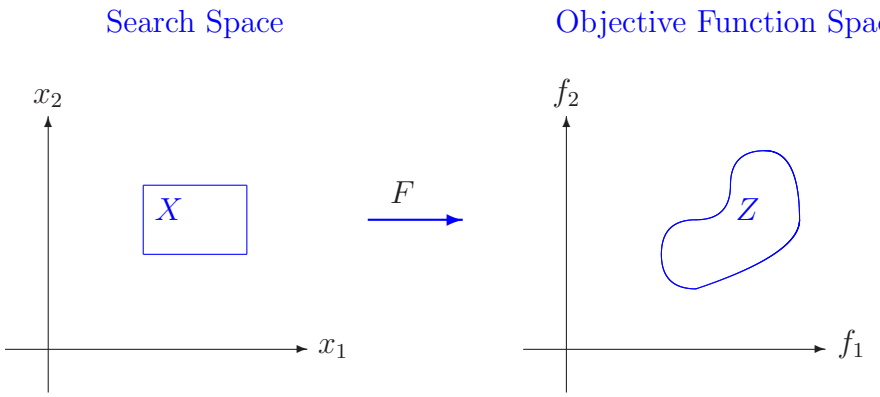

Figure 2.3: The Search Space and the Objective Function Space.

is highly influenced by the set of candidate solutions we use, it is of vital importance to use the best candidate solutions we can find. We want a set of solutions that characterize the most information of the best trade–offs between objective functions. But, how do we identify the "best" and "more informative" solutions? When we have several objective functions, what are the most promising candidate solutions?

2.5.1 Pareto Optimality Criteria

Consider the following problem: in a given school, it has been decided to choose the best science student. In order to decide who the best student is, three subjects are considered: math, chemistry and physics. The staff responsible for selecting the best student is formed by Mr. Smith, professor of math; Ms. Jones, professor of chemistry and Mr. Lee, professor of physics. The candidates with their respective scores are shown in Table 2.1.

The score on every subject is evaluated on a scale from 1 to 10, 10 is the best score and 1 is the worst score. The problem is how to decide whether one student has performed better than another.

According to some statistical studies, it is necessary to study roughly 10 hours in order to pass the final chemistry exam. Based on the same studies,

Table 2.1: Students Nominated

	Score		
Student	Math	Chemistry	Physics
Angel	9	8	9
Ivvan	9	8	6
Oyuki	10	9	7
Sergio	8	7	8
Hayday	7	8	7
Alonso	8	8	9
Noemi	5	5	5
Brenda	6	7	7
Judith	8	9	8
Jean	6	7	6

it is necessary to study 12 hours in order to pass the final physics exam and 15 hours to pass the final math exam. Based on these proportions Mr. Smith suggests multiplying the scores of the students in math, chemistry and physics by 1.5, 1 and 1.2 correspondingly, and add these weighed scores up to obtain a single number that will be used to compare the performance of the students.

Unfortunately, Ms. Jones and Mr. Lee are not happy with the idea, because they consider that this method under evaluates their subjects. They suggest assigning the same weight to each subject and to use the sum of the scores to evaluate the students. Mr. Smith insists that his subject deserves more weight, and as a consequence the staff does not arrive at an agreement.

The deadline to name a winner is near and the staff decide to name a group of finalists from which the winner will be chosen. The members of the staff are not sure about the criteria to compare the performance of two students, but it is evident that the higher the score in one subject, the better the student in that subject. And if student A has a better score than student B in all subjects, than A is better than B. Based on this, the professors can eliminate some students that, for sure, are not the best ones.

Based on Table 2.1, we can see that Noemi is not the best student, because Alonso has better scores in all subjects, so we can eliminate her. In a similar way, we can eliminate Ivvan, Hayday, Brenda and Jean, because their scores are inferior to those of Oyuki, Judith, Angel and Alonso respectively.

There is another special case where we can discard candidates. Consider the scores of Alonso and Angel, we see that they have the same score in chemistry and physics, but Angel has a better score in math. It is acceptable to consider Angel as a better student than Alonso, because Angel's scores are not worse than Alonso's in any subject, and, Angel has better score than Alonso in one subject. This criteria is not as strong as when a student does better than another one in all subjects, but it is acceptable. According to this new criteria we eliminate Alonso as a candidate for the best student. So the finalists are Oyuki, Angel and Judith.

At this point, we see that it does not matter how we combine the scores of the students in a weighted sum, as long as the weights we assign to each score are bigger than zero, the best student will be one of the three finalists. So we have reduced our search space to only three people, until more information is available to choose a winner.

This example shows one of the most popular criteria to compare two vectors in order to decide which one is better than the other. This is known as the "Pareto Optimality Criteria", and it is defined through a relation between vectors known as dominance.

Dominance

The concept of dominance is central in this work and its discussion deserves some space. Next we introduce some definitions of dominance that are frequently used in the literature. For a vector $x \in \mathbf{R}^n$, we denote by $x^{(i)}$, $i \in \{1, \ldots, n\}$ the *i–th* component of x.

Definition 8. *Dominance between two vectors. For two vectors* $a, b \in \boldsymbol{R}^n$, *we say that a dominates b if* $a^{(i)} \leq b^{(i)}$ *for* $i = 1$ *to n, and there exists at*

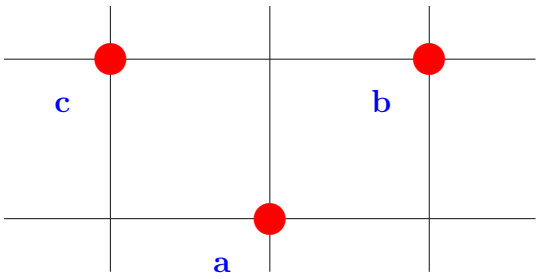

Figure 2.4: An example of dominance. Here, a dominates b, c dominates b, a and c are mutually non–dominated.

least one index $j \in \{1, \ldots, n\}$ *for which* $a^{(j)} < b^{(j)}$. *Otherwise, we say that* a *does not dominate* b.

The dominance relation defined above sometimes is called weak dominance, or Pareto dominance. There are several alternative definitions of dominance, for example, we may consider that a dominates b if $a^{(i)} < b^{(i)}$ for $i = 1$ to n. In this work we consider only Definition 8 as the definition of dominance between two vectors. A graphical example of dominance is shown in Figure 2.4. Next, we present another definition of dominance in the context of a multi–objective problem.

Definition 9. *Dominance between two candidate solutions of a multi–objective problem. We say that a candidate solution* x *dominates a candidate solution* y *if* $F(x)$ *dominates* $F(y)$.

The idea behind the concept of dominance, is that a candidate solution x is preferred over a candidate solution y, if x is not worse than y in all objective functions and there is at least one objective function for which x is strictly better than y. We present a third definition of dominance that states Definition 9 in a different way.

Definition 10. *Dominance between two candidate solutions with respect to a vector of objective functions* $F(x)$. *Denote with* $F(a > b)$, $a, b \in \boldsymbol{R}^n$, *the*

set of objective functions from $F(x)$ for which a attains a better value than b. We say that a candidate solution x dominates a candidate solution y if $F(x > y) \neq \emptyset$ and $F(y > x) = \emptyset$.

If, as in the example of the best student, we are maximizing the value of the elements of the vectors, then "better" in Definition 10 means greater than, and "not worse" means greater or equal. When we are minimizing, better means smaller than, and not worse means smaller or equal.

Note, that Definition 8 is useful when we are comparing vectors in the objective function space, while Definition 9 is useful when we are comparing two candidate solutions in the search space. It must be clear by the context what definition is being used.

Preference Structure

A preference structure refers to a set of rules that allow us to decide when one element of a set is better than another. Let A be a set, and a, b two arbitrary elements of A. One and only one of the following binary relations is possible:

List 3.

1. *a is better than b. We denote this by $a \prec b$.*
2. *a is worse than b. We denote this by $a \succ b$.*
3. *a is equivalent or equally good as b. We denote this by $a \sim b$.*
4. *We do not have enough information to decide if any of 1, 2 or 3 holds. We denote this by a ? b.*

Definition 11. *$A_{\prec}$ is the set of ordered pairs (a, b), such that for $a, b \in A, a \prec b$. $A_{\succ}$ is the set of ordered pairs (a, b), such that for $a, b \in A, a \succ b$. $A_{\sim}$ is the set of ordered pairs (a, b), such that for $a, b \in A, a \sim b$. $A_{?}$ is the*

set of ordered pairs (a, b)*, such that for* $a, b \in A, a \; ? \; b$*. The combination of* $A_{\prec}$*,* $A_{\succ}$*,* $A_{\sim}$*,* $A_{?}$ *is a preference structure.*

In mono–objective optimization, the image of the candidate solutions are real numbers, $Z \subset \mathbf{R}$. In this case the rules to compare candidate solutions in X are as follows for $x, y \in X$:

List 4.

1. *x is better than y if $f(x) < f(y)$.*
2. *x is worse than y if $f(x) > f(y)$.*
3. *x is equivalent or equally good to y if $f(x) = f(y)$.*

Note that $X_{?} = \emptyset$, all pairs of elements of X can be compared. In multi–objective optimization, we can use the dominance relation to create a preference structure between two arbitrary candidate solutions x, y, with the following relations:

List 5.

1. *x is better than y if x dominates y.*
2. *x is worse than y if y dominates x.*
3. *x is equivalent or equally good to y if $F(x) = F(y)$.*
4. *x and y are not comparable or incomparable if x does not dominate y and y does not dominate x.*

This preference structure for a multi–objective problem is defined on the elements of the search space. It can also be used when x and y are elements of the objective function space. We only need to change Item 3 in previous List 5 to "x is equivalent or equally good to y if $x = y$".

In a preference structure, it is desirable that $X_{?} = \emptyset$, because this makes it a lot easier to identify an optimal solution in an optimization problem.

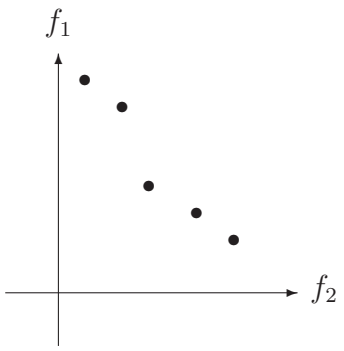

Figure 2.5: An example of a non–dominated set, represented by the black dots.

Non–dominated Set

A very important concept in multi–objective optimization is that of a non–dominated set (NS):

Definition 12. *Let $B \subset \boldsymbol{R}^n$. B is a non–dominated set if and only if for every pair of vectors $a, b \in B$, a and b are not comparable under the dominance relation.*

A non–dominated set in $\mathbf{R}^n$ is a set of mutually non–dominated vectors in $\mathbf{R}^n$ (see Figure 2.5). A non–dominated set can also be defined in the context of a multi–objective problem.

Definition 13. *Consider a multi–objective problem. A non–dominated set of candidate solutions is a set $A \subset X$, such that for every pair of candidate solutions $x, y \in A$, $F(x)$ does not dominate $F(y)$ and $F(y)$ does not dominate $F(x)$.*

So, a non–dominated set of candidate solutions is a set of candidate solutions whose image in objective functions space is a non–dominated set, according to Definition 12. It must be evident by the context if we are talking about a non–dominated set in the search space or in the objective

functions space. Finally, there is a set of non–dominated sets that is very important for multi–objective optimization.

Definition 14. *Consider a multi–objective problem. We denote by* Ω *the set of all non–dominated sets that are subsets of* Z *(*Z *given by Definition 7). We denote an arbitrary element of* Ω *with script capital letters. For example* $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, *etc. may represent different non–dominated sets from* Ω.

Note that Ω is defined in the objective functions space. Another important concept is that of the non–dominated elements of a set:

Definition 15. *The non–dominated elements of a set* $A \subset \boldsymbol{R}^n$, *denoted by* $ND(A)$, *is the set of all vectors in* A *that are not dominated by any element of* A. *In mathematical notation,* $ND(A) = \{a \in A \mid \nexists b \in A, b \text{ dominates } a\}$.

For a multi–objective problem, this definition applies both to the elements of the search space and the elements of the objective functions space. If A is in the search space, then we use the definition of dominance given in Definition 9. If A is in the objective functions space, than we use the definition of dominance given in Definition 8. There are two very important non–dominated sets related to a multi–objective problem.

Definition 16. *Consider a multi–objective problem. We call the Pareto Front (*$\mathscr{PF}$*) to* $ND(Z)$.

Definition 17. *Consider a multi–objective problem. We call the Pareto Set (*PS*) to* $\{x \in X \mid F(x) \in \mathscr{PF}\}$.

There is a function from the Pareto Set to the Pareto Front. This relation is given through the vector of objective functions of the multi–objective problem, $\mathscr{PF} = F(PS)$. A graphical representation of these sets is shown in Figure 2.6. Both PS and $\mathscr{PF}$ may have many topologies.

The Pareto Front for a multi–objective problem with m objective functions, is usually described as an $(m-1)$ dimensional surface. Veldhuizen [71]

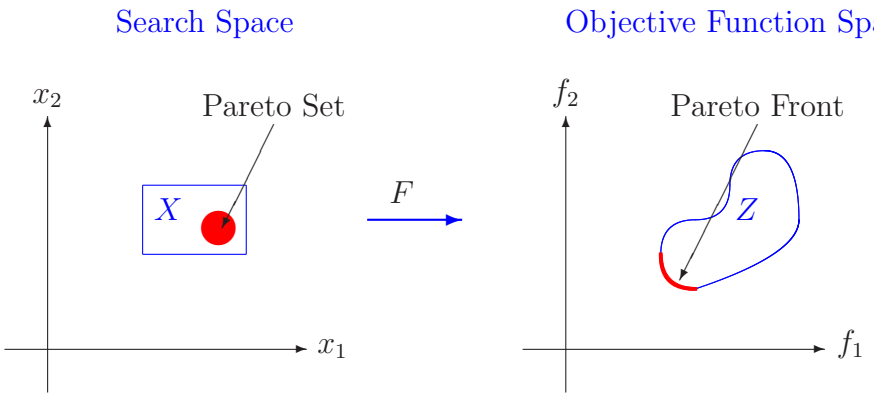

Figure 2.6: The Pareto Front and the Pareto Set.

proved that the Pareto Front is *at most* an $m-1$ surface. He also proved that the cardinal number of Z is, at most, the same cardinal number of the continuous line.

The importance of the Pareto Set (and its corresponding image, the Pareto Front) is that it contains an optimal solution for all possible user preferences. This is guaranteed through the following assumption:

General Assumption 1. *In a multi–objective problem, non–dominated solutions are preferred over dominated ones. In other words, for $x, y \in X$, if x dominates y then x is considered a better solution than y.*

At this point, it should be evident that General Assumption 1 is the most basic and general supposition when we work with multi–objective optimization using the Pareto Optimality Criteria. If the final preferences of the Decision Maker are concordant with General Assumption 1, then the final solution selected by the DM is an element of the Pareto Set. To see this, imagine that the final solution chosen (x_f) is not a member of PS. This means that there exists a vector $y \in X$ such that y dominates x_f. But if x_f is the final solution, this means that it is preferred over y, contradicting General Assumption 1.

2.5.2 The Goal of Multi–Objective Optimization

We have mentioned in previous subsections, that we need to present the Decision Maker with a set of candidate solutions to clarify preferences and to create a criterion to identify a final solution to the problem. The set of solutions that we present to the Decision Maker must be the Pareto Set, because PS contains all possible solutions to any DM preference. The Pareto Set may have a huge number of elements, so in practice we present the Decision Maker with a subset of the Pareto Set. Hence, in this work, the goal of multi–objective optimization is the following:

Goal of Multi–Objective Optimization. *The objective of multi–objective optimization is to collect as much information as possible of the multi–objective problem, so the Decision Maker can define preferences with more precision and find a unique solution for the problem. More specifically, we want to obtain an approximation of $\mathscr{PF}/PS$ that contains as much information as possible of $\mathscr{PF}$ surface, so the DM can either choose an element of this subset as the final solution, or use this information to specify preferences that allow us to search and find a final solution.*

We consider that any set of vectors that approximate the Pareto Front/Set is a non–dominated set, and from now on we refer to these sets as "an approximation set", "an approximation" or a non–dominated set. An interesting point is: what makes an approximation set more informative and thus more desirable? There are two properties that have been identified as desirable for an approximation: convergence and diversity. Convergence is related to how near an approximation is (in objective function space) to the Pareto Front. Diversity is related to how well distributed the vectors in the approximation are (in objective function space) among the Pareto Front. A graphical representation is shown in Figure 2.7. We speak more about convergence and diversity later.

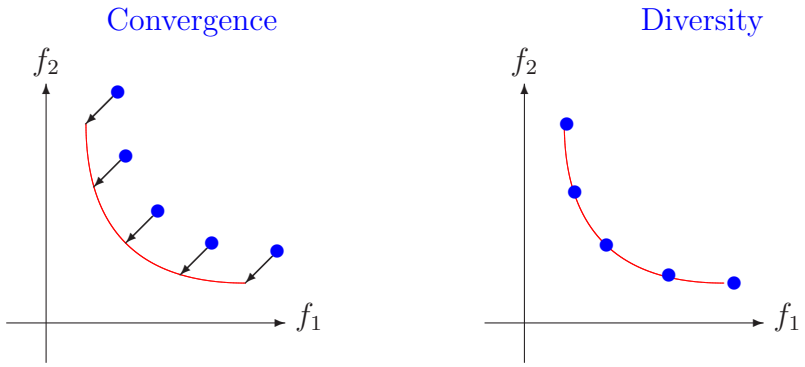

Figure 2.7: Convergence and Diversity.

2.5.3 Methods for Multi–Objective Optimization

Multi–objective optimization has been worked on in the past with several approaches. A good review of these approaches can be found in Miettinen [49].

The special case where all objectives and constraints are linear functions, in a multi–objective problem, is known as multi–objective linear programming (MOLP). In MOLP, any element of the Pareto Set can be obtained using variations of the Simplex Method (see [52]). In this work we only consider multi–objective problems that are not linear.

Depending on the role of the Decision Maker, we have different approaches to solve a multi–objective problem, for example:

List 6.

- *A posteriori methods*
- *A priori methods*
- *Interactive methods*

A Posteriori Methods

In the "a posteriori methods" the idea is to construct an auxiliary objective function $f : \mathbf{R}^m \rightarrow \mathbf{R}$ using a conversion of the form:

$$f(x) = g(f_1(x), f_2(x), \ldots, f_m(x)) \tag{2.17}$$

The function $g : \mathbf{R}^m \rightarrow \mathbf{R}$ takes as argument the vector of objective functions from a multi–objective problem and returns a real number. Then, we substitute the vector of objective functions in Formula 2.13 by Formula 2.17 to obtain an auxiliary mono–objective optimization problem. The next step is to optimize the auxiliary mono–objective problem to obtain its optimal solution. Function g in Formula 2.17, usually depends on several parameters, and by changing the parameters we obtain different objective functions $f(x)$. If g and its set of parameters are chosen carefully, the solutions of the auxiliary mono–objective problems are elements of the Pareto Front. For example, if we want to obtain ten elements of the Pareto Front/Set, we choose an auxiliary function $f(x)$, and ten randomly generated sets of parameters for g. Next, we optimize the ten different optimization problems, obtaining as a result ten different (we hope) elements of the Pareto Front/Set.

Finally, the solutions are presented to the Decision Maker in order to choose the best solution. The Decision Maker does not interact in the optimization process until the set of solutions is obtained.

An example of an auxiliary function that is commonly used in a posteriori method is the weighted sum:

$$f(x) = \sum_{i=1}^{m} w_i \cdot f_i(x), \tag{2.18}$$

where $w_i > 0$ for $i = 1$ to m and $\sum_{i=1}^{m} w_i = 1$. The parameters of the auxiliary function are the different weights w_i. Using different sets of weights we may obtain different solutions in the Pareto Front/Set. Adjusting the weights

to obtain a well distributed set of solutions along the Pareto Front is not trivial and, in some cases, some elements of the Pareto Front are impossible to obtain. The advantage of this auxiliary function is that it is easy to implement.

Another example of a posteriori method is the ϵ–Constraint Method. In this method, the vector of objective functions in Formula 2.13 is transformed into a single objective function $f_k(x)$ and $m-1$ constraints $f_i(x) \leq \epsilon_i$, that together with the original constraints form a new mono–objective problem. The index k is an element of $\{1, \ldots, m\}$ and $i = 1, \ldots, m$, $i \neq k$. The values of ϵ_i are upper bounds for the objective functions. By choosing different values for k and ϵ_i we can obtain different elements of the Pareto Front/Set, but special care must be taken in order to guarantee Pareto optimality (see [52]). The advantage of this method is that it is possible to obtain any element of the Pareto Front/Set.

A Priori Methods

In the "a priori methods" the Decision Maker specifies preferences before the optimization process starts, so a unique optimal solution can be found. This can be done in several ways, for example, the user may construct a utility function that substitutes the vector of objective functions in Optimization Problem 3, obtaining a mono–objective problem.

Another example of a priori method is the lexicographical ordering. In lexicographical ordering, the Decision Maker sorts the objective functions according to their importance. Next, we minimize the most important objective function, $f_{k1}(x)$, ignoring the others, to attain its optimal value f_{k1}^{min}. Next, we minimize the second most important function, $f_{k2}(x)$, adding the constraint $f_{k1}(x) = f_{k1}^{min}$. We continue minimizing the objective functions according to their importance and adding constraints to preserve the optimal value of the previous functions until all objective functions have been minimized.

In this work we do not consider a priori methods, because we assume that the Decision Maker is unable to provide more preferences until new information of the problem is available.

Interactive Methods

In interactive methods, the Decision Maker collaborates with the search process in an iterative way. For example, we can use a posteriori approach to create an iterative method. In each iteration, an approximation of the Pareto Front/Set is obtained. Based on this approximation the DM gets a clearer idea of preferences and provides more information to the search process, so a more specific and promissory part of the Pareto Front/Set is detected. This iteration is repeated until the DM finds a solution that is satisfactory, or until the DM gives enough information to choose a unique solution.

There are many interactive methods that use some specific information in each iteration, for example the Geoffrion–Dyer–Feinberg Method [28], the Tchebycheff Method [68], and many others.

These methods that deal with multi–objective problems are part of a research area known as Multi Criteria Decision Making (MCDM) or Multi Attribute Decision Making (MADM). There is an abundant bibliography on this subject, some examples are [52], [69], [77], [8], [45], [13], [45].

2.5.4 Multi–Objective Evolutionary Algorithms

We call multi–objective evolutionary algorithm (MOEA) an evolutionary algorithm designed to obtain an approximation of the Pareto Front/Set. Evolutionary algorithms have been used widely to solve multi–objective optimization problems. This is because genetic algorithms work with populations, or sets of solutions. As the goal of multi–objective optimization is to obtain a set of solutions that approximate the Pareto Front, genetic algorithms fit well in the solution of the problem, obtaining several Pareto optimal solutions in

a single run.

MOEAs are a posteriori methods, they approximate the Pareto Front/Set without any assumption of the Decision Maker preferences. The advantage of MOEAs over other a posteriori methods is that they can obtain many Pareto optimal vectors in a single run, while other algorithms need to make a run per solution. Another advantage of MOEAs is that we can design their search mechanism in such a way that the final set of vectors are well distributed among the Pareto Front, increasing the amount of information that the approximation provides to the Decision Maker.

In recent years, an increasing number of MOEAs have been proposed, for example VEGA [61], NPGA [38], MOGA [22], NSGA [67], NSGAII [15], SPEA [80], SPEAII [79], ϵ–MOEA [47] and many others.

Designing MOEAs is an area of research with a huge development in recent years. To explore all details of MOEAs is beyond the scope of this work. For a deeper study in multi–objective optimization with evolutionary algorithms we recommend reading [14] and [9]. The characteristics of a MOEA that are more relevant to this work is that they are algorithms that work with an optimization problem in the form of Optimization Problem 3 and return a non–dominated set as an output. Also, MOEAs are stochastic algorithms, so for different runs, they may return different non–dominated sets.

2.6 Preferences

2.6.1 Introduction

In order to solve a multi–objective optimization problem it is necessary to obtain extra information in order to be able to find a final solution. In this section we review some of the different preferences that can be helpful.

2.6.2 Decision Maker Preferences

It is important to remember that obtaining an approximation set is not the final step when solving a multi–objective optimization problem. In the end, a Decision Maker (DM) will choose a particular solution. The Decision Maker is a person, or group of people, that can make the final decision about the best solution for the problem. Here, we refer to the Decision Maker as a single person. We call *Decision Maker Preferences* the knowledge associated to the DM, that allows the identification of the optimal solution from a set of candidate solutions. The DM Preferences can be very abstract and subjective, making it necessary for the DM to "see" the candidate solutions in order to decide which one is the best. DM Preferences may be different for different DMs, and they also may change over time. As a consequence, it is possible for different elements of an approximation set to be the optimal solution of the problem for different situations.

We assume in this work, that a DM Preference is a preference structure defined on Z, where $Z_? = \emptyset$. We also assume that the preferences are transitive: if the DM considers that if $x \prec y$ and $y \prec z$ then $x \prec z$, or if $x = y$ and $y = z$ then $x = z$. Hanzen and Jazkiewicz [35] consider that the DM Preferences can be modeled by utility functions $u : \mathbf{R}^m \rightarrow \mathbf{R}$, where $\mathbf{R}^m$ is the objective functions space. This consideration is valid because a model is a representation of a system. In this case, the system is the DM Preference and it is always possible to propose a real function to model them. A different issue is how accurate the model is. For a discussion of when a DM preference can be modeled by an utility function see [52], page 678.

The goal of the DM is to maximize the utility, so the candidate solution that provides the maximum value of u is the optimal solution according to that DM preference. The maximum value for a utility function in an approximation set $\mathscr{A}$ is denoted by $u^*(\mathscr{A}) = \overset{max}{z \in A}\{u(z)\}$. We denote with $z^u_{\mathscr{A}}$ the element of $\mathscr{A}$ for which u attains the best value. For different DM preferences we may have different optimal solutions, this is illustrated in

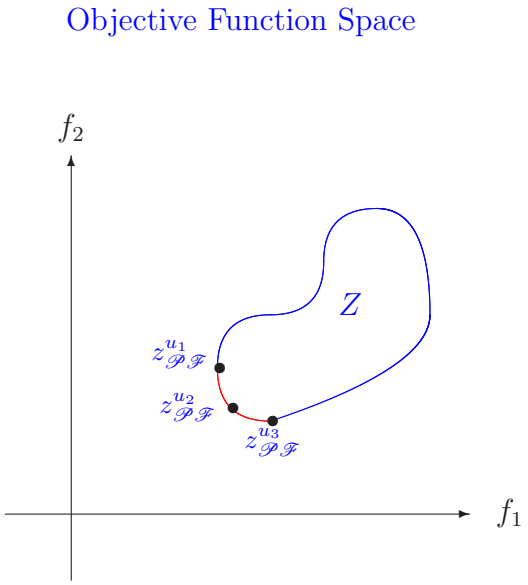

Figure 2.8: For three different utility functions, u_1, u_2 and u_3 we may have three different optimal solutions $z^{u_1}_{\mathscr{P}\mathscr{F}}$, $z^{u_2}_{\mathscr{P}\mathscr{F}}$ and $z^{u_3}_{\mathscr{P}\mathscr{F}}$.

Figure 2.8

Next we present some important definitions given in [35].

Definition 18. *A utility function u is compatible with the dominance relation if and only if $\forall z_1, z_2 \in Z$, z_1 dominates $z_2 \Rightarrow u(z_1) \geq u(z_2)$. U_C denotes the set of all utility functions that are compatible with the dominance relation. A utility function u is strictly compatible with the dominance relation if and only if $\forall z_1, z_2 \in Z$, z_1 dominates $z_2 \Rightarrow u(z_1) > u(z_2)$. U_{SC} denotes the set of all utility functions that are strictly compatible with the dominance relation.*

In this work we assume that all DM Preferences are compatible with General Assumption 1. So, all DM Preferences must be modeled by elements of U_{SC} or U_C. A utility function u can be used as an objective function to transform a multi–objective problem into a mono–objective problem, in a similar way as in the "a priori" methods.

2.6.3 Analyst Preferences

Another important entity in optimization is the analyst. We call "the analyst" the person, or group of people that are experts in optimization and are working with the DM to solve the optimization problem. The analyst is a professional in problem solving and the DM is the client. We refer to the analyst as a single person. We assume that the DM is not an expert in multi–objective optimization and that is not familiar with concepts such as Pareto Front, convergence, diversity, etc. We adopt this assumption for simplicity, but it is not true in the general case, because the DM and the analyst may be the same person, or the DM may have some knowledge about optimization.

The analyst has a well defined goal, to provide the DM with a good approximation to the Pareto Front/Set. The job of DM is to choose a final solution from the approximation. Or, if we are in an iterative approach, the DM must use the approximation to clarify preferences and use this information to update the optimization problem, so the analyst can obtain a more specific approximation and so on until a final solution is obtained.

An important question is: what is a good approximation? or what makes one approximation better than another? There are several desirable properties of an approximation that the experts have identified. These properties are supposed to make an approximation more informative for the DM. We call these properties the "Analyst Preferences". We consider two: convergence and diversity. There may be other properties but convergence and diversity are the most widely accepted.

Convergence

The most important characteristic of an approximation is convergence. As mentioned in Section 2.5.2, convergence refers to how close to the Pareto set a non–dominated set is. Unfortunately, there is not a rigorous definition of convergence, the concept of closeness to the Pareto Front has not been

defined. This incapacity to clearly define convergence is the source of many complications when trying to compare different non–dominated sets. If a non–dominated set is a subset of the Pareto Front, we can conclude that the set has complete convergence, because all its elements are Pareto optimal. But, when we are comparing two non–dominated sets, and none of them have completely converged to the Pareto Front, the question of which one is "nearer" to the Pareto Front needs to be answered.

Some quality indicators designed to measure the closeness of a non–dominates set to the Pareto Front are proposed in [62] and [71]. These quality indicators use Euclidian distances between the elements of a NS and a reference set. Unfortunately, it has been demonstrated that these quality indicators can be misleading in some cases [42][83]. Another approach is to compute the convergence of one approximation in relation to another, for example we can consider that $\mathscr{A}$ has a better convergence than $\mathscr{B}$ if all vectors in $\mathscr{B}$ are dominated by vectors in $\mathscr{A}$.

Vectors near to, or in the Pareto Front are more informative for the DM because they provide more information about the potentially optimal solutions. Solutions far away from the Pareto Front represent suboptimal options for the DM when choosing a final solution, and give misleading information to the DM when trying to define preferences. Even when there is no rigorous definition of convergence, its intuitive notion is understood and accepted by most researchers in the area.

Diversity

Just like in the case of convergence, there is no rigorous definition of diversity in the literature. As mentioned in 2.5.2, diversity refers to the distribution of the elements of an approximation set over the Pareto Front. Actually, the concept of diversity has been described in many ways, for example:

1. We want to have a good idea of the shape of the trade–off surface.

2. We want to represent the Pareto Front with the maximum resolution possible.

3. We want the vectors in the non–dominated set to be as evenly distributed as possible.

4. We want to have as many solutions as possible in the Pareto Front.

5. It is desirable to know the extreme values of the Pareto Front.

Some authors divide diversity into more specific characteristics, for example in [82] it is considered dispersion (the vectors must be evenly distributed along the Pareto Front surface), and extension (the range of values of all objective functions must be as wide as possible), instead of diversity.

This variety of definitions show how difficult it is to define the concept of diversity. Also, some of the definitions may be incorrect depending on the topology of the Pareto Front. For example, if a Pareto Front in m–dimensional space is a $(m-1)$–dimensional surface, having the elements of a non–dominated set evenly distributed is desirable, but this is not adequate if the Pareto Front is a finite set of irregularly spaced vectors.

Another example is when the Pareto Front consists of a single point or has a very short extension (compared with other zones of Z). In these cases, a non–dominated set that is located far from the Pareto Front can have a better extension, when the Pareto Front has a poor one. Using the number of vectors generated to evaluate diversity can be misleading in cases where the Pareto Front consists of a small number of vectors, or when an approximation has all its vectors clustered in a small region. Even combinations of these concepts can be problematic to interpret.

These difficulties are the reason why some authors view diversity suspiciously [44]. Actually, some methodologies to evaluate non–dominated sets ignore diversity completely [83].

Even when it is extremely complicated to define and measure diversity, we cannot ignore it. Here is a list of justifications for considering diversity:

1. The final goal of evaluating the quality of an approximation set is to create a methodology to compare two (or more) MOEAs. Most MOEAs in the literature, and especially all the state of the art MOEAs, have been designed considering diversity, using implicit or explicit mechanisms to generate approximation sets with good diversity. Ignoring diversity in our comparison methods may result in unfair comparisons between MOEAs.

2. One of the justifications of using the Pareto Optimality Criteria instead of "a priori" methods in multi–objective optimization, is that it is possible to obtain a set of solutions in a single run. If diversity is ignored, there will be no difference between one NS with one solution and another NS with many, as long as both are in the Pareto Front. So, ignoring diversity may lead us to ignore MOEAs and use the more efficient "a priori" methods.

3. The goal of presenting a set of solutions to the DM is to give him/her a good idea of the different possible trade offs between objective functions. So, the DM can have a better idea of the attainable solutions and can make a decision based on more and better information. Consider the two NSs in Figure 2.9. $\mathscr{A}$ has many elements and $\mathscr{B}$ has only one. Assume that both sets are in the Pareto Front and that they are disjoint. Appealing to the intuitive notion of convergence, it can be argued that both sets have the same convergence, but they have a huge difference in diversity. As analysts, what approximation would we choose between $\mathscr{A}$ and $\mathscr{B}$, to present to the DM?

The conclusion is that diversity is relevant and, together with convergence, is necessary to completely evaluate the quality of a non–dominated set. We are convinced that many of the problems and contradictions regarding diversity are a consequence of the lack of a clear definition. Finding an

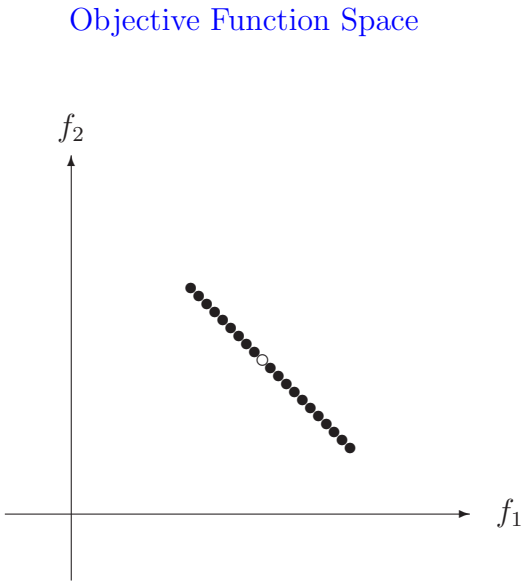

Figure 2.9: Two sets with a huge difference in diversity. $\mathscr{A}$ (•) has a better diversity than $\mathscr{B}$ (○).

appropriate definition of diversity may lead us to design a better notion of when one approximation has a better diversity than another.

2.7 Convergence and Diversity

In this work we consider both convergence and diversity to evaluate the quality of a non–dominated set. Diversity can be divided into other concepts, like number of vectors generated, distribution of the vectors in objective function space, etc, but here we consider diversity as a whole.

A very important issue is how to combine diversity and convergence to obtain a complete evaluation of a non–dominated set. One approach is to consider convergence and diversity as two incomparable objective functions and to create a secondary multi–objective problem where we are trying to optimize these two objective functions. So, in the future, the Decision Maker can decide the trade off between convergence and diversity. We do not agree

with this approach for the following reasons:

1. Considering convergence and diversity as two different goals to optimize does not solve our problem. We want to choose the best one between two sets of vectors. Measuring the quality of an approximation with two characteristics (convergence and diversity) leave us almost at the same point, we still can not decide in all cases which approximation is the best one.

2. We as analysts, are suppose to provide the DM with a good set of solutions so a good decision can be made. Choosing a good approximation is a problem that the analyst must solve, we are not supposed to pass on the problem to the Decision Maker. The DM can tells us what solutions of a set are the best, is not supposed to know about convergence and diversity.

3. If convergence and diversity are considered two non comparable goals, we can arrive at contradictions. Consider the two NSs in Figure 2.10. We see that $\mathscr{A}$ has better convergence, while $\mathscr{B}$ has better diversity. So, we have that they are not comparable, but this is a contradiction to General Assumption 1, because all vectors in $\mathscr{B}$ are dominated by vectors in $\mathscr{A}$. If we assume that these sets are not comparable, we are implying that for some DM preferences $\mathscr{B}$ has better solutions than $\mathscr{A}$, but this means that the DM may prefer dominated solutions over non–dominated ones.

We consider that both convergence and diversity must be combined to obtain the total evaluation of an approximation. Diversity must be in second place after convergence because, as mentioned above, considering diversity and convergence as two equally important objectives leads to a contradiction. On the other extreme, ignoring diversity can lead us to unfair comparisons. In our opinion, the best approach is to use convergence to make an initial

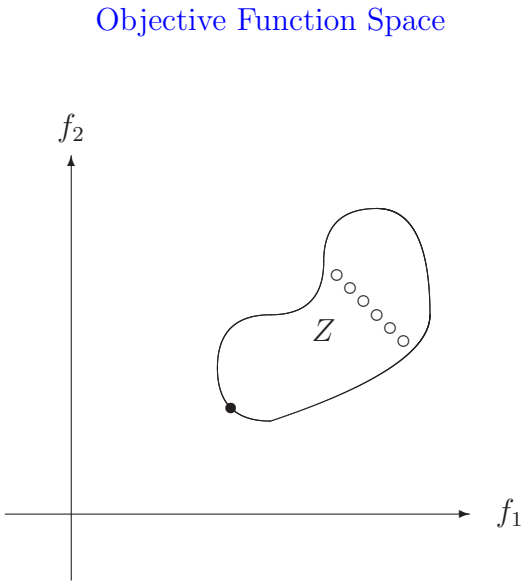

Figure 2.10: Convergence versus diversity. $\mathscr{A}$ ($\bullet$) has better convergence but $\mathscr{B}$ ($\circ$) has better diversity.

classification of the sets and then, use diversity to solve any ties when convergence is unable to distinguish between two non–dominated sets. This is the approach we use to elaborate a comparison method, and similar ideas have been proposed before [43].

Chapter 3

Quality Indicators for Non–dominated Sets

3.1 Introduction

How can we measure the quality of an approximation set? How can we decide if one approximation is better than another? These questions are really challenging to answer. Several approaches to solve the questions above have been presented in the literature but still there is a lot to be done. In this chapter we make a brief review of the work that has been done to compare the quality of two or more approximation sets.

We use $\mathscr{A} > \mathscr{B}$ to denote that $\mathscr{A}$ is better than $\mathscr{B}$, for some definition of "better". The quality of a candidate solution is usually evaluated based on its values of objective functions. So, it is common to evaluate an approximation set based on its image in the objective function space. In the rest of this work, we locate points, vectors, solutions, approximations, etc. in objective function space.

In the next subsections we review some concepts related with the quality of an approximation set and some methodologies to evaluate this quality. We want to remark that this revision is based on the "Analyst Preferences"

(convergence and diversity), and the effectiveness of the various approaches studied here is evaluated according to these preferences. Also, we assume that the final goal of evaluating the quality of an approximation, is to compare the performance of multi–objective algorithms.

3.2 Quality Indicators and their Properties

At the beginning, NSs were compared using visual validation, if $\mathscr{A}$ looks better then $\mathscr{B}$ than $\mathscr{A} > \mathscr{B}$. Unfortunately, this technique has several disadvantages. First, there is a subjective component in a visual comparison. For example, it is possible that $\mathscr{A}$ looks better than $\mathscr{B}$ for one researcher, while $\mathscr{B}$ looks better than $\mathscr{A}$ for another researcher. Second, for more than three dimensions, it is very hard to visualize a non–dominated set. As a consequence, visual validation is not trustable in all cases, and is limited to two or three dimensions.

On the other hand, visual validation can be very useful under some conditions. When there are considerable differences in convergence and diversity (for 2d and 3d problems), visual validation can give a good idea of the quality of the sets. Actually, visual validation has been useful to detect errors in some numerical comparison methods. While these methods state that $\mathscr{A} > \mathscr{B}$, when the approximations are plotted, it is evident that $\mathscr{B} > \mathscr{A}$. So, under some conditions and used with precaution, visual validation can be useful to support the development of new comparison methods.

In order to make a numeric evaluation of the outputs of MOEAs, quality indicators (QIs) are introduced:

Definition 19. *An m–ary quality indicator is a function $I : \Omega^m \to \boldsymbol{R}$ which assigns to each m–tuple $(\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_m)$ a real value $I(\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_m)$.*

Ω is given by Definition 14. A unary quality indicator $I(\mathscr{A})$ takes one approximation as an argument. Unary quality indicators can be considered

models of quality for an approximation. Different quality indicators represent different models of quality, with different levels of accuracy. Higher (or lower, depending on how it was defined) values of $I(\mathscr{A})$ mean that $\mathscr{A}$ is "better". So, $\mathscr{A} > \mathscr{B}$ if $I(\mathscr{A}) > I(\mathscr{B})$, based on a quality indicator I.

Quality indicators can be classified depending on the aspect of quality they are evaluating. There exist quality indicators designed to measure convergence, others are designed to measure diversity and there are others designed to measure the general quality of a non–dominated set.

Some unary quality indicators use extra information about the multi–objective problem in order to make their evaluations. For example, Generational Distance [71] uses a set of vectors as a reference to evaluate the convergence of a non–dominated set. The reference set may be a subset of the Pareto Front, an approximation of the Pareto Front, or a synthetic set of vectors defined by the analyst. Some QIs have parameters to adjust. For example the S–metric [81] needs a reference point, whose position is determined by the analyst. The problem is that, in general, the evaluation of these QIs depends on the set of parameters and reference information used. So, it is possible for a quality indicator I and a pair of approximations $\mathscr{B}$ and $\mathscr{A}$, that under a set of parameters $I(\mathscr{A}) > I(\mathscr{B})$ and under another set of parameters $I(\mathscr{B}) > I(\mathscr{A})$.

Moreover, it is important to remember that a unary quality indicator is only a model, so its evaluations may not always agree with reality. Some unary QIs may contradict our knowledge of when one non–dominated set is better than another. The problem is that the quality of a non–dominated set is very difficult to model and some quality indicators usually introduce biases in their evaluations, or do not capture well the essence of the properties they are trying to evaluate.

A binary quality indicator $I(\mathscr{A}, \mathscr{B})$ takes two approximations as an argument and returns a real number. This number represents the quality of one of the two approximations with respect to the other approximation. Usually,

binary quality indicators need more interpretation than unary quality indictors. For example, for $\mathscr{A}$ and $\mathscr{B}$, we have two real numbers $I(\mathscr{A}, \mathscr{B})$ and $I(\mathscr{B}, \mathscr{A})$. Depending in how the quality indicator is defined, and we need to interpret these numbers to decide whether $\mathscr{A}$ or $\mathscr{B}$ is better. An example of a binary quality indicator $I(\mathscr{A}, \mathscr{B})$, is the proportion of vectors in $\mathscr{B}$ that are dominated by vectors in $\mathscr{A}$.

Unary quality indicators have some advantages over binary ones. Unary QIs allow us to order the elements of Ω. This makes the analysis of performance a considerably easier, especially if we are comparing $m > 2$ approximations. When using a binary QI, the number of evaluations that we need to make is $O(m^2)$, this makes the analysis of the results even more complicated. Moreover, some binary quality indicators can induce cycles. For example, for approximations $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$, some binary QIs can consider $\mathscr{A} > \mathscr{B}$, $\mathscr{B} > \mathscr{C}$ and $\mathscr{C} > \mathscr{A}$. For these reasons, unary QIs have been more popular than binary QIs.

Zitzler et al. published a study [83] where it is claimed that binary QIs are superior to unary QIs. According to that study, all quality indicators, or combination of quality indicators, must have a property known as "compatibility and completeness", because only then a QI can detect whether one approximation is better than another. Then, it is demonstrated that unary quality indicators cannot be, in general, compatible and complete. Due to this, unary QIs are considered "politically incorrect" and their use have been discouraged. It is notorious that, in practice, unary QIs are used more often than binary ones. Another interesting point is that, when it was demonstrated that unary QI can not be compatible and complete, some assumptions that are not met in practice were assumed. When we consider practical conditions we obtain a different result (see Appendix A).

Originally, quality indicators were called "metrics", because their intention is to measure the quality of a non–dominated set. Unfortunately, the term "metric" has a specific meaning in mathematics. This meaning is not

necessarily compatible with the meaning we are using here. The term metric is considered incorrect, so the term "performance measure" has become very popular to refer to a quality indicator. But, performance measure is also considered incorrect, because "performance" is usually related with efficiency with respect to time. In the rest of the work we refer to these functions as quality indicators or performance measures.

There are some properties that are desirable to have for a quality indicator, because quality indicators with these properties is more robust in misleading cases. Two examples of these properties are *monotony* and *relativity* [44].

Definition 20. *A quality indicator I has the property of monotony if adding a new non–dominated vector to an approximation $\mathscr{A}$, improves its evaluation of $\mathscr{A}$. A quality indicator I has the property of weak monotony if adding more non–dominated vectors to an approximation $\mathscr{A}$, does not degrade its evaluation of $\mathscr{A}$. By "adding a new non–dominated vector to $\mathscr{A}$", we mean that we add to $\mathscr{A}$ a vector $c \notin \mathscr{A}$, that neither is dominated by current elements of $\mathscr{A}$, nor dominates any current element of $\mathscr{A}$.*

Definition 21. *A quality indicator I has the property of relativity if it evaluates the Pareto Front as better than any other non–dominated set in Z. A quality indicator I has the property of weak relativity if it evaluates the Pareto Front as not worse than any other non–dominated set in Z.*

The importance of these properties must be evident. The Pareto Front is better than any of its approximations, so it must be evaluated as better (or at least not worse) than any other approximation. Adding more elements to a non–dominated set increases the information contained in the set, so its evaluation must increase (or at least not decrease).

3.3 Hansen and Jaszkiewicz Approach

One of the most influential studies of performance measures is a technical report written by Hansen and Jaszkiewicz [35]. In that report, the authors create a framework to analyze quality indicators. Also, they propose some quality indicators and give some directions about how to create QIs. Hansen and Jaszkiewicz rely strongly on the concept of utility function to find a partial answer to the question of when one non dominated set is better than another. First, they present from the following definition:

Definition 22. *$\mathscr{A}$ is better than $\mathscr{B}$ if for some user preferences, $\mathscr{A}$ contains a better solution than $\mathscr{B}$, but there is no user preference for which $\mathscr{B}$ contain a better solution than $\mathscr{A}$.*

Also, they assume that all user preferences can be modeled by utility functions u, and denoted by U a set of utility functions. Let $(\mathscr{A} > \mathscr{B})_U \subset U$ denote the set of utility functions in U that attain a better value in $\mathscr{A}$ than in $\mathscr{B}$. We have the following definition:

Definition 23. *Out–performance relation subject to a set of utility functions U. $\mathscr{A}$ out–performs $\mathscr{B}$ subject to a set of utility functions U, denoted by $\mathscr{A}\ O_U\ \mathscr{B}$, if $(\mathscr{A} > \mathscr{B})_U \neq \emptyset$ and $(\mathscr{B} > \mathscr{A})_U = \emptyset$. In other words, $\mathscr{A}\ O_U\ \mathscr{B}$ if some utility functions $u \in U$ achieves better values for $\mathscr{A}$ than for $\mathscr{B}$ but no function $u \in U$ achieves a better value for $\mathscr{B}$ than for $\mathscr{A}$.*

An out–performance relation, is a binary relation between two non–dominated sets, so if $\mathscr{A}$ out–performs $\mathscr{B}$ (denoted by $\mathscr{A}\ O\ \mathscr{B}$) then we can conclude that $\mathscr{A}$ is better than $\mathscr{B}$. Note that the out–performance relation given in Definition 23 is an extension of the dominance relation between two candidate solutions, given in Definition 10. In Definition 10 we have candidate solutions to be evaluated in a vector of objective functions, in Definition 23 we have sets of vectors to be evaluated by a set of utility functions.

Remember that in this work we only consider utility functions compatible with the dominance relation (we recommend to read Definition 18 for the discusion in the following paragraphs). In order to apply Definition 23 it is necessary to define a set of utility functions. Hansen and Jaszkiewicz also defined other out–performance relations that do not depend on a set of utility functions.

Definition 24. Weak outperformance*: $\mathscr{A}$ weakly outperforms $\mathscr{B}$, denoted by $\mathscr{A}\ O_W\ \mathscr{B}$, when for every point $b \in \mathscr{B}$ there exists a point $a \in \mathscr{A}$ such that a dominates b or $a = b$ and there exists at least a point $c \in \mathscr{A}$ such that $c \notin \mathscr{B}$.*

Definition 25. Strong outperformance*: $\mathscr{A}$ strongly outperforms $\mathscr{B}$, denoted by $\mathscr{A}\ O_S\ \mathscr{B}$, when for every point $b \in \mathscr{B}$ there exists a point $a \in \mathscr{A}$ such that a dominates b or $a = b$ and there exists at least a pair of points $r \in \mathscr{A}$ and $s \in \mathscr{B}$ such that r dominates s.*

Definition 26. Complete outperformance*: $\mathscr{A}$ completely outperforms $\mathscr{B}$, denoted by $\mathscr{A}\ O_C\ \mathscr{B}$, when for every point $b \in \mathscr{B}$ there exists at least one point $a \in \mathscr{A}$ such that a dominates b.*

The definitions of O_W, O_S and O_C are concordant with Definition 23 if we use U_C as the reference set of utility functions. For example, if $\mathscr{A}\ O_C\ \mathscr{B}$ then the set $(\mathscr{B} > \mathscr{A})_{U_C}$ is empty, because every vector in $\mathscr{B}$ is dominated by a vector in $\mathscr{A}$, and all $u \in U_C$ give a better or no worse evaluation to dominating vectors. At the same time, $(\mathscr{A} > \mathscr{B})_{U_C}$ is not empty, because some $u \in U_C$ give a better evaluation to dominating vectors. So, if $\mathscr{A}\ O_C\ \mathscr{B}$ there are some utility functions for which $\mathscr{A}$ is better than $\mathscr{B}$, but there is no utility function for which $\mathscr{B}$ is better then $\mathscr{A}$. Similarly, if $\mathscr{A}\ O_W\ \mathscr{B}$, or $\mathscr{A}\ O_S\ \mathscr{B}$, it is easy to find that $(\mathscr{B} > \mathscr{A})_{U_C}$ is empty and that $(\mathscr{A} > \mathscr{B})_{U_C}$ is not empty.

For the cases where $\neg(\mathscr{A}\ O\ \mathscr{B})$ and $\neg(\mathscr{B}\ O\ \mathscr{A})$ (O may be O_W, O_S, O_C and $\neg$ denotes the logic negation), we say that $\mathscr{A}$ and $\mathscr{B}$ are not comparable

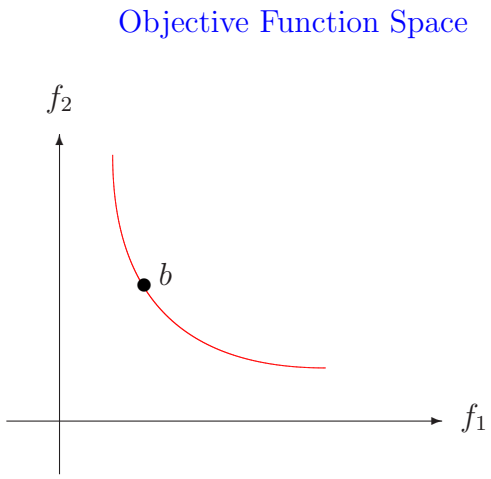

Figure 3.1: For $\mathscr{A}$ ($\cdot$) and $\mathscr{B}$ ($\bullet$), suppose that $b \in \mathscr{PF}$ and $b \notin \mathscr{A}$. $\mathscr{A}$ and $\mathscr{B}$ are non–comparable under the out–performance relations.

under the out–performance relation O.

The reason why the out–performance relations represent a partial answer to the question of when one non–dominated set is better than another, is because when $\mathscr{A}$ and $\mathscr{B}$ are not comparable, no conclusion can be derived. This is very important because it is possible for $\mathscr{A}$ to be better than $\mathscr{B}$ even if $\mathscr{A}$ does not out–perform $\mathscr{B}$.

In order to understand how the out–performance relations work, it is useful to study the negations of the out–performance relations. For example, the negation of weak out–performance is as follows:

Definition 27. Negation of weak outperformance. *$\mathscr{A}$ does not weakly outperform B when $\mathscr{A} = \mathscr{B}$, or when there exists at least one point $b \in \mathscr{B}$ that is neither contained in $\mathscr{A}$ nor dominated by any point in $\mathscr{A}$.*

We want to remark that a single vector $b \in \mathscr{B}$, that is neither contained in $\mathscr{A}$ nor dominated by elements of $\mathscr{A}$ is enough to conclude that $\mathscr{B}$ is not weakly out–performed by $\mathscr{A}$. Actually, the same condition is enough to consider that $\mathscr{B}$ is not strongly or completely out–performed by $\mathscr{A}$. This leads us

to realize how limited is the inference power of the out–performance relations (O_C, O_S and O_W) when the approximations have a similar convergence.

An extreme example can be seen in Figure 3.1. The red line represents $\mathscr{A}$ and the black dot represents $\mathscr{B}$. Suppose that $b \notin \mathscr{A}$ and that b is in the Pareto Front for a multi–objective problem. Evidently, $\mathscr{A}$ is better than $\mathscr{B}$, but according to the out–performance relations described above, $\mathscr{A}$ and $\mathscr{B}$ are not comparable. $\mathscr{B}$ does not outperform $\mathscr{A}$ because there exist many elements in $\mathscr{A}$ that neither are in $\mathscr{B}$ nor are dominated by any element of $\mathscr{B}$. $\mathscr{A}$ does not outperform $\mathscr{B}$ because there exists an element in $\mathscr{B}$, b, that neither is in $\mathscr{A}$ nor is dominated by any element of $\mathscr{A}$.

The conclusion is that for many pairs of approximations, there is a considerable difference in quality, but the out–performance relations are not able to detect it. Hansen and Jaszkiewicz recommended to use stronger assumptions about DM's preferences for the cases where two non–dominated sets were non comparable. As a result, a family of QIs known as the *R–metrics* was created. We describe the R–metrics later.

The out–performance relations are useful to establish a minimum of what characteristics are expected from a quality indicator. Let I be a quality indicator, and let $I(\mathscr{A} > \mathscr{B})$ means "I considers $\mathscr{A}$ better than $\mathscr{B}$" and $I(\mathscr{A} \geq \mathscr{B})$ means "I considers $\mathscr{A}$ better than/equal to $\mathscr{B}$". A desirable property of a QI is the following:

Definition 28. *A quality indicator I is compatible with an out–performance relation O (where O is any of O_U, O_W, O_C or O_S) when I always evaluates $\mathscr{A}$ as better than $\mathscr{B}$ if $\mathscr{A}$ O $\mathscr{B}$. In mathematical notation, I is compatible with O when $\mathscr{A}$ O $\mathscr{B} \Rightarrow I(\mathscr{A} > \mathscr{B})$. I is weakly compatible with an out–performance relation O when $\mathscr{A}$ O $\mathscr{B} \Rightarrow I(\mathscr{A} \geq \mathscr{B})$.*

An indicator compatible with the out–performance relations is more robust to misleading cases, and it is desirable to design and use quality indicators that are compatible with these relations.

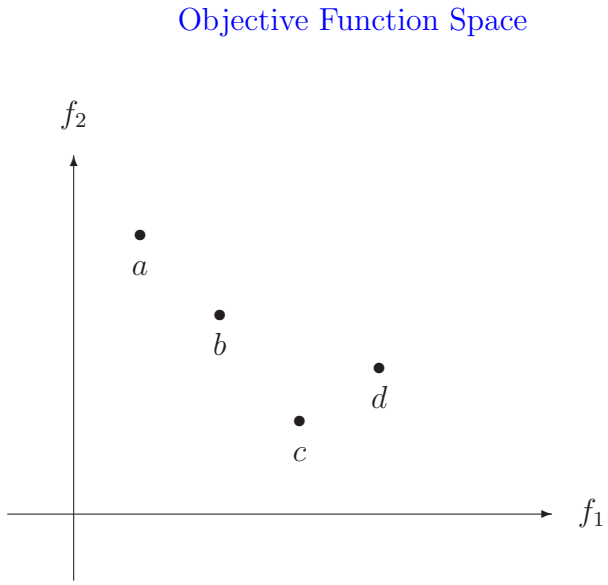

Figure 3.2: Some elements of Z.

It is interesting to note that strong out–performance can be contradictory to the concept of monotony in some cases. To see this, consider the vectors in Figure 3.2. Let $\mathscr{A} = \{a, b, c\}$ and $\mathscr{B} = \{a, b\}$. It is evident that $\not(\mathscr{A}\ O_W\ \mathscr{B}$. But if we add to $\mathscr{B}$ the vector d, we have that $\mathscr{A}\ O_S\ \mathscr{B}$. The evaluation of $\mathscr{B}$ is degraded after adding a new vector to $\mathscr{B}$.

3.4 Zitzler et al. Framework

Another influential study on performance measures was realized by Zitzler et al. [83]. In that study, the authors strongly relied on the work of Hansen and Jaszkiewicz and went forward with the objective of analyzing how useful current comparison methods are. They considered the scenario where more than one quality indicator may be used to evaluate or compare the quality of non–dominated sets. Under this scenario, it may be necessary to introduce an interpretation function E. So, when comparing two approximations, we use E to interpret the results of all the quality indicators. The result is a

formal definition of a comparison method:

Definition 29. *Let $\boldsymbol{I} = (I_1, I_2, ..., I_k)$ be a k–tuple of quality indicators. Let $\boldsymbol{I}(\mathscr{A})$ be a vector of real values $\langle I_1(\mathscr{A}), I_2(\mathscr{A}), ..., I_k(\mathscr{A})\rangle$ generated by a list $\boldsymbol{I}$ of unary quality indicators. Let $\boldsymbol{I}(\mathscr{A}, \mathscr{B})$ be a vector of real values $\langle I_1(\mathscr{A}, \mathscr{B}), I_2(\mathscr{A}, \mathscr{B}), ..., I_k(\mathscr{A}, \mathscr{B})\rangle$ generated by a list $\boldsymbol{I}$ of binary quality indicators. Let $E : \boldsymbol{R}^k \times \boldsymbol{R}^k \rightarrow \{false, true\}$ a Boolean function which takes two real vectors of length k as arguments. When $\boldsymbol{I}$ consist of unary quality indicators only, a comparison method $C_{\boldsymbol{I},E}$ is a function in the form:*

$$C_{\boldsymbol{I},E}(\mathscr{A}, \mathscr{B}) = E(\boldsymbol{I}(\mathscr{A}), \boldsymbol{I}(\mathscr{B})) \tag{3.1}$$

When $\boldsymbol{I}$ consists of binary quality indicators only, a comparison method $C_{\boldsymbol{I},E}$ is a function in the form:

$$C_{\boldsymbol{I},E}(\mathscr{A}, \mathscr{B}) = E(\boldsymbol{I}(\mathscr{A}, \mathscr{B}), \boldsymbol{I}(\mathscr{B}, \mathscr{A})). \tag{3.2}$$

If $C_{\boldsymbol{I},E}(\mathscr{A}, \mathscr{B})$ is true, then $\mathscr{A}$ is better than $\mathscr{B}$, according to $C_{\boldsymbol{I},E}$. And if $C_{\boldsymbol{I},E}(\mathscr{A}, \mathscr{B})$ is false, then $\mathscr{A}$ is not better than $\mathscr{B}$ according to $C_{\boldsymbol{I},E}$. Note also that it is possible to create a comparison method using a combination of unary and binary quality indicators in $\boldsymbol{I}$.

These definitions established a more formal way to define a methodology to compare two non–dominated sets. In [83] is defined a property known as "compatibility and completeness"[1], similar to the property of compatibility with the out–performance measures described in Definitions 24, 25 and 26.

Definition 30. *A comparison method $C_{\boldsymbol{I},E}$ is compatible and complete with respect to an out–performance relation O when $C_{\boldsymbol{I},E}(\mathscr{A}, \mathscr{B}) = true$ if and only if $\mathscr{A}$ O $\mathscr{B}$.*

[1]Zitzler et al. [83] define two properties for quality indicators. One property is called compatibility and the other one is called completeness. In this work we refer to the property of being both compatible and complete as "compatibility and completeness".

So, a comparison method is compatible and complete with respect an out–performance relation O[2] if and only if $\forall \mathscr{A}, \mathscr{B}(\mathscr{A}\ O\ \mathscr{B} \Leftrightarrow C_{\mathbf{I},E}(\mathscr{A}, \mathscr{B}) = true)$.

In [83] it is postulated that a comparison method must be compatible and complete. According to the study of Hansen and Jaszkiewicz [35] it is clear that a comparison method must evaluate $\mathscr{A}$ as better than $\mathscr{B}$ *if* $\mathscr{A}\ O\ \mathscr{B}$, but what is the justification to consider that a comparison method must evaluate $\mathscr{A}$ as better than $\mathscr{B}$ *only if* $\mathscr{A}\ O\ \mathscr{B}$? The justification also comes from [35], where it is clearly stated that when two non–dominated sets are not comparable under the out–performance relations, it is not possible to decide what NS is better than the other under General Assumption 1. In that case more assumptions about DM's preferences must be used in order to find differences between the sets.

From the framework just described, several important conclusions are derived. Maybe the most important conclusion is the affirmation that unary comparison methods have a limited inference power. This conclusion is derived from the fact that a comparison method based on a finite number of unary quality indicators cannot be compatible and complete (see the demonstration in [83]). Many of the quality indicators available in the bibliography were revised and it was found that most quality indicators are not compatible and complete.

When we study the definition of a compatible and complete comparison method, we arrive at some interesting conclusions. One of these conclusions is that we must not speak of *a* compatible and complete comparison method (CCCM) but of *the* compatible and complete comparison method. In other words, all compatible and complete comparison methods are exactly the same function.

Two functions are considered equal if they have the same graph. For

[2]In [83], the property of compatibility and completeness is defined for any binary relation between approximations. We focus only on the out–performance relations.

a function $f : A \rightarrow B$, its graph is the set of ordered pairs (a, b), where $a \in A, b \in B$ and $b = f(a)$. For example, consider a function f_1 whose domain is $A = \{1, 2, 3, 4\}$ and its codomain is $B = \{1, 4, 9, 16\}$. In order to relate the elements of the domain and codomain, f_1 uses the following rule:

$$f_1(a) = a^2 \tag{3.3}$$

The graph of f_1 is the set $\{(1, 1), (2, 4), (3, 9), (4, 16)\}$. Now, consider another function f_2 with the same domain and codomain of f_1, but the rule that f_2 uses to relate the elements of the domain with the elements of the codomain is:

$$f_2(a) = \begin{cases} 3(a-1)+1 & \text{if } 1 \leq a \leq 2 \\ 7(a-3)+9 & \text{if } 3 \leq a \leq 4 \end{cases}$$

The graph of f_2 is $\{(1, 1), (2, 4), (3, 9), (4, 16)\}$. As we can see, f_1 and f_2 are the same function, because both have that same graph. The "rule" used to associate the domain with the codomain is irrelevant, as long as we obtain the same graph we have the same function.

A compatible and complete comparison method is a function in the form: $f : \Omega \times \Omega \rightarrow \{true, false\}$, so its graph is the set $\{(w, y) | w \in \Omega \times \Omega, y \in \{true, false\}\}$ of ordered pairs, where $y =$ "$true$" if and only if the first set of w out–performs the second set in w. The consequence of all this is that all compatible and complete comparison methods are exactly the same function, because they have the same graph. As mentioned above, the "rule" we use to make the map from $\Omega \times \Omega$ to $\{true, false\}$ is irrelevant. Thus, it does not matter what vector $\mathbf{I}$ of quality indicators we use, at the end all of them give the same information. In other words, once we know one compatible and complete comparison method, we know them all.

Further, two examples of compatible and complete comparison methods where given in [83] based in two different quality indicators. The question is, if we already have some of these methods and all of them give the same

information, is it necessary to look for new ones? A possible answer to that question is to improve the computational complexity of the comparison method. The computation complexity of all known CCCM is $O(|\mathscr{A}||\mathscr{B}|)$ but, is it possible to find a faster CCCM?

Unfortunately, compatible and complete comparison methods have the same limitations in inference power of the out–performance relations. Recalling the example of Figure 3.1, where $\mathscr{A}$ and $\mathscr{B}$ do not out–perform each other, a compatible and complete comparison method, by definition, considers these sets as non–comparable. This is a disadvantage; usually researchers want a comparison method to be able to identify such considerable differences in the quality of the approximations. But, by definition, a CCCM cannot detect any difference when the approximations are not comparable under the out–performance relations. The conclusion is that, contrary to what is widely believed, being compatible and complete is not desirable for a comparison method, or for a quality indicator. An important consequence of this, is that unary quality indicators are not inferior to binary ones. Unary QIs are considered inferior because they cannot be compatible and complete. But if compatibility and completeness is an undesirable property, then there is no known reason to consider unary QIs as inferior.

3.5 Farhang-Mehr and Azarm's Framework

Another interesting framework for the analysis and design of performance measures is that of Farhang-Mehr and Azarm [48]. A peculiar aspect of this study is that the authors do not focus on answering the question of when one non–dominated set is better than another. Instead, they investigated how to select different quality indicators in the most efficient way such that there is not redundant information, and all aspects of quality of a non–dominated set are evaluated.

Their work introduces the concept of excellence relation, whose definition

we reproduce next:

Definition 31. *An excellence relation, denoted by R, is defined as a strict partial order in* Ω *that relates all non-dominated sets that are objectively comparable with respect to a common aspect of quality.*

Where Ω denotes the set of all non–dominated sets. A strict partial order between the elements of a set A, is an irreflexive, antisymmetric, and transitive binary relation that compares some and not every pair of elements of A.

An excellence relation is associated with some aspects of quality of a non–dominated set. For example, Farhang-Mehr and Azarm consider the out–performance relations as excellence relations related to convergence. They also define a new excellence relation related to diversity.

Only binary and symmetric quality indicators are considered in that study and some interesting conclusions were derived. For example, suppose that we consider n different aspects of quality when evaluating some approximations. In order to obtain a complete comparison, no less than n different quality indicators must be used. If we do not want to obtain redundant information, then no more than n quality indicators must be used. As a conclusion, the most efficient and complete way to evaluate approximation sets considering n aspects of quality is to use n quality indicators, one for each aspect of quality.

The framework presented in [48] is an interesting approach to solve the problem of how to compare approximation sets. But, in this work we follow a different road for several reasons, for example:

1. In [48] it is assumed that the DM will provide a set of quality aspects to evaluate an approximation set. In our approach, we assume that the DM is not familiar with concepts such as convergence and diversity (this is also mentioned in [48]). On the contrary, we assume that the

DM is able to compare two candidate solutions but not two approximations. The decision of when one approximation is better than another comes from the analyst. We have two different tasks: one task is to optimize a multi–objective problem, and the other task is to compare the performance of two MOEAs. When solving a multi–objective problem, we assume that either the DM or the analyst already has chosen a MOEA to work with, and that we can introduce DM preferences. When comparing the performance of different MOEAs, the quality of an approximation must not depend on the preferences of the DM, because for each DM we must make an analysis of what MOEA performs better. This is inefficient and makes it almost impossible to decide when one MOEA is better than another.

2. In [48] it is recommended to use several quality indicators to evaluate approximations but it says nothing about how to combine their evaluations. Using n QIs give us n different numbers for each approximation, but how do we use these numbers to decide when one approximation is better than another? Without a criterion to combine those numbers many approximations may not be comparable, and we return to the point where we started. One of the goals in this work is to be able to rank m approximations from the best to the worst.

3.6 Quality indicators, Comparison Methods and Others.

3.6.1 Introduction

In this subsection we describe a few of the quality indicators that can be found in the literature. It is not our intention to be exhaustive in this review; we only want to give an idea of the different approaches that have been proposed.

For more examples of quality indicators we recommend reading [62], [71], [81], [54], [76], [65], [44], [19].

3.6.2 Reference Points Found

This quality indicator is mentioned in [35] [3] and consists of the number of elements of the Pareto Front in an approximation, divided by the number of elements in the Pareto Front. In mathematical notation:

$$C1(\mathscr{A}) = \frac{|\mathscr{A} \cap \mathscr{PF}|}{|\mathscr{PF}|}. \tag{3.4}$$

This quality indicator gives a value of zero to NSs that have no elements in $\mathscr{PF}$ and give and increasing value to NSs with more elements in $\mathscr{PF}$. It is evident that in cases where the Pareto Front has an infinite number of elements or where the Pareto Front is unknown, this quality indicator cannot be used. For those cases, we can use a different version, where $\mathscr{PF}$ is substituted by a reference set $\mathscr{R}$. $\mathscr{R}$ could be a subset of $\mathscr{PF}$ or the best non–dominated set known. Then, we count the number of vectors in $\mathscr{A}$ that are not dominated by any element in $\mathscr{R}$. The mathematical description of this modified version is presented next:

$$C2(\mathscr{A}) = \frac{|\{a \in \mathscr{A} |\ \nexists r \in \mathscr{R}, r \text{ dominates } a\}|}{|\mathscr{A}|}. \tag{3.5}$$

Hansen and Jaszkiewicz [35] mention some disadvantages of these quality indicators:

- They are insensitive to improvements of the non–dominated sets. For example, if $\mathscr{R}\ O_C\ \mathscr{A}$ and $\mathscr{A}\ O_C\ \mathscr{B}$, C1 and C2 consider $\mathscr{A}$ and $\mathscr{B}$ as equally good, but $\mathscr{A}$ is better than $\mathscr{B}$.

[3] Hansen and Jaszkiewicz [35] mention that this quality indicator is used in [70] in 1993. We could not find a copy of [70], so all our information about the "number of reference points found" is based on [35].

- C1 and C2 are not sensitive to some differences in the diversity of the NSs. For example, suppose that $\mathscr{A}$ has k elements in $\mathscr{PF}$, but clustered in a small region and $\mathscr{B}$ has k elements in $\mathscr{PF}$ well distributed along all the extension of $\mathscr{PF}$. C1 and C2 give the same value to both NSs, even when $\mathscr{B}$ has richer information about $\mathscr{PF}$.

- If none of the NSs to compare converged to $\mathscr{PF}$, C1 and C2 will not be able to give any more information about the non–dominated sets.

A binary version of this quality indicator is described in [42]:

$$C3(\mathscr{A},\mathscr{B}) = \frac{|\{b \in \mathscr{B} | \exists a \in \mathscr{A}, a \text{ dominates } b\}|}{|\mathscr{B}|}. \tag{3.6}$$

3.6.3 Non–dominated Set Spacing

Proposed in [62], this metric measures the spacing between the elements of a non–dominated set. The formula is:

$$Spacing(\mathscr{A}) = \frac{1}{|\mathscr{A}|-1}\sum_{i=1}^{|\mathscr{A}|}(d-d_i)^2, \tag{3.7}$$

where d is the mean value of all d_is and $d_i = \overset{min}{j}\{|a_i^{(1)} - a_j^{(1)}| + |a_i^{(2)} - a_j^{(2)}|\}$, $a_i, a_j \in \mathscr{A}$, $j \neq i$, $a_l^{(k)}$ stands for the k–th component of the l–th vector in $\mathscr{A}$. This quality indicator is defined for two objective functions, but it can be extended for more objective functions.

3.6.4 S–metric

Proposed in [81], the S–metric consists of the space enclosed by a non–dominated set $\mathscr{A}$ and a fixed reference point r^*. For a minimization problem, the bigger the space enclosed the better the evaluation of $\mathscr{A}$. This is one of the most popular performance measures in the literature.

The evaluation of the S–metric for a non–dominated set $\mathscr{A}$, can be expressed with the formula [51]:

$$S(\mathscr{A}) = \mu\left(\bigcup_{a\in\mathscr{A}}\{x \mid (a \text{ dominates } x) \text{ and } (x \text{ dominates } r^*)\}\right), \qquad (3.8)$$

where μ denotes the Lebesgue measure and a is an arbitrary element of $\mathscr{A}$.

S–metric measures the overall quality of an approximation. It is a unary quality indicator, so it creates a total order between the non–dominated sets. The S–metric is compatible with all the out–performance relations if the reference point is well chosen.

It has been reported that the S–metric has a bias towards the central zone of the Pareto Front and towards convex areas of the Pareto Front [78]. Some research is being done to solve these problems [78]. The election of the reference point is vital for the good behavior of the S–metric. Different reference points may result in different evaluations. Also, the reference point must be dominated by all possible vectors in the comparison, otherwise the evaluations of the S–metric may not make any sense, for example, reporting negative values.

There are efficient algorithms to calculate the value of the S–metric. Fonseca et al. [23] proposed an $O(|\mathscr{A}|\ log\ |\mathscr{A}|)$ algorithm for three dimensions. There is another algorithm with complexity $O(|\mathscr{A}|^{d/2})$ [6] that works on any dimension. This last algorithm is based on another algorithm that solves Klee's measure problem [55]. The S–metric has been used to lead the search process in some multi–objective algorithms [51].

3.6.5 ϵ–Indicator

A family of performance measures based on ϵ–dominance [47] is proposed in [83]. The binary ϵ–indicator $B_\epsilon(\mathscr{A}, \mathscr{B})$ is the minimum factor for which

all components of all vectors in $\mathscr{B}$ must be multiplied in order to have each vector in $\mathscr{B}$ dominated or equal to some vector in $\mathscr{A}$. If we substitute the set $\mathscr{B}$ for a reference set $\mathscr{R}$, we obtain the unary ϵ–indicator, denoted $I_\epsilon(\mathscr{A})$.

There is another version of the ϵ–indicator mentioned above, known as the additive ϵ–indicator. The binary additive ϵ–indicator, $B_{+\epsilon}(\mathscr{A}, \mathscr{B})$ is the minimum value that must be added to all components of all vectors in $\mathscr{B}$, in order to have each vector in $\mathscr{B}$ dominated or equal to some vector in $\mathscr{A}$. The unary additive ϵ–indicator is obtained when we substitute the set $\mathscr{B}$ by a reference set $\mathscr{R}$.

$B_\epsilon(\mathscr{A}, \mathscr{B})$ can be calculated with the formula:

$$B_\epsilon(\mathscr{A}, \mathscr{B}) = \overset{max}{b\in\mathscr{B}} \overset{min}{a\in\mathscr{A}} \overset{max}{1\leq i\leq m} \frac{a^{(i)}}{b^{(i)}}, \tag{3.9}$$

where the vectors in the approximations are in $\mathbf{R}^m$ and $a^{(i)}$ stands for the i–th component of vector a.

This family of quality indicators have gained much attention in recent years. They have an intuitive meaning, they are very easy to program and they are relatively cheap to calculate. Using the binary multiplicative version it is possible to construct a compatible and complete comparison method. To construct such a comparison method, use the function [83]: $F = (I_\epsilon(\mathscr{A}, \mathscr{B}) \leq 1 \wedge I_\epsilon(\mathscr{B}, \mathscr{A}) > 1)$. This formula is true if and only if $\mathscr{A}$ weakly out–performs $\mathscr{B}$, so the comparison method $C_{I_\epsilon, F}$ is compatible and complete.

The members of the ϵ–indicator family are defined based on concepts of dominance only, but they are often used to evaluate the general quality of the non–dominated sets.

3.6.6 Generational Distance

Generational distance is designed to measure the convergence of a non–dominated set. For a NS $\mathscr{A}$ and a reference set $\mathscr{R}$, Generational Distance

(GD) is defined as [71]:

$$GD(\mathscr{A}) = \frac{\sqrt{\sum_{i=1}^{|\mathscr{A}|} d_i}}{|\mathscr{A}|}, \tag{3.10}$$

where d_i is the distance between the i–th element of $\mathscr{A}$ and the nearest element of $\mathscr{R}$. Smaller values of GD mean a better convergence of the set. This performance measure is intuitive, easy to calculate and creates a total order between non–dominated sets. Unfortunately, GD depends on a reference set and, in general, is not even weakly compatible with weak out–performance. Adding more vectors to a non–dominated set can degrade its evaluations, so it has no property of monotony. Nevertheless, this performance measure is still popular because it is easy to use.

3.6.7 R–Metrics

Hansen and Jaszkiewicz [35] define several quality indicators based on concepts of probability. Let $U \subset U_C$ be the set of utility functions used to model all possible DM preferences for a multi–objective problem. For $u \in U$, let $u^*(\mathscr{A})$ denote the maximum value reached by u in $\mathscr{A}$. Let $p(u)$ be the probability that the DM chooses $u \in U$. The binary quality indicator $R1$ is defined as follows:

$$R1(\mathscr{A}, \mathscr{B}, U, p) = \int_{u \in U} C(\mathscr{A}, \mathscr{B}, u, p) p(u) du, \tag{3.11}$$

where:

$$C(\mathscr{A}, \mathscr{B}, u) = \begin{cases} 1 & \text{if } u^*(\mathscr{A}) > u^*(\mathscr{B}) \\ 1/2 & \text{if } u^*(\mathscr{A}) = u^*(\mathscr{B}) \\ 0 & \text{if } u^*(\mathscr{A}) < u^*(\mathscr{B}). \end{cases} \tag{3.12}$$

An intuitive (but not completely accurate) interpretation of $R1$ is that it counts how many utility functions $\mathscr{A}$ provide a better solution than $\mathscr{B}$, but considers the probability of these utility functions to be the chosen one. $\mathscr{A}$ is better than $\mathscr{B}$ if $R1(\mathscr{A}, \mathscr{B}) > 1/2$, $\mathscr{A}$ is equal to $\mathscr{B}$ if $R1(\mathscr{A}, \mathscr{B}) = 1/2$ and $\mathscr{A}$ is worse than $\mathscr{B}$ if $R1(\mathscr{A}, \mathscr{B}) < 1/2$. $R1$ is compatible with the out–performance relations under some special conditions [35].

A disadvantage of $R1$ is that it can induce cycles. For example, it is possible to have situations where $R1(\mathscr{A}, \mathscr{B}) > 1/2$, $R1(\mathscr{B}, \mathscr{C}) > 1/2$ and $R1(\mathscr{C}, \mathscr{A}) > 1/2$. A variation of $R1$ is the unary quality indicator $R1_{\mathscr{R}}$:

$$R1_{\mathscr{R}}(\mathscr{A}, U, p) = R1(\mathscr{A}, \mathscr{R}, U, p), \tag{3.13}$$

where $\mathscr{R}$ is a reference set. $R1_{\mathscr{R}}$ is weakly compatible with the out–performance relations, but it is not compatible even with complete out–performance. Another disadvantage of $R1_{\mathscr{R}}$ is that it can not detect differences in the quality of approximations that are completely out–performed by $\mathscr{R}$, because the value of $R1_{\mathscr{R}}$ for those sets is zero.

$R1(\mathscr{A}, \mathscr{B}, U, p) > 1/2$ can be interpreted as "it is more likely that $\mathscr{A}$ yields a better solution than $\mathscr{B}$", but $R1$ does not consider the differences in the amount of benefit provided by the solutions.

Imagine that all utility functions are defined so the value of different utility functions can be added. The expected benefit from an approximation can be calculated as:

$$E(u^*(\mathscr{A})) = \int_{u \in U} u^*(\mathscr{A}) p(u) du. \tag{3.14}$$

The expected value of the utility of an approximation set is, itself, a good unary quality indicator. Suppose that all utility functions $u(z)$ are formulated to measure the income in dollars obtained from the candidate solution z. $E(u^*(\mathscr{A}))$ represents the expected income that we can obtain

using $\mathscr{A}$. It is hard to think of a better quality indicator than that. Hansen and Jaszkiewicz [35] define a binary QI based on Formula 3.14 known as $R2$:

$$R2(\mathscr{A}, \mathscr{B}, U, p) = E(u^*(\mathscr{A})) - E(u^*(\mathscr{B})). \tag{3.15}$$

$\mathscr{A}$ is better than $\mathscr{B}$ if $R2(\mathscr{A}, \mathscr{B}, U, p) > 0$. $R2$ is weakly compatible with the out–performance relations, and is compatible under the same conditions as $R1$.

Other "R"–metrics based on $R1$ and $R2$ are also defined in [35]. One of the main disadvantages of the R–metrics is that they require a considerable amount of information in order to be used. The probability function $p(z)$ and the set of utility functions U are not known "a priori" and they can be very difficult to obtain. Hansen and Jaszkiewicz [35] give some recommendations about how to set $p(z)$ and U in an artificial way, but we must remember that the evaluations of the R–metrics are heavily affected by those parameters and the selection of a particular set of parameters must be justified.

When Hansen and Jaszkiewicz designed the R–metrics, one of their goals was to use those quality indicators in the search process. For example, in an interactive approach, we can obtain an initial approximation to the Pareto Front. Next, based on that approximation, the DM can clarify preferences, select a set of utility functions and decide which of them are more likely to be the best ones. Then, we can include this information in the R–metrics and use it as a stopping criterion.

Unfortunately, when we want to evaluate the performance of different MOEAs, there is a major drawback in the use of the R–metrics. The general procedure to compare several algorithms, is to run the algorithms in a benchmark. Next we evaluate the outputs of the algorithms using one or more quality indicators. Then we perform a statistical study of the evaluations of the outputs. In the end, we want to infer if one MOEA performs better than another in the "general case". The R–metrics are designed to introduce very specific information of a particular Decision Maker in their

evaluations. If a DM provides a density functions $p(z)$ and a set of utility functions U, when using this information the only inference we can make is that a MOEA performs well for these particular DM preferences but not for a multi–objective problem in general.

It can be argued that most quality indicators have parameters to adjust, and different sets of parameters represent different preferences. That is true, but the disadvantage of the R–metrics is that they are explicitly designed to introduce preferences. The most favorable case for the R–metrics, when the DM preferences are completely known, is the least favorable case when trying to make general inferences about the performance of a MOEA.

3.6.8 Dominance Ranking

Next, we give the definition of what we call a ranker in the context of quality indicators for approximation sets:

Definition 32. *Given a finite number m of approximation sets, a ranker is a procedure that ranks the m approximations from the best to the worst based on some definition of quality. The lower the rank, the better the approximation. The ranking is created considering all the information of the m sets.*

The difference between a unary quality indicator and a ranker is that a unary QI evaluates an approximation ignoring the information of other approximations. The evaluations of a ranker are relative to the m approximations we are evaluating.

In [43] a ranker is described, called Dominance Ranking (DR). DR compares each of the m approximations $\mathscr{A}_i$, $i \in \{1, \ldots, m\}$, with the others using weak out-performance, and ranks them based on how many times each set is out–performed. The rank of a set is defined by the formula:

$$rank(\mathscr{A}_i) = 1 + \sum_{j=1}^{m} F(\mathscr{A}_j, \mathscr{A}_i), \tag{3.16}$$

where:

$$F(\mathscr{A},\mathscr{B}) = \begin{cases} 1 & \text{if } \mathscr{A}\ O_W\ \mathscr{B} \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

The DR is intended to be used in combination with a statistical rank test to detect if there is a significative difference in the ranks of the sets.

DR is extremely easy to understand and to implement. Its computational complexity is relatively low, $O(m^2|\mathscr{M}|^2)$ in the worst case, where $\mathscr{M}$ is the biggest set in the comparison. Also, it is not biased towards any DM preference.

A disadvantage of DR is that it does not consider diversity in its evaluations, so it cannot detect differences in the quality of two approximations when they are not comparable under the out–performance relations. Another disadvantage is that it can give some strange evaluations for some configuration of sets. For example, consider the approximations in Figure 3.3. It is evident that $\mathscr{G}$ is the worst set. It can also be seen that $\mathscr{G}$ does not follow exactly the same arrangement of the other approximations. $\mathscr{G}$ is "shifted" a little bit to the left and a little bit down. As a consequence, the leftmost element of $\mathscr{G}$ is dominated by no vector in $\mathscr{B}$, $\mathscr{D}$ and $\mathscr{F}$, and the rightmost element of $\mathscr{G}$ is dominated by no vector in $\mathscr{A}$, $\mathscr{C}$ and $\mathscr{D}$. Thus, when we apply the weak out–performance relation, $\mathscr{G}$ is out–performed by no other approximation! At the end, $\mathscr{G}$ will be located in the first rank, together with $\mathscr{A}$ and $\mathscr{B}$, contrary to what is expected.

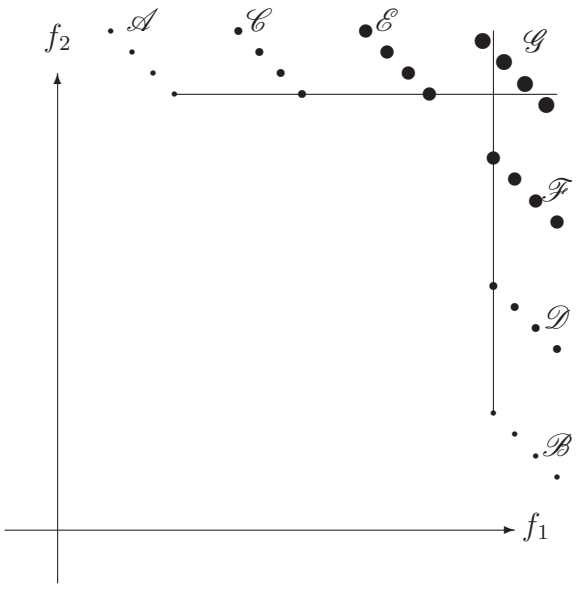

Figure 3.3: A misleading case for the Dominance Ranking.

Chapter 4

Problem Statement

4.1 Introduction

In single objective optimization, it is relatively easy to compare the performance of stochastic algorithms. The value of the objective function is used to measure the quality of a solution. Thus, the quality of the outputs of algorithms is the value of the objective function of the best individual found. The output of the algorithm is considered a random variable X, and it is desirable to know the distribution of this variable. Knowing the distribution allows us to infer how likely and with what consistency the algorithm approximates the optimal solution. A typical procedure is to run the algorithm several times to obtain a sample of its outputs, and use a measure of central tendency, as the mean and median, and a measure of variability, as the variance. This way we can compare two algorithms, comparing the mean, variance or any other measure, of the sample of their outputs. This procedure has been used in several papers, for example in [60].

It is even better to use statistical tests to compare stochastic algorithms. There are many statistical tests designed to compare two or more samples of a distribution, and to infer if there is a significant difference between the samples. Statistical tests have a lot of theoretical support and the conclusions

derived from them are more trustable. For an introduction to statistical tests, we recommend consulting [64].

In multi–objective optimization, the comparison between MOEAs is more difficult because the outputs of a MOEA are sets of vectors. In [43], three approaches to compare the performance of MOEAs (based only on the outputs of the algorithms) are described: quality indicator transformation, attainment functions and ranking.

4.2 Quality Indicator Transformation

In this approach, the idea is to transform each approximation into a single figure of merit using a unary quality indicator (binary quality indicators could be used also). Assuming that a QI gives a bigger (or smaller, depending on how it is defined) value to better approximations, we can use the value of the QI as a random variable X produced by the multi–objective algorithm. This way we can make a statistical analysis based on the transformation of a sample of approximations into a sample of real numbers.

Using a quality indicator to perform a statistical analysis has several advantages. Transforming an approximation into a real number simplifies the statistical analysis. The problem is that there are many quality indicators with both advantages and disadvantages, and it is not clear which one must be used. There is the alternative of using a set of quality indicators, but it is not clear what set must be used and how to interpret the different results of the QIs.

4.3 Attainment Function

An attainment function [24] is a function defined on the objective function space. It gives the probability for a vector $z \in \mathbf{R}^m$ to be dominated by, or equal to, an element of an approximation $\mathscr{A}$, generated by a MOEA

M. Let $\mathscr{A}_M$ represent an arbitrary outcome of MOEA M, let $z \in \mathbf{R}^m$ be an element of objective function space, and denote by $\mathscr{A} \trianglelefteq z$, that z is dominated by an element of $\mathscr{A}$ or contained in $\mathscr{A}$. The attainment function $\alpha_{\mathscr{A}_M}(\cdot) : \mathbf{R}^m \to [0,1]$ is defined as:

$$\alpha_{\mathscr{A}_M}(z) = P(\mathscr{A}_M \trianglelefteq z). \tag{4.1}$$

In practice, the attainment function can be estimated as follows. Define the function:

$$\mathbf{I}(\mathscr{A}, z) = \begin{cases} 1 & \text{if } \mathscr{A} \trianglelefteq z \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

The empirical attainment function, based on n outputs of a MOEA, is defined as:

$$\alpha_n(z) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{I}(\mathscr{A}, z). \tag{4.3}$$

Given the attainment function of two MOEAs, a combination of statistical tests and visual comparison can be used to determine which MOEA generates better sets (for details, see [43]).

The attainment function approach has many advantages. For example, most of the information of the approximations is preserved, unlike the quality indicator approach. Also, arguably all the aspects of quality of an approximation are intrinsically considered when using attainment functions. The disadvantage is that for higher dimensions ($d > 3$), the empirical attainment function is hard to evaluate and visualize. Since most of the experimentation in multi–objective optimization is done in low dimensions, and the Pareto Optimality Criteria loses strength as the number of dimension increases, the dimensionality may not be a big problem for the use of attainment functions.

4.4 Ranking

Knowles et al. [43] propose the following approach. Imagine we want to compare q MOEAs: MOEA_1, MOEA_2, ..., MOEA_q, and we make r_i runs for MOEA_i, $i \in \{1, \ldots, q\}$. We end up with $r = \sum_{i=1}^{q} r_i$ approximations, each one associated with a different MOEA. Next, we rank the r sets using Dominance Ranking (see Section 3.6.8), so each approximation $\mathscr{A}$ has associated a figure of merit equal to $rank(\mathscr{A})$. This way, each MOEA has associated a list of integer numbers (the ranks) in the form $(rank(\mathscr{A}_1^j), rank(\mathscr{A}_2^j), \ldots, rank(\mathscr{A}_r^j))$, where $\mathscr{A}_i^j$ denotes the i–th output of MOEA_j. Finally, a statistical rank test is used to find whether there are significant differences between the ranks of the MOEAs.

Actually, this procedure is a particular case of the indicator transformation. Any quality indicator could be used to generate a ranking between a combination of non–dominated sets. The advantages of the ranking is that it is easy to understand and implement, and that no further assumptions are made about Decision Maker preferences.

The disadvantages of this approach is that the Dominance Ranking cannot detect differences in diversity, and it can produce some misleading rankings (see Section 3.6.8). In [43] it is recommended to complement this approach with attainment functions and quality indicators to obtain a finer evaluation, but no detailed procedure is given.

4.5 The Problem

In this work we focus on the ranking approach mentioned above. A quality indicator or a ranker is necessary to rank the approximations and make a rank statistical test. Unfortunately, all known quality indicators have disadvantages. They introduce strong biases in their evaluations, do not consider all aspects of quality, etc.

The problem and area of opportunity is the following:

Problem Statement. *Comparing the performance of two MOEAs is an open problem in multi–objective optimization. When a researcher wants to compare the performance of two MOEAs, it is not clear what methodology must be used. As a consequence, it is difficult to validate new algorithms. It is common to use several quality indicators at the same time, but it is not clear how the different evaluations must be interpreted, and the use of many quality indicators at the same time does not guarantee to obtain a good evaluation. This lack of a good methodology to compare MOEAs has slowed down the advance of multi–objective optimization.*

4.6 The Objective

Main Objective 1. *The main objective of this work is to create a ranker able to rank a collection of approximation sets according to our intuition of when one approximation is better than another.*

Next we present some of the characteristics that we want in our ranker:

- *We consider our ranker as a model of the analyst preferences.* Our goal is to model a complex system, the analyst preferences. This system has been very difficult to model in the past, and different approaches have had different degrees of success. In many cases we have an intuitive notion of when one approximation is better than another. This intuitive notion is what we want to imitate in our model. The goal is to design a model accurate enough to make acceptable inferences when evaluating the performance of MOEAs.

- *We want to include all aspects of quality.* More specifically, we want to consider both convergence and diversity in our model. Ignoring one of those aspects of quality may result in incomplete evaluations.

- *We want to transform the evaluation of an approximation into a single figure of merit.* Evaluating different aspects of quality independently or

using several indicators has a disadvantage, we still have to decide how to combine the different evaluations. Quality indicators that evaluate different aspects of quality are important and their creation must be encouraged, but the ideal scenario is to have a proper way to combine them. In our model we combine convergence and diversity in a single evaluation.

- *We want to reduce the number of parameters as much as possible.* Adjusting parameters can be difficult, especially if it is not clear what are the consequences of changing a parameter. On the other hand, the existence of parameters in a model has some advantages. Parameters allow us to make adjustments for particular conditions. We want our model to have a small number of parameters, and we want them to be easy to adjust.

- *We want our model to have desirable properties.* When engineers create a model of complex systems, for example the maquette of a bridge, there are some properties that guarantee that the model has a behavior similar to the real system. For example, there are properties related with geometry (the scale in the main axis must be adequate). There are also properties related with statics and dynamics (the response of the model to mechanic elements like vibrations and stress must be similar to that of the real system). For quality indicators, there are some desirable properties that have been identified. For example monotony, relativity, compatibility with out–performance relations (as is defined by Hansen and Jaszkiewicz [35]), etc.

- *We want our model to be easy to calculate.* Low computational complexity and simplicity are desirable properties for any quality indicator or ranker. Unfortunately, in most cases the relationship between effectiveness and simplicity is inversely proportional. The most effective algorithms to evaluate an approximations are the most demanding in

computational effort (for example, the S–metric and attainment functions). We are trying to model a very complex system and the calculation of our model may not be cheap.

- *We do not want to depend on extra information of the multi–objective problem.* Including information about the multi–objective problem may be very useful, because we can adjust the evaluations of a quality indicator. The drawback is that our model may become dependent on the availability of such information. We want our model to be able to work under the worst conditions, when there is no other information about the problem, only the approximations to compare.

In order to be able to compare all possible approximations, Hansen and Jaszkiewicz [35] introduce very specific information of a Decision Maker in R–metrics. This approach has the disadvantage that it introduces a big bias in the evaluations of R–metrics, losing generality. Ziztler et al. [83] decided not to include any assumption or preference, besides General Assumption 1, in the design of the compatible and complete comparison method. The disadvantage of this approach is that it is not possible to detect evident differences in quality between some approximations. The rest of the quality indicators are somewhere in the middle, with different capacity to detect differences in quality and introducing some biases.

It is a common belief that introducing any assumption to a quality indicator will make its evaluations biased. In this work, we assume that there must be some extra information that can be introduced to improve the inference power of a quality indicator without introducing important biases. We deduce this from the behavior of the compatible and complete comparison method. CCCM has no biases, but it is clear that something is lacking in its evaluations. We shall try to identify such information and to introduce it into our model of quality.

Another objective of this thesis it to create a benchmark to study the behavior of the quality indicators under some synthetic cases. The only bench-

mark for quality indicators published, as far as we know, is that of Okabe et al. [54]. In that work, the authors create six synthetic approximations with different levels of quality. The disadvantage of that benchmark is that some interpretation is required to decide the correct order, from the best to the worst, between the sets. For example, one of the synthetic approximations has good convergence, good extension and bad distribution, while another approximation has good convergence, bad extension and good distribution. But, it is not clear what approximation is better, because it can be discussed whether we prefer extension over distribution or distribution over extension. As a consequence, it is not clear what to expect from a quality indicator when evaluating these approximations.

We want to create some test cases where it is clear whether one approximation is better than another, or at least, where we can justify why we consider some approximations better than others. This way, when using a quality indicator in the test cases, we know exactly what is the desired behavior of the QI.

Chapter 5

The G–Ranker

5.1 Introduction

In this chapter we describe our ranker and some theoretical concepts about convergence and diversity. We begin with a list of non–dominated sets $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$. The elements of this list are supposed to be generated by different MOEAs, but this information is not necessary in order to rank sets. Thus, we only consider that we need to rank the sets in L and assume that the information of what MOEA generated which approximation is known and used in a posterior analysis. It is possible for two MOEAs to generate the same set, or for a MOEA to generate the same set more than once. For that reason we put the approximations in a list, not in a set. In order to describe operations with the elements of L, we use an auxiliary set of indices $C = \{1, 2, \ldots, r\}$. Each element $i \in C$ is associated with the i–th element of L, the approximation $\mathscr{A}_i$. Originally, when we designed our ranker, we considered some ideas from geometry and computational geometry, such as Voronoi diagrams, balls, etc., for this reason we call our approach the (Geometric) G–Ranker.

The G–Ranker has two main components, a convergence component and a diversity component. In the following subsections, we explain the different

mechanisms that conform the G–Ranker and the ideas behind them.

5.2 Convergence Component

As mentioned before, convergence is the most important characteristic of a NS, so it is necessary to consider this property. An important problem is how to define convergence. Some of the first ideas were that a multi–objective algorithm has converged to the optimum if it has found one or more elements of the Pareto Front. We follow this idea and present a definition of convergence.

Definition 33. *An approximation $\mathscr{A}$ has converged to the Pareto Front ($\mathscr{PF}$), if $\mathscr{A} \cap \mathscr{PF} \neq \emptyset$.*

So, an approximation has converged to the Pareto Front if it has elements in common with the Pareto Front. Definition 33 has some disadvantages. The main disadvantage is that we need to know the Pareto Front in order to apply this definition. We assume that the only information we have are the approximations in list L. For this reason we use a relaxed version of Definition 33. Let $S_1 = \bigcup_{i \in C} \mathscr{A}_i$, and $\mathscr{R}_1 = ND(S_1)$. $\mathscr{R}_1$ consists on all the non–dominated vectors from the sets in L and can be used as a "relative" Pareto Front (or a reference set) based only on the information of the approximations we are evaluating. Next, we can create a list of the indices of the approximations that converged to the "relative" Pareto Front with the following formula:

$$C_1 = \{i \in C \mid \mathscr{A}_i \cap \mathscr{R}_1 \neq \emptyset\}. \tag{5.1}$$

The sets that have "converged" are those whose index is in C_1. Another disadvantage of Definition 33 is that it classifies the sets in only two classes: the approximations that have elements in the Pareto Front and those that have not. But no further distinctions between the sets are possible. It is possible to have big differences in the quality of the sets whose indices are

not in C_1. In order to detect such differences we proceed as follows. First, we ignore the sets whose indices are in C_1 (we already know that they are the best ones). Next, we detect the non–dominated elements for the rest of the approximations. These non–dominated elements represent a new "relative" Pareto Front. Finally, the approximations that contribute at least one element to the new relative Pareto Front are the second best approximations according to convergence.

In a more algorithmic way, given the original set of indices and the set C_1, firstly we update the list of indices in the following way: $C = C - C_1$. Next, we recalculate the union of the sets $S_2 = \bigcup_{i \in C} \mathscr{A}_i$, with the updated list of indices C. Then, we recalculate the non–dominated elements $\mathscr{R}_2 = ND(S_2)$. Finally, we define a new set of indices:

$$C_2 = \{i \in C \mid \mathscr{A}_i \cap \mathscr{R}_2 \neq \emptyset\}. \tag{5.2}$$

The sets with the second best convergence are those whose indices are in C_2. We can repeat this process until all the indices in C have been located in a class C_k. The algorithm for our convergence component is shown in Figure 5.1.

The convergence operator of the G–Ranker is an extension of the non–dominated sorting proposed by Goldberg [32]. Goldberg's non–dominated sorting (GNS) is designed to classify vectors according to the relation of dominance between them. In the first class of GNS we have the vectors that are not dominated by any other vector. In the second class we have the vectors that are dominated only by members of the first class. The following classes consist of vectors that are dominated only by vectors in previous classes. Our convergence operator classifies sets of vectors based on Definition 33.

After we use our convergence procedure, we assign an integer number $conv(\mathscr{A}_i)$ to each approximation in the list $(\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$. This integer number is the evaluation of the convergence of approximation $\mathscr{A}_i$ and is

Require: A list of r non–dominated sets $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$.
Ensure: A partition of $C = \{1, 2, \ldots, r\}$ into K classes $C_1, C_2, \ldots, C_K$. Each C_k contains the indices of the k–est best non–dominated sets according to convergence.

1: $C = \{1, 2, \ldots, r\}$
2: $k = 0$
3: **repeat**
4: $\quad k = k + 1$
5: $\quad S_k = \bigcup_{i \in C} A_i$
6: $\quad \mathscr{R}_k = ND(S_k)$
7: $\quad C_k = \{i \mid \mathscr{A}_i \cap \mathscr{R}_k \neq \emptyset\}$
8: $\quad C = C - C_k$
9: **until** $C = \emptyset$

Figure 5.1: Convergence Algorithm.

defined as:

$$conv(\mathscr{A}_i) = k \mid i \in C_k, \tag{5.3}$$

the smaller the value of $conv(\mathscr{A}_i)$, the better the evaluation of convergence of $\mathscr{A}_i$. For simplicity, we use the expression "$\mathscr{A}_i$ is in class C_k" as a synonym of "the index i of $\mathscr{A}_i$ is an element of C_k"

As an illustrative example, consider the approximations in Figure 5.2. We have that $r = 5$, $L = (\mathscr{A}_1, \mathscr{A}_2, \mathscr{A}_3, \mathscr{A}_4, \mathscr{A}_5)$, $C = \{1, 2, 3, 4, 5\}$. Considering the union of all approximations in L, the "relative" Pareto Front are the vectors enclosed by circles in Figure 5.3. The only sets that have elements in the "relative" Pareto Front are $\mathscr{A}_3$ and $\mathscr{A}_4$, so $C_1 = \{3, 4\}$. In order to find C_2, we discard the sets in C_1 and we find the new "relative" Pareto Front, whose elements are enclosed by circles in Figure 5.4. The approximations with elements in the new "relative" Pareto Front are $\mathscr{A}_1$ and $\mathscr{A}_2$, so $C_2 = \{1, 2\}$. It is evident that $C_3 = \{5\}$. The evaluation of convergence for the approximations is $(conv(\mathscr{A}_1), conv(\mathscr{A}_2), conv(\mathscr{A}_3), conv(\mathscr{A}_4), conv(\mathscr{A}_5)) = (2, 2, 1, 1, 3)$.

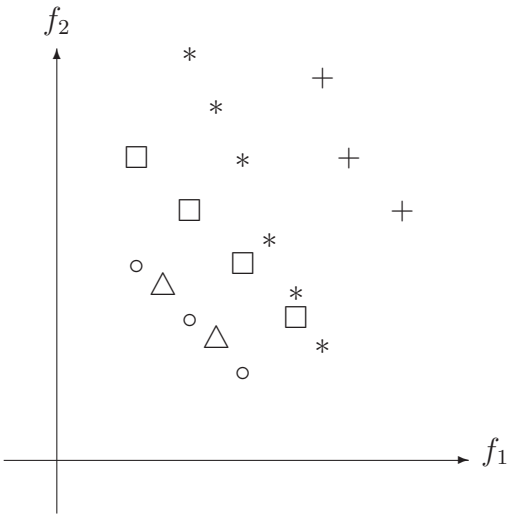

Figure 5.2: Five approximation sets, three classes of dominance. $\mathscr{A}_1$ ($\square$), $\mathscr{A}_2$ ($*$), $\mathscr{A}_3$ ($\triangle$), $\mathscr{A}_4$ ($\circ$), $\mathscr{A}_5$ ($+$).

The computational complexity of this convergence component in the worst case is $O(m^3|A_{max}|^2)$, A_{max} is the approximation with more elements and m is the number of approximations. The worst case occurs in cases where there is only one approximation per class.

5.3 Diversity Component

Measuring diversity has proven to be very difficult. One of the difficulties is that there is no unique definition of diversity. It is common to divide diversity in other aspects such as: number of vectors generated, evenness, extension, etc., and to create different quality indicators for these aspects. We want to create a model of diversity that considers all these aspects together. In order to create such model, it is necessary to find a good definition of diversity. In order to find that definition we introduce some theoretical concepts in this section.

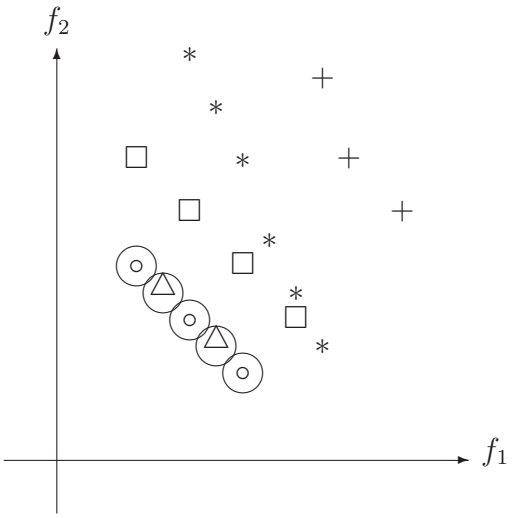

Figure 5.3: The first "relative" Pareto Front is found. $\mathscr{A}_1$ ($\square$), $\mathscr{A}_2$ ($*$), $\mathscr{A}_3$ ($\triangle$), $\mathscr{A}_4$ ($\circ$), $\mathscr{A}_5$ ($+$).

5.3.1 A New Interpretation of Diversity

The preferences of the Decision Maker are influenced by the approximation that is shown. The more and better information presented to the DM, the more "optimal" his/her decisions can be. Imagine the hypothetical situation where we know the whole Pareto Front and the Decision Maker is able to analyze all its elements. Under these conditions it is valid to assume that the DM can define perfectly his/her preferences[1] and select the real optimal solution. We call the preferences derived under these conditions "the True DM preferences" or "True preferences" and the optimal solution derived under these conditions "the True optimal solution".

In the following discussion, in order to simplify some explanations, we only consider approximations whose elements are also elements of the Pareto Front. In other words, we only consider approximations that are subsets of

[1]We call Decision Maker preferences the information from the DM that can allow us to identify a unique optimal solution for a multi–objective problem.

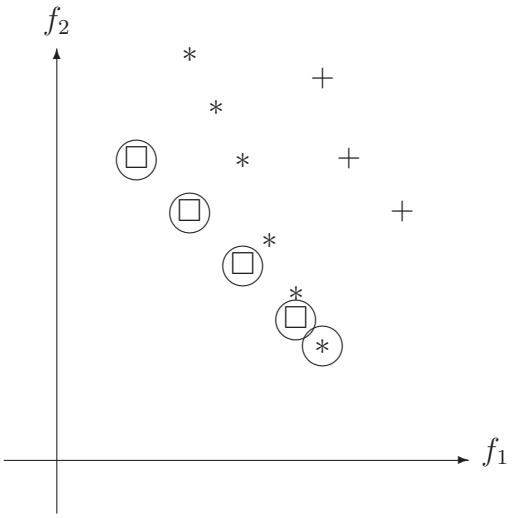

Figure 5.4: The second "relative" Pareto Front is found. $\mathscr{A}_1$ ($\square$), $\mathscr{A}_2$ ($*$), $\mathscr{A}_3$ ($\triangle$), $\mathscr{A}_4$ ($\circ$), $\mathscr{A}_5$ ($+$).

the Pareto Front.

Different Decision Makers may have different True preferences. Even the same DM may have different True preferences over time. We represent an arbitrary True preference with $\mathfrak{p}$. We denote by $z_{\mathfrak{p}}$ the true optimal solution associated with a True preference $\mathfrak{p}$[2]. We assume that each $z_{\mathfrak{p}}$ is an element of the Pareto Front. We denote with $\mathfrak{P}$ the set of all possible True DM preferences. We assume that the Pareto Front is the set of all possible True optimal solutions.

Remember that we are working with a posteriori approach, where the DM preferences are completely unknown a priori, the only information available is the model of multi–objective optimization problem (Formula 2.13). In Chapter 2, we present General Assumption 1, that can be related with convergence. There is another assumption related with diversity, that is widely

[2]This framework is similar to that of Hanzen and Jazckiewicz [35], where they represented the DM preferences using utility functions.

accepted in the a posteriori approach:

General Assumption 2. *A priori, all the elements of the Pareto Front are equally important, no element of the Pareto Front is preferred over the others.*

All Pareto Optimal solutions have the same potential to be the True optimal solution. There are some situations where General Assumption 2 is not accepted. For example, some algorithms search for specific zones of the Pareto Front known as "knees" [7]. In these cases, more information about the DM preferences is introduced and we lose generality. We consider General Assumption 2 as valid, as do most MOEAs.

Now, imagine that for any multi–objective problem, there exists an intensity function $p(\mathfrak{p})$ over the elements of $\mathfrak{P}$, related to the probability of $\mathfrak{p} \in \mathfrak{P}$ to be the True preference chosen by the Decision Maker. Also imagine a multi–objective problem whose Pareto Front is the one in Figure 5.5 and we begin to randomly select DM preferences from $\mathfrak{P}$ based on the intensity function $p(\mathfrak{p})$ associated with this multi–objective problem. For any DM preference chosen, we find its corresponding $z_{\mathfrak{p}}$ in the Pareto Front, and keep a record of how many times each element of the Pareto Front was the optimal solution. After a large number of experiments, we construct an intensity function $q(z)$ defined over the elements of the Pareto Front. Function $q(z)$ presents the probability of an element of the Pareto Front to be the True Pareto optimal solution of the multi–objective problem.

If, for the multi–objective problem described above, $q(z)$ is as shown in Figure 5.6, left side, we can conclude that vector b is more likely to be the True optimal solution, followed by vectors c and a, respectively. If $q(z)$ is as shown in Figure 5.6, right side, then we see that all the elements of the Pareto Front have the same probability to be the True optimal solution.

In a case with two objective functions where the Pareto Front is a continuous line, as in Figure 5.7 (points a, b, c, d, e and f are landmarks), we can also obtain an intensity function on the Pareto Front (at least in our

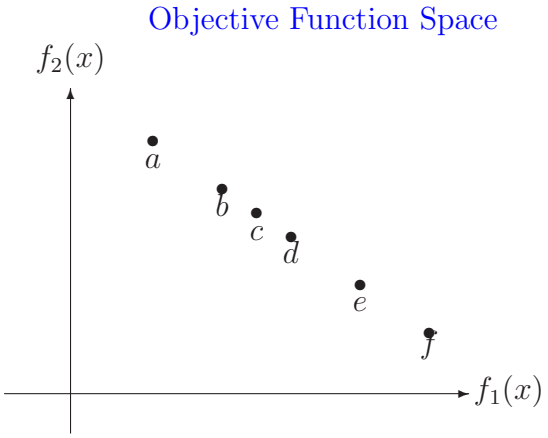

Figure 5.5: An example of a Pareto Front.

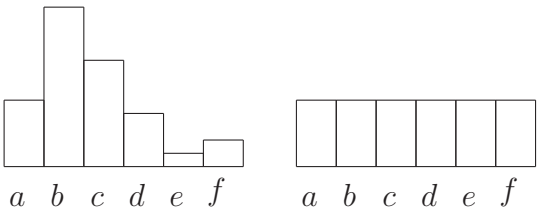

Figure 5.6: Two probability distributions.

imagination), using an intensity function on $\mathfrak{P}$. In Figure 5.8 we show two possible intensity functions $q(z)$, where the elements of the Pareto Front have been aligned horizontally. If the resulting $q(z)$ is as in Figure 5.8, left side, we see that the most promising zones of the Pareto Front are between b and c and in the neighborhood of e. In Figure 5.8, right side, we have that all the elements of the Pareto Front have the same probability to be $z_{\mathfrak{p}}$ for an arbitrary DM preference $\mathfrak{p}$.

Different probability functions represent different information about the DM preferences. From the point of view of entropy, a constant function $q(z)$ represents the minimum quantity of information, and it is harder to make

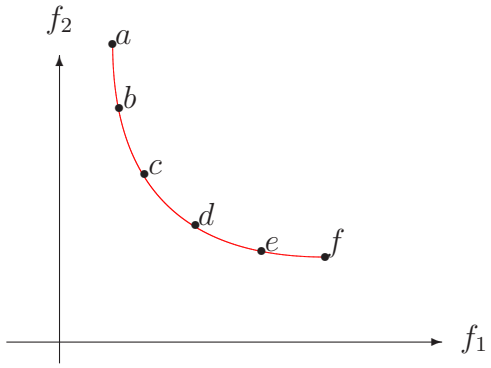

Figure 5.7: A continuous Pareto Front.

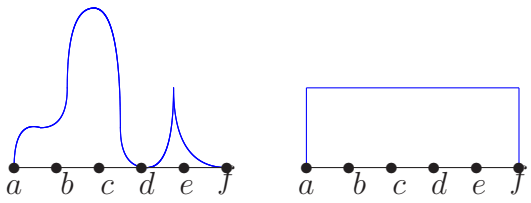

Figure 5.8: Two density functions for a continuous Pareto Front.

predictions of the final outcome. These are the conditions that we assume when we say that the DM preferences are unknown a priori. Based on this discussion we give an alternative definition to General Assumption 2.

Alternative Definition of General Assumption 2. *The "a priori" density function $q(z)$ that gives the probability of an element z of the Pareto Front to be the True optimal solution for the Decision Maker is a constant function.*

Now, imagine a Pareto Front with an infinite number of elements, like the one in Figure 5.7. Suppose that we obtain an approximation from a MOEA. A MOEA always generates a finite set of solutions. If the size of an

approximation is finite, and the size of the Pareto Front is infinite, and the Alternative Definition of General Assumption 2 is true, what is the probability for the True Optimal solution to be an element of the approximation? The answer is, of course, zero. Except for the cases where the Pareto Front has a relatively small number of elements, the True optimal solution will not be an element of any approximation. So, when choosing a final solution from an approximation, the Decision Maker will not get $z_{\mathfrak{p}}$, but at most will get an alternative solution, hopefully similar to the optimal one. Based on these ideas, we propose the following definition of diversity for subsets of the Pareto Front:

Definition 34. *The diversity of a subset of the Pareto Front is proportional to its capacity to provide a similar alternative solution to any element of the Pareto Front.*

Remember, we consider the Pareto Front as the set of all possible True optimal solutions. An approximation set that provides acceptable solutions to any DM preferences (not necessarily the optimal one) can be considered to have a good diversity. As mentioned before, the DM will take the solution most similar to $z_{\mathfrak{p}}$ from an approximation. If an approximation provides good substitutes for any possible optimal solution, the DM is more likely to find a satisfactory solution.

Unfortunately, Definition 34 is ambiguous and needs to be interpreted in order to create an useful indicator of diversity. What makes one alternative solution similar to another? How do we measure the capacity to provide an alternative solution? We consider that two solutions are similar if they provide a similar benefit to the Decision Maker. An intuitive assumption is that the similarity between two solutions is inversely proportional to their distance in objective function space. Similar assumptions are frequently accepted in evolutionary computation, for example in the concept of niches, or in the diversity mechanism of the MOEAs. From now on we consider that the smaller the distance between two solutions, the more similar the solutions

are. So, we need to find a way to evaluate the capacity of an approximation to provide solutions that may be similar to any arbitrary element z of the Pareto Front.

5.3.2 Some Ideas For Indicators of Diversity

An Error Function

In this subsection, we construct an error function whose value is higher for approximations with lower capacity to provide alternative solutions. This function is defined for subsets of the Pareto Front only.

Consider the Pareto Front in Figure 5.7. This front can be considered a continuous line in $\mathbf{R}^2$. Suppose that we straighten this Pareto Front, preserving the relative distance between its elements through the continuous line. Now, suppose that we have an approximation that is a subset of the Pareto Front. For each element z of the Pareto Front, we add a penalty based on the distance of the nearest element of the approximation. For example, we can use $d(z,a)^2$, where $d(x,y)$ is the Euclidean distance between x and y. Our penalty diversity measure is defined as:

$$E(\mathscr{A}) = \int_{z \in \mathscr{P}\mathscr{F}} d(z, a_{near})^2 dz \tag{5.4}$$

where $\mathscr{A}$ is the approximation set and a_{near} is the element of $\mathscr{A}$ nearest to z. An example of E can be seen in the top panel of Figure 5.9, where a, b, c, d, e and f are an approximation set.

As a curious note, it is a common assumption that the extreme elements of the Pareto Front must be found, but according to our error diversity measure this is not the best option. For example, in the top panel of Figure 5.9, we have six vectors (set $\mathscr{A}$) evenly distributed and extended to reach the extreme values of the Pareto Front. In the same Figure, bottom panel, we have another six vectors (set $\mathscr{B}$), that are evenly distributed. But, the extreme

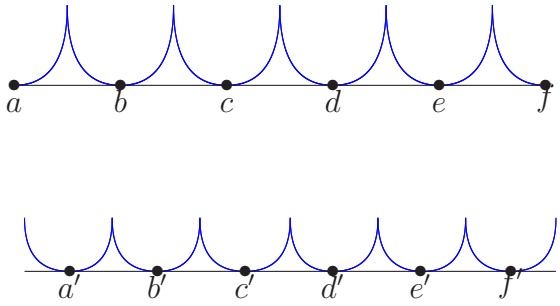

Figure 5.9: Penalty functions for two different configurations of vectors. $\mathscr{A} = \{a, b, c, d, e, f\}$, $\mathscr{B} = \{a', b', c', d', e', f'\}$. For this case, $E(\mathscr{A}) > E(\mathscr{B})$.

vectors of $\mathscr{B}$ are not coincident with the extreme vectors of the Pareto Front. The extreme vectors of $\mathscr{B}$ are separated from the nearest extreme element of the Pareto Front by half the distance between two contiguous elements of $\mathscr{B}$. The value of Formula 5.4 is higher in $\mathscr{A}$ compared to its value in $\mathscr{B}$, thus $\mathscr{B}$ has a better diversity. One reason for which $\mathscr{A}$ has a bigger error than $\mathscr{B}$, is that the maximum distance between an element of the Pareto Front and the nearest element of $\mathscr{A}$ is higher than the corresponding distance in $\mathscr{B}$.

Two of the desirable properties in a quality indicator are monotony and relativity. A quality indicator with monotony improves the evaluation of an approximation as we add new non–dominated vectors. A quality indicator with relativity gives the Pareto Front a better evaluation than any other non–dominated set. Formula 5.4 has both properties. When we add a new element a to an approximation, the elements of the Pareto Front nearer to a than to any other element of the approximation, will have a reduction in their penalty value reducing the value of Formula 5.4. The value of Formula 5.4 for the Pareto Front is zero, the minimum possible value.

The error quality measure, described above, has many interesting theoretical properties. But, unfortunately, there are many practical restrictions for its use. First, we need to know the Pareto Front in order to use it, and

for more than two dimensions, it is not possible to "straighten" the Pareto Front. A partial solution to this problem is: to project the approximations to a plane P, to define a domain in P and calculate Formula 5.4 using this domain and the projected elements of the approximation. Here, the domain is a substitute of the true Pareto Front, and the projection to the plane P is a substitute of "straightening" the Pareto Front. For a discussion of how to project a non–dominated set to a plane, see [19]. The main problem of the approach just described, is to define the domain for the integral in Formula 5.4. Different domains may result in different evaluations, and approximation $\mathscr{A}$ may be better than approximation $\mathscr{B}$ for one domain and the other way for another domain. Also, a Voronoi tessellation is necessary to find the nearest element of an approximation to an element of the domain. A Voronoi tessellation is expensive in high dimensions.

Due to all those problems, it is desirable to find another function that is cheaper to evaluate, that models diversity and that does not need knowledge about the Pareto Front.

A Benefit Function

Instead of penalizing based on how far the elements of an approximation are from the elements of the Pareto Front, we can sum based on how many of the elements of the Pareto Front are near to the approximation.

Again, consider a Pareto Front like the one in Figure 5.7 and suppose that we straighten this set and align it horizontally. Suppose that there exists a distance U such that if the distance between a candidate solution a and the True optimal solution $z_{\mathfrak{p}}$, is less than U, then the DM could use a instead of $z_{\mathfrak{p}}$. The elements of the Pareto Front whose distance from a is smaller than U represent "the zone of influence" (I_a) of a, see the top panel of Figure 5.10.

For an approximation set $\mathscr{A}$, the union of all the zones of influence of its elements ($I_{\mathscr{A}}$) can be used as a measure of its diversity. $I_{\mathscr{A}}$ is the set of all the elements of the Pareto Front for which $\mathscr{A}$ can produce an alternative

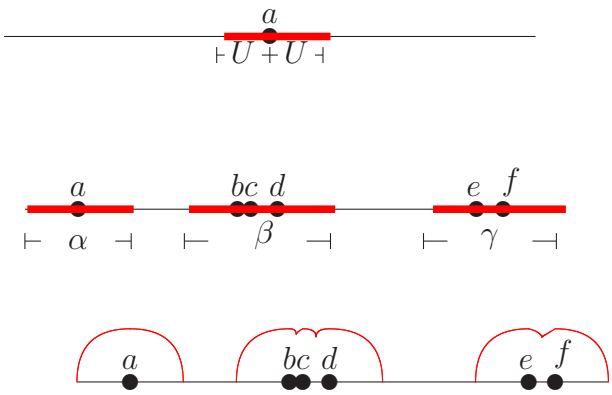

Figure 5.10: The radius U define a zone of influence I.

solution (assuming that we can find the value of U). We can measure the size of $I_{\mathscr{A}}$, denoted by $\mu(I_{\mathscr{A}})$, measuring the length of the union of all I_as. For the approximation in Figure 5.10, center, $\mu(I_{\mathscr{A}}) = \alpha + \beta + \gamma$.

For a fixed value of U, there is a point where adding more elements to an approximation does not improve the value of $\mu(I_{\mathscr{A}})$. In order to solve this problem, we can add a weight $w(z)$ to the elements z of the Pareto Front, depending on their closeness to an element of $\mathscr{A}$. Many weight functions can be proposed, but in this work we use the following:

$$w(z) = \begin{cases} \sqrt{U^2 - d(z, a_{near})^2} & \text{if } d(z, a_{near}) < U \\ 0 & \text{otherwise} \\ , \end{cases} \tag{5.5}$$

where a_{near} is the element of $\mathscr{A}$ nearest to z. The benefit quality indicator can be defined as:

$$B(A) = \int_{z \in \mathscr{PF}} w(z)dz.. \tag{5.6}$$

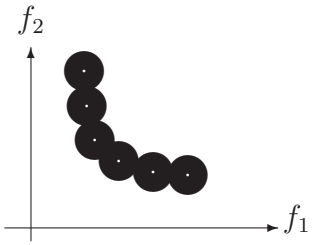

Figure 5.11: B2.

A graphical representation of $w(z)$ can be seen in the bottom panel of Figure 5.10, where we see that Formula 5.6 can be interpreted as the area of the union of the semicircles centered in the elements of $\mathscr{A}$. $B(\mathscr{A})$ has the property of monotony, because adding a new vector to $\mathscr{A}$ increases the area of the union of the semicircles. $B(\mathscr{A})$ also has relativity, because the Pareto Front has the maximum area possible for the union of semicircles.

The restrictions of the practical implementation of $B(\mathscr{A})$ are similar to those of $E(\mathscr{A})$, since we need to define a domain for the integral. In order to solve this problem, we introduce some modifications to $B(\mathscr{A})$. First, instead of semicircles, we can associate full circles to the elements of an approximation (imagine that we multiply $w(z)$ by two). This way, in order to evaluate the integral in Formula 5.6 we only need to calculate the area of the union of circles of radius U, centered on each element of $\mathscr{A}$.

Next, instead of "straightening" the Pareto Front, we can associate the circles (balls, from now on) to the elements of the approximation in their original positions, as is shown in Figure 5.11. Denote with $b(U, a)$ the ball of radius U and center a, and with $\mu(Y)$ the measure of the set Y (in 2d it means area, in 3d it means volume, etc.). We define the diversity quality indicator $B2$ as:

$$B2(\mathscr{A}, U) = \mu\Big(\bigcup_{a \in \mathscr{A}} b(U, a)\Big). \tag{5.7}$$

The definition of the diversity quality measure above, depends only on the position of the vectors in $\mathscr{A}$ and the balls associated to them. We do not need to know the Pareto Front in order to use $B2$. As a consequence, we can use $B2$ independently of the topology of the Pareto Front, even when the Pareto Front is discontinuous.

$B2$ has some desirable properties. For a fixed value of U, adding a new element to an approximation $\mathscr{A}$ does not degrade the value of $B2(\mathscr{A})$. Also, the value of $B2$ for the Pareto Front is not inferior to any approximation that is a subset of the Pareto Front.

By intuition, we can see that $B2$ has a good behavior as a diversity indicator. The measure of the union of balls associated to the vectors is directly proportional to the diversity of the vectors. A set $\mathscr{B} \subset \mathscr{PF}$ with good diversity has a high value of $B2(\mathscr{B})$, because there is little intersection between the balls (Figure 5.12, right side). A set $\mathscr{A} \subset \mathscr{PF}$ with poor diversity has a lot of intersections between the balls associated to its elements, and the value of $B2(\mathscr{A})$ decreases (Figure 5.12, left side).

The measure of the union of N balls can be calculated in $O(N\ log\ N)$ time for 2d and in $O(N^2)$ time in 3d [4]. No exact algorithm is known by the authors for more than three dimensions, but it can be approximated in $O(dN)$ time with a Monte Carlo integration, where d is the number of dimensions [66].

We know two approaches for the evaluation of diversity, that are similar to $B2$. One of them, the entropy based metric indicator (EM), is inspired by the concept of entropy and is proposed by Farhang-Mehr and Azarm [19]. EM consists of projecting the elements of an approximation $\mathscr{A}$ to a plane P. Denote by $\mathscr{A}'$ this projection of $\mathscr{A}$. Then, an influence function Ω with domain $P' \subset P$ is associated to each element of $\mathscr{A}'$. The influence function has positive values on P' that is inversely proportional to the distance from its "center". A density function is defined as:

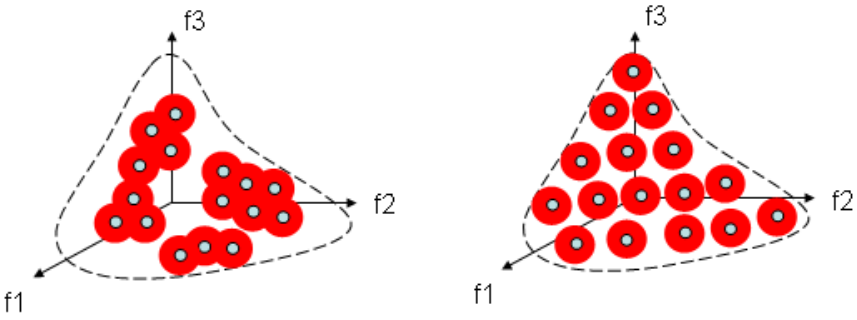

Figure 5.12: A graphic representation of the Diversity Component of the G–Ranker.

$$D(p) = \sum_{i=1}^{|\mathscr{A}|} I(d(a_i, p)), \tag{5.8}$$

where a_i is an element of $\mathscr{A}'$, $d(a_i, p)$ is the Euclidean distance between a_i and p. Finally, the diversity of the approximation $\mathscr{A}$ depends on the "flatness" of the density function described in Formula 5.8. This flatness is measured through the standard deviation of the value of D for some evenly distributed elements of P'.

Even when we use the same idea of influence functions, there are many differences between EM and $B2$. In $B2$, we do not project the elements of an approximation on a plane. In the theoretical development of $B2$ (when we define B), our weight function is based on the influence of the nearest element of the approximation, while in EM, the density function is based on the sum of the influence of all elements of the approximation. In our approach, the value of diversity is based on the area (volume, hyper–volume, etc) of a set of balls, not on the standard deviation of a set of weights, as in EM.

Another approach is the integrated sphere counting (ISC), proposed by Silva et al. in [65]. In ISC, in order to evaluate the diversity of an approximation $\mathscr{A}$, a list of values ISC_1, ISC_2, ..., ISC_n is calculated. To calculate the value of ISC_i, the first step is to propose a real value $r_i > 0$. Then, a vector $a_1 \in \mathscr{A}$ is chosen and marked as "selected", and all unmarked vectors $b \in \mathscr{A}$ whose distance from a_1 is smaller than r_i are marked as "eliminated". Next, a new unmarked vector a_2, is chosen and marked as "selected", and all unmarked vectors $b \in \mathscr{A}$ whose distance from a_2 is smaller than r_i are marked as "eliminated". This marking process is repeated until all vectors in $\mathscr{A}$ are marked. Let m_i be the number of vector marked as selected and denote by $\mu(b_i)$ the measure of a ball of radius r_i and let $ISC_i = \mu(b_i) \cdot m_1$. At the end, the value of diversity for $\mathscr{A}$ is $\sum_{i=1}^{n} ISC_i$.

Both ISC and $B2$ use the measure of balls in $\mathbf{R}^m$, but they have many differences. In $B2$ we use a single radius to calculate the diversity of an approximation, while in ISC they use several radii. In $B2$ there is no process of elimination based on the distances between vectors, but all vectors are used. The evaluations of $B2$ are based on the union of the balls associated to each vector in $\mathscr{A}$ while in ISC the intersections between balls are not considered.

The Parameter U

As we mentioned before, the overlap between the balls associated to each element of an approximation is very important for the good behavior of $B2$ as a diversity indicator. The overlap depends on the parameter U, so the numeric value of U must be chosen carefully. It is important to allow some level of overlap, so we can discriminate between sets with bad and good diversity. If U is too small, the overlap will be zero and the evaluations of $B2$ will not be related to the distribution of the points. At the same time, the value of U must not be too big, because the difference between the values of $B2$ for different approximations will be very small, even for approximations

with different diversity.

We want the value of U to be determined by the the configuration of the sets we are comparing. The procedure we use to calculate U is the following: suppose that we are evaluating the diversity of the approximations $\mathscr{A}_j$, $1 \leq j \leq m$. For each vector $a_i \in \mathscr{A}_j$ we find its nearest neighbors in $\mathscr{A}_j$ (the definition of "nearest neighbor" that we use here is that given in [76]). Next we calculate the mean value r_{ij} of the distances between a_i and its nearest neighbors. Finally, the value of U is calculated as half the mean value of the r_{ij}s of all vectors in all sets $\mathscr{A}_j$:

$$U = 0.5 \frac{\sum_{j=1}^{m} \sum_{i=1}^{|\mathscr{A}_j|} r_{ij}}{\sum_{j=1}^{m} |\mathscr{A}_j|}. \tag{5.9}$$

This value of U produces at least a small amount of overlap between the balls of the approximation with the worst diversity.

The concept of nearest neighbors given in [76] is the following: let A be a set of vectors $a \in \mathbf{R}^n$. For a given a, denote by $a^{(k)}$ the k–th component of a and denote by $B_{a^{(k)}+}$ the set vectors b in A so that $a^{(k)} < b^{(k)}$. One of the nearest neighbors of a is the element of $B_{a^{(k)}+}$ whose distance from a is the smallest with respect to the other elements of $B_{a^{(k)}+}$. This way we can find n nearest neighbors of a, one for each component of a. If we substitute the set $B_{a^{(k)}+}$ by the set $B_{a^{(k)}-}$ of the vectors b in A so that $a^{(k)} > b^{(k)}$, we can find other n nearest neighbors. This way we can find at most $2n$ nearest neighbors of a. Note that we say "at most", because some nearest neighbors may not exist or some of them may be repeated.

5.4 Combining Diversity and Convergence

In Section 5.2 we describe a procedure to classify a list of approximations to the Pareto Front, based on convergence. In Section 5.3, we describe a method

to measure the diversity of subsets of the Pareto Front. In this section we describe how we combine both methodologies to obtain a single method to evaluate the full quality of a list of approximations.

After evaluating convergence, we obtain a classification of the approximation in K classes C_k. Independently of their diversity, an approximation in a class C_i is better than an approximation in a class C_j, if $i < j$. We use diversity to detect differences between the approximations in the same class. So, our diversity method is applied K times, once per class C_k.

After the application of our diversity method, we obtain a list of real values associated to the list of approximations. These real values are the evaluation of diversity for the approximations. For a list of approximations ($\mathscr{A}_1$, $\mathscr{A}_2$, ..., $\mathscr{A}_r$) we get another list of real numbers ($div(\mathscr{A}_1)$, $div(\mathscr{A}_2)$, ..., $div(\mathscr{A}_r)$). $div(\mathscr{A}_i)$ is the evaluation of the diversity of approximation $\mathscr{A}_i$. We use both the convergence and diversity to rank the approximations from the best to the worst.

Several precautions must be taken when using the diversity operator. Going back to the example in Figure 5.2, suppose that we want to apply our diversity method to the approximations in class $C_2 = \{1, 2\}$, shown in the top panel of Figure 5.13. If we apply our diversity method directly, $\mathscr{A}_2$ will be considered as better than $\mathscr{A}_1$, because $\mathscr{A}_2$ has more diversity than $\mathscr{A}_1$. Unfortunately, this is incorrect, because even when $\mathscr{A}_2$ has more diversity, most of its elements are dominated by elements in $\mathscr{A}_1$. In order to prevent this situation, we evaluate our diversity method using only vectors that are not dominated by other vectors in the same class. For example, in the bottom panel of Figure 5.13, we eliminate all vectors in $\mathscr{A}_2$ that are dominated by vectors in $\mathscr{A}_1$. And, considering only these vectors, we use our diversity method.

So, given a list of approximations $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$ and a set of indices $C_k = \{j1, j2, \ldots, jn\}$, $1 \leq j1, \ldots, jn \leq r$, where each element $js \in C_k$ is associated with the approximation $\mathscr{A}_i \in L \mid js = i$, we proceed as follows.

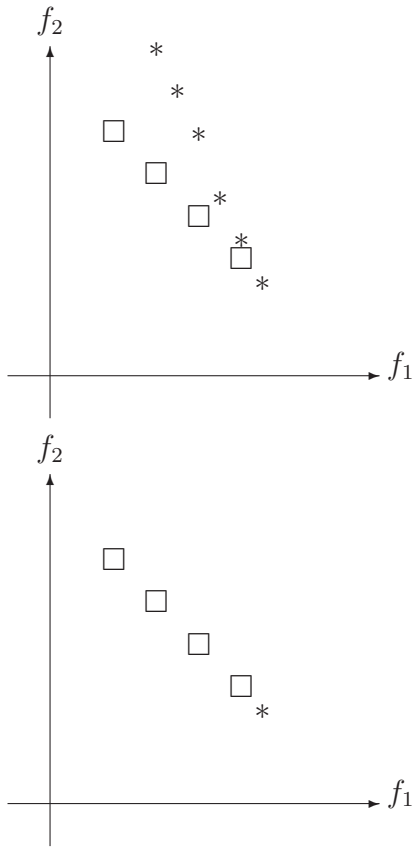

Figure 5.13: Two approximations in the same class of dominance. $\mathscr{A}_1$ ($\square$), $\mathscr{A}_2$ ($*$).

We use the "relative" Pareto Front $\mathscr{R}_k$ of this class ($\mathscr{R}_k = ND(\bigcup_{js \in C_k} \mathscr{A}_{js})$) to create a new list of n approximations $L' = (\mathscr{A}'_{j1}, \mathscr{A}'_{j2}, \ldots, \mathscr{A}'_{jn})$, defined as $\mathscr{A}'_{js} = \mathscr{A}_{js} \cap \mathscr{R}_k$. Then, we use the sets in L' to calculate the radius U and the diversity of each $\mathscr{A}'_{js}$, for $js \in C_k$. Finally, the diversity of the set $\mathscr{A}_{ji}$ is $div(\mathscr{A}_{ji}) = B2(\mathscr{A}'_{ji}, U)$. The algorithm is shown in Figure 5.14, where $CALCULATE_U$ is a function that uses Formula 5.9 to calculate U.

Require: A list of r approximation sets $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$ and a set of indices $C_k = \{j_1, j_2, \ldots, j_n\}$, $1 \leq j_1, \ldots, j_n \leq r$, $n \leq r$ where each element $js \in C_k$ is associated with the approximation $\mathscr{A}_i \in L \mid js = i$.
Ensure: The evaluation of diversity for $\mathscr{A}_{js}$, $js \in C_k$.

1: $S_k = \bigcup_{js \in C_k} A_{js}$
2: $\mathscr{R}_k = ND(S_k)$
3: **for all** $js \in C_k$ **do**
4: $\quad \mathscr{A}'_{js} = \mathscr{A}_{js} \cap \mathscr{R}_k$.
5: **end for**
6: $U = CALCULATE_U(\mathscr{A}'_{j1}, \mathscr{A}'_{j2}, \ldots, \mathscr{A}'_{jn})$
7: **for all** $js \in C_k$ **do**
8: $\quad div(\mathscr{A}_{js}) = B2(\mathscr{A}'_{js}, U)$.
9: **end for**

Figure 5.14: Diversity Algorithm.

5.5 Normalization

Our diversity operator (Formula 5.7) mixes the values of different objective functions. This can be problematic when there are huge differences in the scale of the objective functions, because one objective function may become more important than the others if its scale is bigger. For this reason it is important to normalize the values of the vectors in the approximations, so there is a balance in their scale.

Before applying our convergence and diversity operators, we normalize the vectors in all approximation sets using the maximal and minimal value of the set of non–dominated vectors. The normalization procedure is shown in Figure 5.15.

5.6 Ranking

After using our operators of convergence and diversity, we have two numbers, $conv(\mathscr{A}_i)$ and $div(\mathscr{A}_i)$ associated to each approximation $\mathscr{A}_i$. At the end, the

Require: A list of r approximation sets $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$ and a set of indices $C = \{1, 2, \ldots, r\}$, where the each element $i \in C$ is associated with the element $\mathscr{A}_i \in L$. The elements of the approximations are vectors $a \in \mathbf{R}^m$, $a^{(j)}$ stands for the j–th component of a.

Ensure: The same list of approximations but with the values of their elements normalized.

1: $S = \bigcup_{i \in C} A_i$
2: $\mathscr{R} = ND(S)$
3: **for all** $j = 1 : m$ **do**
4: $\quad MAX_j = a^{(j)} \mid a \in \mathscr{R}, \forall b \in \mathscr{R}, a^{(j)} \geq b^{(j)}$.
5: $\quad MIN_j = a^{(j)} \mid a \in \mathscr{R}, \forall b \in \mathscr{R}, a^{(j)} \leq b^{(j)}$.
6: **end for**
7: **for all** $\mathscr{A}_i$ **do**
8: $\quad$ **for all** $a \in \mathscr{A}_i$ **do**
9: $\quad\quad$ **for all** $j = 1 : m$ **do**
10: $\quad\quad\quad a^{(j)} = \frac{a^{(j)} - MIN_j}{MAX_j - MIN_j}$.
11: $\quad\quad$ **end for**
12: $\quad$ **end for**
13: **end for**

Figure 5.15: Normalization Algorithm.

rule to compare two approximations $\mathscr{A}$ and $\mathscr{B}$ is the following:

$$\begin{array}{lll} \mathscr{A} > \mathscr{B} & \text{if} & \left\{ \begin{array}{l} conv(\mathscr{A}) < conv(\mathscr{B}) \\ \text{or} \\ conv(\mathscr{A}) = conv(\mathscr{B}) \ \text{ and } \ div(\mathscr{A}) > div(\mathscr{B}) \end{array} \right. \\ \mathscr{A} = \mathscr{B} & \text{if} & conv(\mathscr{A}) = conv(\mathscr{B}) \ \text{ and } \ div(\mathscr{A}) = div(\mathscr{B}) \\ \mathscr{B} > \mathscr{A} & & \text{otherwise} \end{array} \tag{5.10}$$

In order to assign ranks to the list of approximations $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$, first we sort the approximations according to Formula 5.10. So, we obtain a list $L2 = (\mathscr{A}_{j1}, \mathscr{A}_{j2}, \ldots, \mathscr{A}_{jr})$ with the same elements of L but sorted. The algorithm to rank the approximations is shown in Figure 5.16.

Require: A list of r approximation sets $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$, a list of numbers $(conv(\mathscr{A}_1), conv(\mathscr{A}_2), \ldots, conv(\mathscr{A}_r))$ and a list of numbers $(div(\mathscr{A}_1), div(\mathscr{A}_2), \ldots, div(\mathscr{A}_r))$. SORT is an algorithm that sorts the elements of L according to Formula 5.10.
Ensure: A list of ranks for the elements of L.

1: $(\mathscr{A}_{j1}, \mathscr{A}_{j2}, \ldots, \mathscr{A}_{jr}) = SORT(L)$
2: $RANK = 1$
3: **for** $s = 1 : r - 1$ **do**
4: $\quad \mathscr{A}_{js} = RANK.$
5: $\quad$ **if** $conv(\mathscr{A}_{j(s+1)}) > conv(\mathscr{A}_{js})$ **then**
6: $\quad\quad RANK = RANK + 1$
7: $\quad$ **else if** $div(\mathscr{A}_{j(s+1)}) < div(\mathscr{A}_{js})$ **then**
8: $\quad\quad RANK = RANK + 1$
9: $\quad$ **end if**
10: **end for**
11: $\mathscr{A}_{jr} = RANK.$

Figure 5.16: Ranking Algorithm.

5.6.1 General Algorithm of the G–Ranker

Once we have explained all the components of the G–Ranker, the general algorithm is shown in Figure 5.17.

5.6.2 Properties of the G–Ranker

G–Ranker has some desirable properties. G–Ranker has the property of weak relativity. If the Pareto Front is included in the comparison, it always is in class C_1, and for any approximation $\mathscr{A}_i$ in C_1 its value of diversity will be evaluated considering only its elements in the Pareto Front. Because of the way that $B2$ is defined, the value of diversity of $\mathscr{A}_i$ will not be better than that of the Pareto Front.

Another property of G–Ranker is weak compatibility with O_W. To explain why G–Ranker is weakly compatible with O_W we introduce some notation. For two approximations $\mathscr{A}$ and $\mathscr{B}$, let the following sets be defined:

Require: A list of approximation sets $L = (\mathscr{A}_1, \mathscr{A}_2, \ldots, \mathscr{A}_r)$.
Ensure: The rank of the elements of L.

1: Normalize the elements of L according to algorithm in Figure 5.15.
2: Obtain the K classes C_k using the convergence algorithm in Figure 5.1
3: **for** k = 1 to K **do**
4: Calculate the value of diversity for the approximation sets in class C_k, using the diversity algorithm in Figure 5.14.
5: **end for**
6: Assign ranks to each approximation using algorithm in Figure 5.16.

Figure 5.17: G–Ranker.

J: is the set of indices of the vectors $a_j \in \mathscr{A}$ so that $a_j = b_r$ for some vector $b_r \in \mathscr{B}$.

K: is the set of indices of the vectors $a_k \in \mathscr{A}$ so that a_k is dominated by some vector $b \in \mathscr{B}$.

L: is the set of indices of the vectors $a_l \in \mathscr{A}_i$ so that a_l is not in $\mathscr{B}$ and a_l is not dominated by any vector in $\mathscr{B}$.

R: is the set of indices of the vectors $b_r \in \mathscr{B}$ so that $b_r = a_j$ for some $a_j \in \mathscr{A}$.

S: is the set of indices of the vectors $b_s \in \mathscr{B}$ so that b_s is dominated by some $a \in \mathscr{A}$.

T: is the set of indices of the vectors $b_t \in \mathscr{B}$ so that b_t is not in $\mathscr{A}$ and b_t is not dominated by any vector in $\mathscr{A}$.

As an example, for $\mathscr{A}$ and $\mathscr{B}$ in Figure 5.18, $J = \{3, 4\}$, because a3 = b4 and a4 = b5, $K = \{5\}$ because a5 is dominated by elements of B, $L = \{1, 2\}$ because a1 and a2 are elements of $\mathscr{A}$ not present in $\mathscr{B}$ and they are not dominated by elements of $\mathscr{B}$. Similarly $R = \{4, 5\}$, $S = \{1, 2, 3\}$ and $T = \{6, 7\}$. Note the J, K and L can be used to make a partition of $\mathscr{A}$, and R, S and T can be used to make a partition of $\mathscr{B}$. Also note that the set of vectors whose indices are in J is the same set of vectors whose indices are in R.

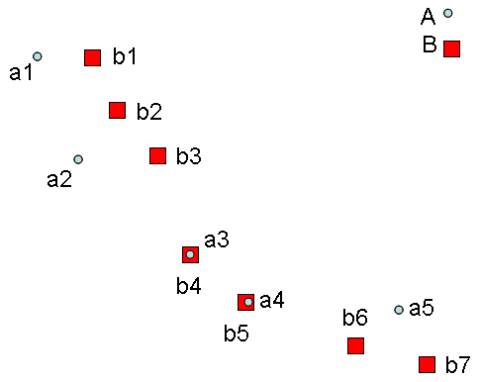

Figure 5.18: Two non-dominated sets.

Denote by $b(r, x)$ the ball of radius r and center x, and considering our diversity algorithm, we have that, $div(\mathscr{A}) = \mu\left(\bigcup_{j\in J} b(U, a_j) \cup \bigcup_{l\in L} b(U, a_l)\right)$. Similarly, $div(\mathscr{B}) = \mu\left(\bigcup_{r\in R} b(U, b_r) \cup \bigcup_{t\in T} b(U, b_t)\right)$. In the evaluation of diversity, we ignore the vectors whose indices are in K and S, because in our diversity algorithm we only consider vectors not dominated by other vectors in the same class.

Theorem 1. *When comparing two NSs, $\mathscr{A}$ and $\mathscr{B}$, G–Ranker is weakly compatible with O_C, O_S and O_W.*

Proof. If $\mathscr{A}$ O_C $\mathscr{B}$, this implies that $\mathscr{A}$ is in a better class than $\mathscr{B}$ because $\mathscr{B}$ cannot have elements in the same "relative" Pareto Front as $\mathscr{A}$, but only in the "relative" Pareto Front of a worse class. Hence, based on the rule given in Formula 5.10, $conv(\mathscr{A}) < conv(\mathscr{B})$ implies that $\mathscr{A}$ is better than $\mathscr{B}$.

If $\mathscr{A}$ O_S $\mathscr{B}$ and $\mathscr{A}$ does not completely outperform $\mathscr{B}$, then both sets are in the same class of convergence. Also, T is an empty set and L contains

at least one element, so:

$$\mu\left(\bigcup_{j\in J} b(U,a_j) \ \cup \ \bigcup_{l\in L} b(U,a_l)\right) \ \geq \ \mu\left(\bigcup_{r\in R} b(U,b_r)\right) \tag{5.11}$$

$$div(\mathscr{A}) \ \geq \ div(\mathscr{B}) \tag{5.12}$$

Inequality 5.11 holds because, by definition, the set of vectors whose indices are in J is equal to the set of vectors whose indices are in R. As a consequence $\bigcup_{j\in J} b(U,a_j) = \bigcup_{r\in R} b(U,b_r)$, $\mu\left(\bigcup_{j\in J} b(U,a_j)\right) = \mu\left(\bigcup_{r\in R} b(U,b_r)\right)$ and $\mu\left(\bigcup_{j\in J} b(U,a_j) \ \cup \ \bigcup_{l\in L} b(U,a_l)\right) \geq \mu\left(\bigcup_{r\in R} b(U,b_r)\right)$. According to the rule given in Formula 5.10, if $conv(\mathscr{A}) = conv(\mathscr{B})$ and $div(\mathscr{A}) > div(\mathscr{B})$ then $\mathscr{A}$ is better than $\mathscr{B}$. Also, if $conv(\mathscr{A}) = conv(\mathscr{B})$ and $div(\mathscr{A}) = div(\mathscr{B})$ then $\mathscr{A}$ is as good as $\mathscr{B}$. So $\mathscr{A}$ is never considered as inferior than $\mathscr{B}$.

If $\mathscr{A}$ O_W $\mathscr{B}$ and $\mathscr{A}$ does not strongly outperform $\mathscr{B}$, then both sets are in the same class of convergence, T is empty and L contains at least one element, so this case is equal to the previous one.

□

This proves that G–Ranker is weakly compatible with the out–performance relations when it evaluates two NSs. With respect to more than two NSs, the G–Ranker is compatible with O_C, because of the mechanism of the convergence operator. With respect to O_S and O_W, G–Ranker is weakly compatible. Suppose that there are three NSs $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$ in the same class and that $\mathscr{A}$ O_S $\mathscr{B}$ or $\mathscr{A}$ O_W $\mathscr{B}$. If we only consider $\mathscr{A}$ and $\mathscr{B}$, this means that T is empty and L has at least one element, so $div(\mathscr{A}) > div(\mathscr{B})$. But when we consider $\mathscr{C}$, it is possible for all a_ls to be dominated by some elements in $\mathscr{C}$, so the elements in L are discarded and $div(\mathscr{A}) = div(\mathscr{B})$. It does not matter how many NSs we add to the comparison, if an element of $\mathscr{A}$ whose index is in J is dominated, the corresponding element in $\mathscr{B}$ whose index is in R is also dominated and the equality holds. As a consequence $div(\mathscr{A})$ will not be smaller than $div(\mathscr{B})$, so the weak compatibility with O_S and O_W is

guaranteed.

According to Knowles and Corne [42], weak compatibility with O_W is enough to exhibit weak monotony, so we have that G–Ranker has weak monotony.

Chapter 6

Experiments

6.1 Introduction

The evaluation of the behavior of a quality indicator has been mainly based on theoretical studies of its properties, but not on a methodological testing. Theoretical studies are fundamental to validate a quality indicator, but the use of a benchmark, in our opinion, is very important also. Some quality indicators that work well in theory may have a poor behavior in practice. There are several benchmarks in the literature for MOEAs [16], [14], but there is only one benchmark for performance measures [54] according to our knowledge.

So, in order to evaluate the performance of our approach, we design some experiments that consider various possible topologies of the Pareto Front. We compare our results with those of the S-Metric [81], the I_ϵ [83], and the Compatible and Complete Comparison Method (CCCM) [83]. All these methods are described in Section 3. We choose these quality indicators because they are very popular in the literature and, like the G–Ranker, they evaluate the general quality of non–dominated sets.

We design our test cases so it is evident which NS is better than the others, or if there are NSs that are equally good. So, it is possible to decide

the correct order from the best approximation to the worst approximation. The challenge for the quality indicators is to evaluate the NSs in such a way that we can find the correct order.

The reference point for the S–metric is the maximum value of all vectors in all non–dominated sets in the corresponding test case. The reference set for the unary ϵ–indicator is the non–dominated vectors from the union of all the sets in the test case.

The evaluations of the S–metric determine an order in the approximations, we only need to sort the approximations based on their values of the S–metric, from the greatest to the smallest. The same is also valid for the I_ϵ. For the case of the G–Ranker, we only need to sort the sets based on their ranks, from the smallest rank to the greatest rank.

For the CCCM, more work is necessary in order to find an order between M approximations. The CCCM is a binary comparison method, so we need to make $M(M-1)/2$ binary comparisons in order to compare all sets. From these comparisons we figure out an order for the approximations.

6.2 Nomenclature

In order to simplify the analysis of the results, we introduce the following nomenclature to represent the different relationships between different non–dominated sets. Suppose we have five NSs $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$ and $\mathscr{E}$. $\mathscr{A} > \mathscr{B}$ means that $\mathscr{A}$ is better than $\mathscr{B}$, $\mathscr{A} = \mathscr{B}$ means that $\mathscr{A}$ and $\mathscr{B}$ are equally good. $\mathscr{A}$? $\mathscr{B}$ means that it is unknown whether $\mathscr{A}$ is better than $\mathscr{B}$. $\mathscr{A} > \mathscr{B}, \mathscr{C}, \mathscr{D}$ means that $\mathscr{A}$ is better than any of $\mathscr{B}$, $\mathscr{C}$ and $\mathscr{D}$; and $\mathscr{A}, \mathscr{B}, \mathscr{C} > \mathscr{E}$ means that $\mathscr{E}$ is worse than any of $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{C}$.

We concatenate strings of symbols to create more complex expressions. For example:

- $\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$ means that $\mathscr{E}$ is better than any of the other sets; $\mathscr{D}$ is better than $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{C}$; $\mathscr{C}$ is better than $\mathscr{A}$ and $\mathscr{B}$; and

Table 6.1: Results of Experiment 1

	2d	3d
G–Ranker	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
S–metric	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
I_ϵ	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
$CCCM$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$

$\mathscr{B}$ is better than $\mathscr{A}$.

- $\mathscr{E} > \mathscr{D} = \mathscr{C} > \mathscr{B} = \mathscr{A}$ means that $\mathscr{E}$ is better than any other set; $\mathscr{D}$ and $\mathscr{C}$ are equally good and better than both $\mathscr{A}$ and $\mathscr{B}$; and $\mathscr{B}$ and $\mathscr{A}$ are equally good.
- $\mathscr{E} = \mathscr{D} = \mathscr{C} = \mathscr{B} > \mathscr{A}$ means that $\mathscr{D}$, $\mathscr{B}$, $\mathscr{C}$ and $\mathscr{D}$ are equally good and that $\mathscr{A}$ is worse than any of the other NSs.

For each test case we make a two dimensional version and a three dimensional version. We consider all the test cases as minimization problems.

6.3 Experiments

6.3.1 Experiment 1

The goal of this experiment is to evaluate the ability of the algorithms to detect various levels of complete out–performance. This experiment is clearly focused on convergence. There are five NSs, $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$ and $\mathscr{E}$, with the same diversity. $\mathscr{A}$ O_C $\mathscr{B}$, $\mathscr{B}$ O_C $\mathscr{C}$, $\mathscr{C}$ O_C $\mathscr{D}$ and $\mathscr{D}$ O_C $\mathscr{E}$. So, from the best to the worst, the order is $\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$. The two dimensional version of the NSs is shown in Figure 6.1.

The results for this experiment can be seen in Table 6.1. All the methods correctly evaluate the approximations.

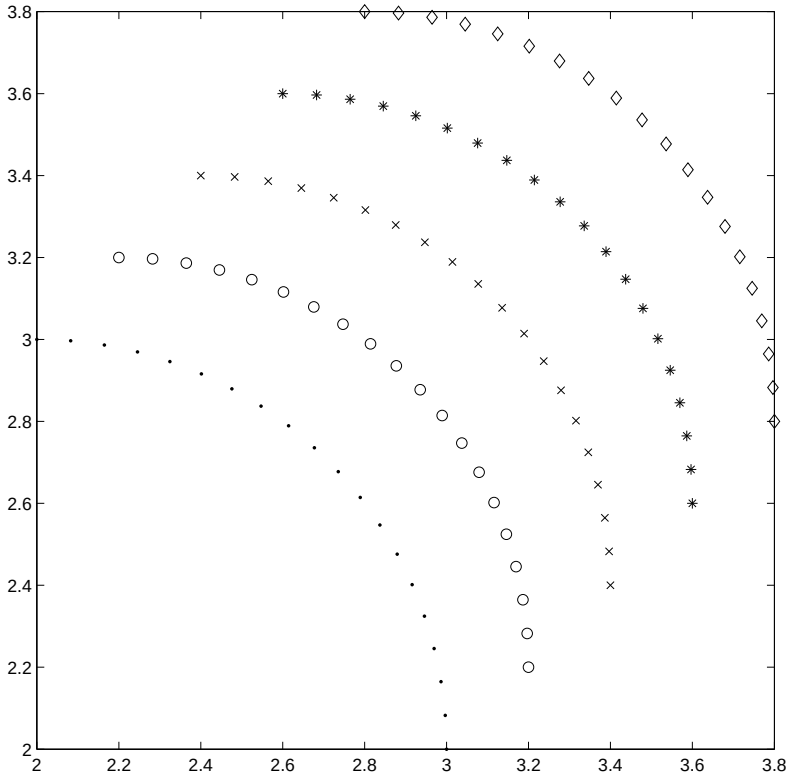

Figure 6.1: Experiment 1 in two dimensions. $\mathscr{A}(\cdot), \mathscr{B}(\circ), \mathscr{C}(\times), \mathscr{D}(*), \mathscr{E}(\Diamond)$.

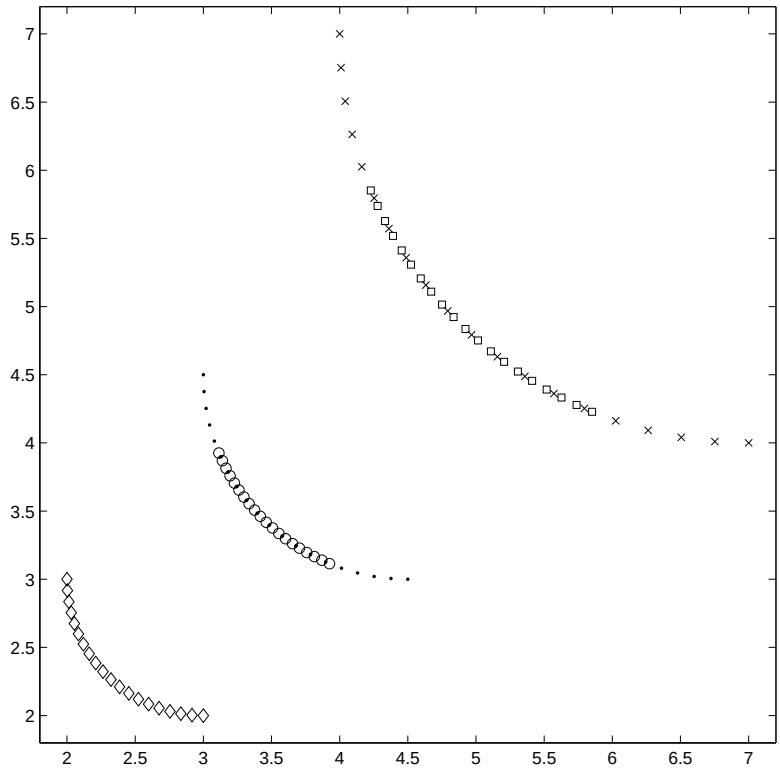

Figure 6.2: Experiment 2 in two dimensions. $\mathscr{A}(\Diamond), \mathscr{B}(\circ), \mathscr{C}(\cdot), \mathscr{D}(\square), \mathscr{E}(\times)$.

6.3.2 Experiment 2

In this experiment (Figure 6.2) we combine convergence with diversity. There are five NSs $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$ and $\mathscr{E}$, where $\mathscr{A}$ completely outperforms the other sets in the experiment. Also, both $\mathscr{B}$ and $\mathscr{C}$ completely outperform both $\mathscr{D}$ and $\mathscr{E}$. $\mathscr{B}$ and $\mathscr{C}$ have the same convergence, but $\mathscr{C}$ has a better diversity than $\mathscr{B}$. $\mathscr{D}$ and $\mathscr{E}$ have the same convergence, but $\mathscr{E}$ has a better diversity than $\mathscr{D}$. So, from the best to the worst, the correct order is $\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{E} > \mathscr{D}$.

Table 6.2: Results of Experiment 2

	2d	3d
G–Ranker	$\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{E} > \mathscr{D}$	$\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{E} > \mathscr{D}$
S–metric	$\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{E} > \mathscr{D}$	$\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{E} > \mathscr{D}$
I_ϵ	$\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{C} > \mathscr{B} > \mathscr{D} > \mathscr{E}$
$CCCM$	$\mathscr{A} > \mathscr{B} \ ? \ \mathscr{C} > \mathscr{E} \ ? \ \mathscr{D}$	$\mathscr{A} > \mathscr{B} \ ? \ \mathscr{C} > \mathscr{E} \ ? \ \mathscr{D}$

Table 6.3: Results of Experiment 3

	2d	3d
G–Ranker	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
S–metric	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
I_ϵ	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
$CCCM$	$\mathscr{A} \ ? \ \mathscr{B} \ ? \ \mathscr{C} \ ? \ \mathscr{D} \ ? \ \mathscr{E}$	$\mathscr{A} \ ? \ \mathscr{B} \ ? \ \mathscr{C} \ ? \ \mathscr{D} \ ? \ \mathscr{E}$

Both G–Ranker and S–Metric find the correct order of the sets, as can be seen in Table 6.2. But $CCCM$ is not able to detect the difference between $\mathscr{B}$ and $\mathscr{C}$, and between $\mathscr{D}$ and $\mathscr{E}$. I_ϵ incorrectly evaluates $\mathscr{D}$ as better than $\mathscr{E}$ in both two dimensions and three dimensions.

6.3.3 Experiment 3

This experiment tests if the performance measures can detect empty zones in the non–dominated sets. We have five NSs with circular shape and with a hole in the center (Figure 6.3). The size of the hole is different for different NSs. We consider that the bigger the hole, the worse the NS. So, from the best to the worst, the order is $\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$.

The result of the experiment is shown in Table 6.3. G–Ranker, I_ϵ and S–Metric find the correct order. CCCM is not able to detect any difference in the quality of the sets.

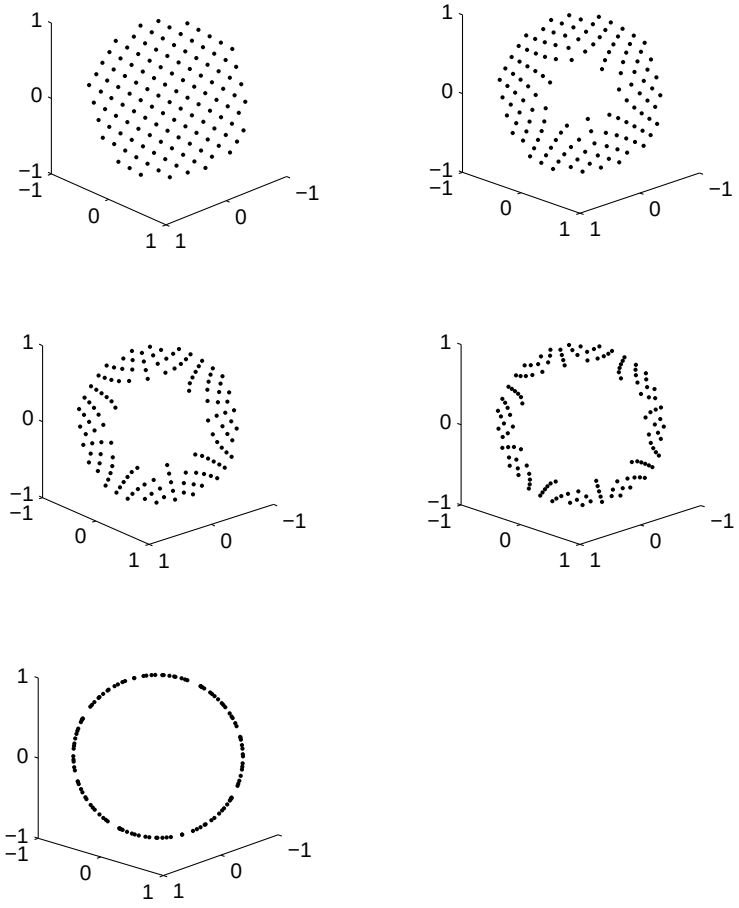

Figure 6.3: Experiment 3 in three dimensions. $\mathscr{A}$ (top, left side), $\mathscr{B}$ (top, right side), $\mathscr{C}$ (center, left side), $\mathscr{D}$ (center, right side), $\mathscr{E}$ (bottom).

Table 6.4: Results of Experiment 4

	2d	3d
G–Ranker	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
S–metric	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$
I_ϵ	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$	$\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{E} > \mathscr{D}$
$CCCM$	$\mathscr{A}\ ?\ \mathscr{B}\ ?\ \mathscr{C}\ ?\ \mathscr{D}\ ?\ \mathscr{E}$	$\mathscr{A}\ ?\ \mathscr{B}\ ?\ \mathscr{C}\ ?\ \mathscr{D}\ ?\ \mathscr{E}$

6.3.4 Experiment 4

This case is inspired by problem DTLZ1 [16]. In this problem, the Pareto Front consists of all the points $p \in \mathbf{R}^d$ with components $p^{(k)} \geq 0$ for $k = 1$ to d and $\sum_{i=1}^{d} p^{(i)} = 0.5$. There are five NSs, where $\mathscr{A}$ has the best diversity. The other NSs are obtained adding different levels of noise to the positions of the points in $\mathscr{A}$. We consider that the bigger the noise in the NS, the worse its diversity (see Figure 6.4). The convergence for all NSs is the same. From the best to the worst, the order is $\mathscr{A} > \mathscr{B} > \mathscr{C} > \mathscr{D} > \mathscr{E}$.

The result of the experiment is shown in Table 6.4. G–Ranker and S–Metric find the correct order. I_ϵ incorrectly considers $\mathscr{E}$ better than $\mathscr{D}$ in the 3d version of the test. CCCM is not able to detect any difference in the quality of the sets.

6.3.5 Experiment 5

In this experiment (Figure 6.5), we evaluate the ability of the algorithms to detect weak out–performance. We create five sets $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$, $\mathscr{D}$ and $\mathscr{E}$ where $\mathscr{E}\ O_W\ \mathscr{D}$, $\mathscr{D}\ O_W\ \mathscr{C}$, $\mathscr{C}\ O_W\ \mathscr{B}$ and $\mathscr{B}\ O_W\ \mathscr{A}$. So, from the best to the worst the correct order is $\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$.

Considering that all the methods are compatible with O_W (G-Ranker is weakly compatible), it is not surprising that all of them find the correct order of the sets (see Table 6.5).

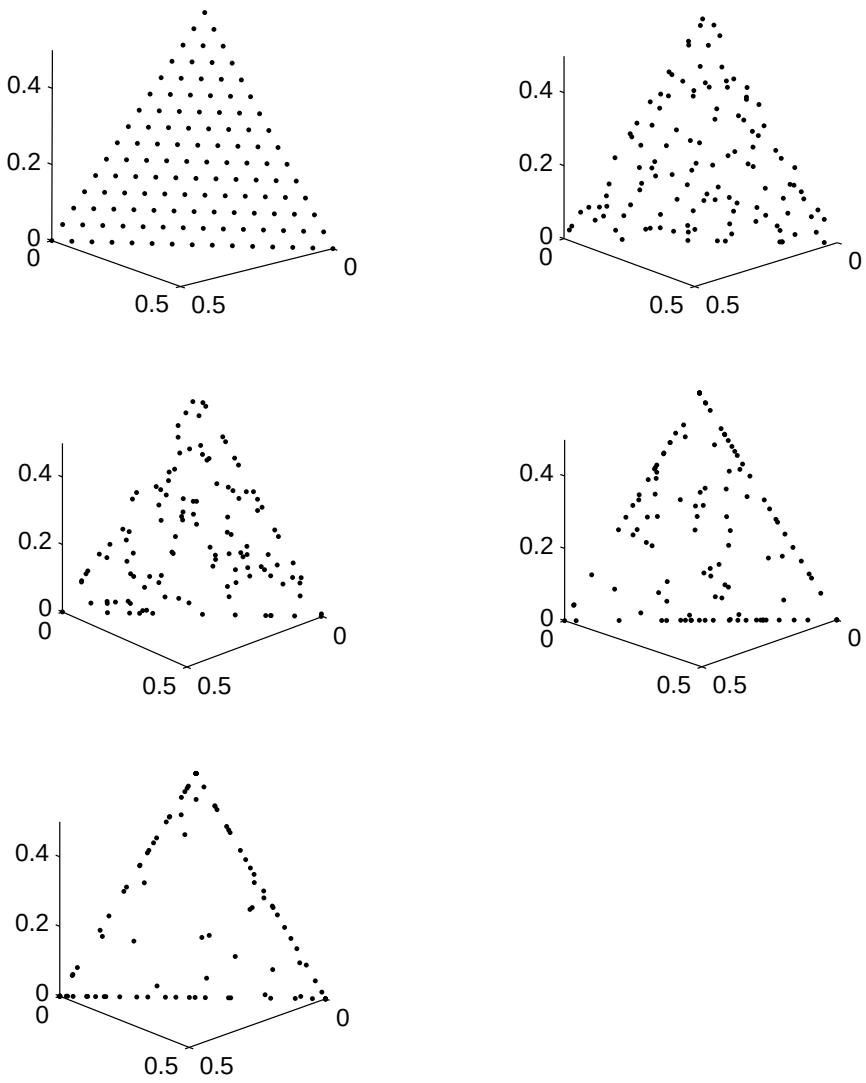

Figure 6.4: Experiment 4 in three dimensions. $\mathscr{A}$ (top, left side), $\mathscr{B}$ (top, right side), $\mathscr{C}$ (center, left side), $\mathscr{D}$ (center, right side), $\mathscr{E}$ (bottom).

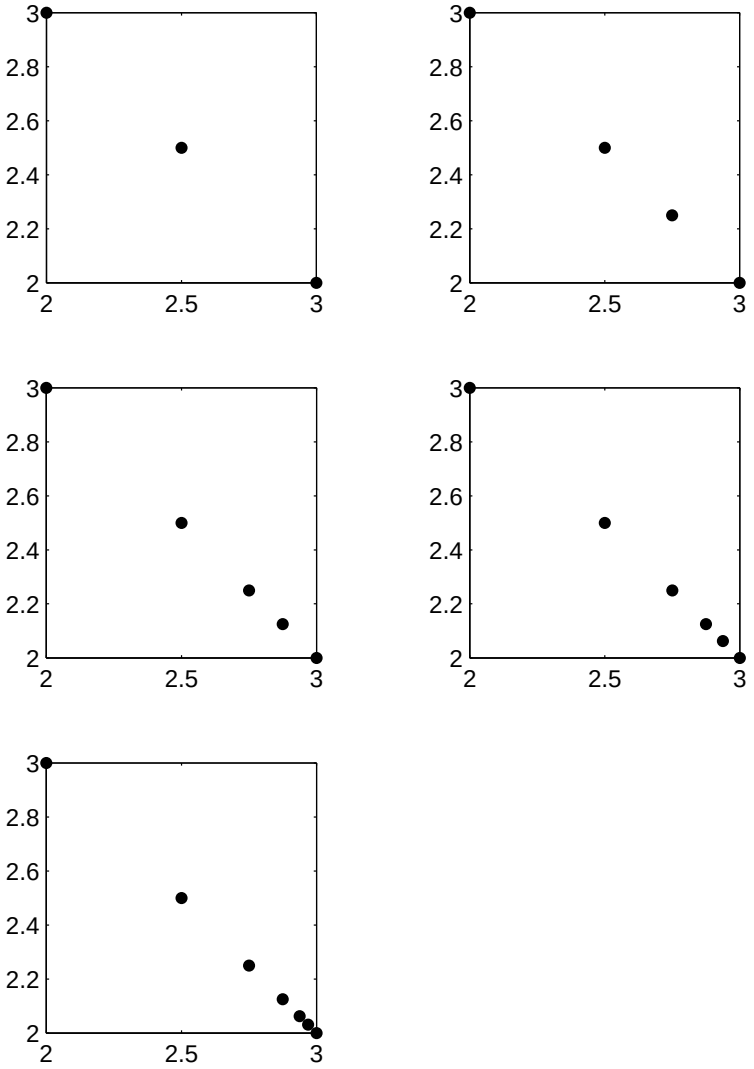

Figure 6.5: Experiment 5 in two dimensions. $\mathscr{A}$ (top, left side), $\mathscr{B}$ (top, right side), $\mathscr{C}$ (center, left side), $\mathscr{D}$ (center, right side), $\mathscr{E}$ (bottom).

Table 6.5: Results of Experiment 5

	2d	3d
G–Ranker	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$
S–metric	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$
I_ϵ	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$
$CCCM$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$

Table 6.6: Results of Experiment 6

	2d	3d
G–Ranker	$\mathscr{A} = \mathscr{B}$	$\mathscr{A} = \mathscr{B}$
S–metric	$\mathscr{B} > \mathscr{A}$	$\mathscr{B} > \mathscr{A}$
I_ϵ	$\mathscr{B} > \mathscr{A}$	$\mathscr{B} > \mathscr{A}$
$CCCM$	$\mathscr{A}$? $\mathscr{B}$	$\mathscr{A}$? $\mathscr{B}$

6.3.6 Experiment 6

The goal of this experiment is to compare the sensibility of the measures to the convexity of the Pareto Front. We have two NSs, $\mathscr{A}$ and $\mathscr{B}$, with the same diversity and convergence, but $\mathscr{A}$ is non-convex while $\mathscr{B}$ is convex (Figure 6.6). We expect the same evaluation for both NSs.

It seems, according to Table 6.6, that both S–Metric and I_ϵ have a bias towards the convex zones of the Pareto Front. The G-Metric gives the same rank to both NSs, as expected. The $CCCM$ evaluates the sets as non–comparable.

6.3.7 Experiment 7

To evaluate the sensibility of the performance measures to the relative position of the different NSs, we create five NSs with the same convergence and diversity but with different positions. An image of the two dimensional version is shown in Figure 6.7.

As is clear from Table 6.7, both S–metric and I_ϵ give a different evaluation

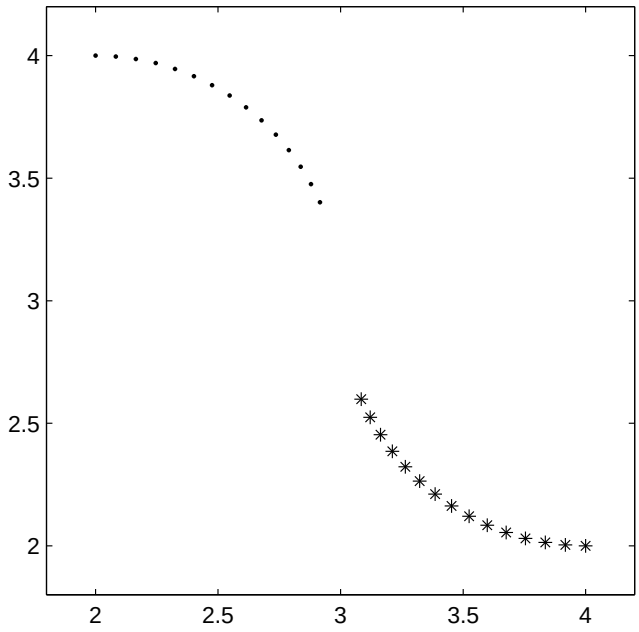

Figure 6.6: Experiment 6 in two dimensions. $\mathscr{A}(\cdot), \mathscr{B}(*)$.

Table 6.7: Results of Experiment 7

	2d	3d
G–Ranker	$\mathscr{A} = \mathscr{B} = \mathscr{C} = \mathscr{D} = \mathscr{E}$	$\mathscr{A} = \mathscr{B} = \mathscr{C} = \mathscr{D} = \mathscr{E}$
S–metric	$\mathscr{C} > \mathscr{B} = \mathscr{D} > \mathscr{A} = \mathscr{E}$	$\mathscr{B} > \mathscr{C} > \mathscr{A} > \mathscr{D} > \mathscr{E}$
I_ϵ	$\mathscr{C} > \mathscr{B} = \mathscr{D} > \mathscr{A} = \mathscr{E}$	$\mathscr{C} > \mathscr{B} = \mathscr{D} > \mathscr{A} = \mathscr{E}$
$CCCM$	$\mathscr{A}\ ?\ \mathscr{B}\ ?\ \mathscr{C}\ ?\ \mathscr{D}\ ?\ \mathscr{E}$	$\mathscr{A}\ ?\ \mathscr{B}\ ?\ \mathscr{C}\ ?\ \mathscr{D}\ ?\ \mathscr{E}$

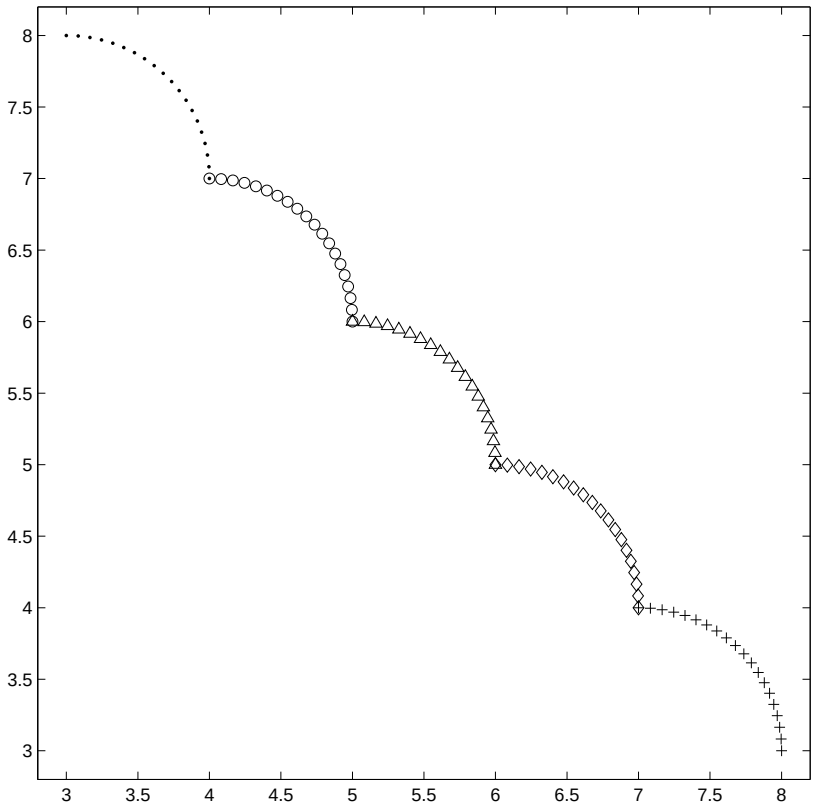

Figure 6.7: Experiment 7 in two dimensions. $\mathscr{A}(\cdot), \mathscr{B}(\circ), \mathscr{C}(\triangle), \mathscr{D}(\Diamond), \mathscr{E}(+)$.

Table 6.8: Results of Experiment 8

	2d	3d
G–Ranker	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$
S–metric	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$
I_ϵ	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$	$\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$
$CCCM$	$\mathscr{A}\ ?\ \mathscr{B}\ ?\ \mathscr{C}\ ?\ \mathscr{D}\ ?\ \mathscr{E}$	$\mathscr{A}\ ?\ \mathscr{B}\ ?\ \mathscr{C}\ ?\ \mathscr{D}\ ?\ \mathscr{E}$

to the NSs, so they fail the test. CCCM is not able to make conclusions about the NSs. G–Ranker evaluates the NSs as equally good, so it is the only method that passes the test.

6.3.8 Experiment 8

For this experiment (Figure 6.8), we generate five NSs with the same extension and convergence. All vectors are evenly distributed, but the sets have a different number of vectors. We have that $|\mathscr{A}| = 4$, $|\mathscr{B}| = 12$, $|\mathscr{C}| = 38$, $|\mathscr{D}| = 98$, $|\mathscr{E}| = 758$ for the two dimensional case. For the three dimensional case, $|\mathscr{A}| = 10$, $|\mathscr{B}| = 21$, $|\mathscr{C}| = 35$, $|\mathscr{D}| = 78$, $|\mathscr{E}| = 465$. The objective is to study how the number of vectors can affect the behavior of the quality indicators. The expected ordering is $\mathscr{E} > \mathscr{D} > \mathscr{C} > \mathscr{B} > \mathscr{A}$.

All methods are able to find the expected ordering for this experiment (Table 6.8) except for CCCM that is not able to detect any difference between the approximations.

6.4 Conclusions of Experiments

The G–Ranker shows the best performance of the comparison methods analyzed, correctly solving all test cases. The S–Metric is the second best of the comparison, being sensitive to the relative position and convexity of the non–dominated sets. As the S–Metric, I_ϵ is sensitive to the relative position

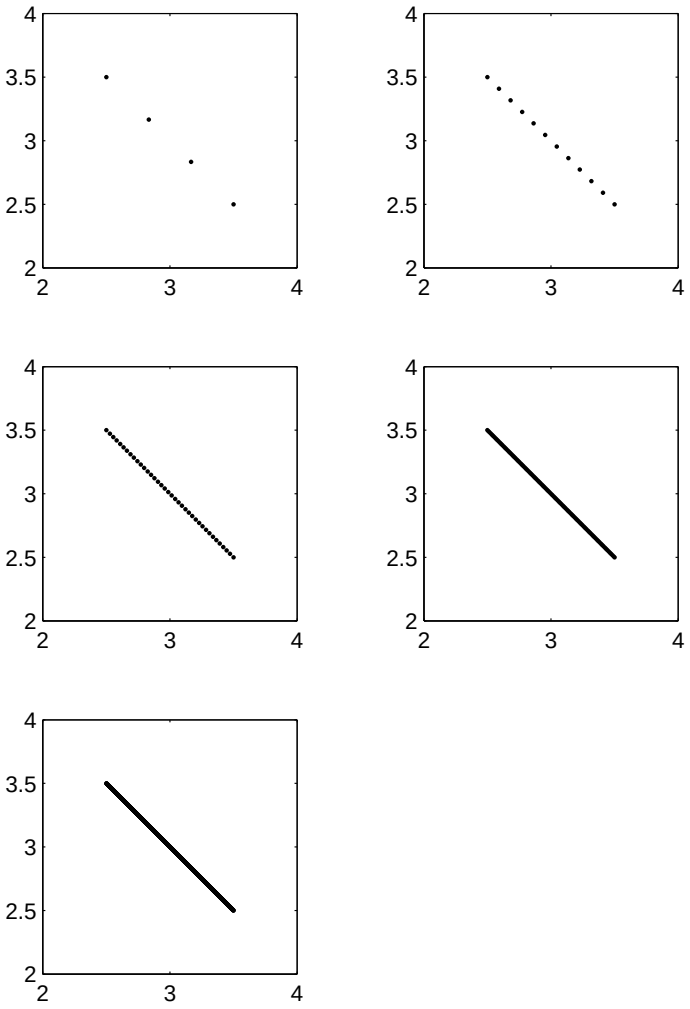

Figure 6.8: Experiment 8 in two dimensions. $\mathscr{A}$ (top, left side), $\mathscr{B}$ (top, right side), $\mathscr{C}$ (center, left side), $\mathscr{D}$ (center, right side), $\mathscr{E}$ (bottom).

and the convexity of the non–dominated sets, but it also fails Experiments 2 and 4. The CCCM presents a very limited inference power. It never makes a wrong statement about when one approximation is better that another, but neither is it able detect important differences in the quality of non–dominated sets.

Chapter 7

Further Details

7.1 Introduction

In this chapter we briefly discuss some details and future work related with the research made in this work.

7.2 The G–Indicator

The G–Ranker assigns a rank to each non–dominated set in order to make a non–parametric comparison of two MOEAs. Ranking a collection of approximations tells us whether one approximation is better than another, but not by how much. It could be desirable to redefine the G–Ranker so it assigns a real number to each approximation so we can make other kinds of analyses. The diversity operator of the G–Ranker already assigns real numbers to the approximations that are proportional to their diversity. We can transform the convergence component into a real number, so we can combine this real number with the diversity evaluation to get a single number that represents the relative quality of an approximation with respect to the others.

An example of how this could be done, is the following. First, we run the algorithm of the G–Ranker shown in Figure 5.17 to obtain K classes C_k

and the evaluation of diversity, $div(\mathscr{A})$, for each approximation $\mathscr{A}$. Next, we measure the distance between the approximations in class C_k and the approximations in class C_{k+1}, in other words, we measure the distance between every pair of contiguous classes of dominance. Calculating the distance between two collections of sets can be done in many ways. We propose the following idea: in order to calculate a distance between classes C_k and C_{k+1} we use the "relative" Pareto Front of each class, $\mathscr{R}_k$ and $\mathscr{R}_{k+1}$.

As an illustrative example, recall the example in Figure 5.2, where we have five approximations and three classes of dominance. In Figure 7.1, we see the three relative Pareto Fronts for the example in Figure 5.2. We take as the distance between two classes C_k and C_{k+1}, the distance between the relative Pareto Fronts $\mathscr{R}_k$ and $\mathscr{R}_{k+1}$. We can define many ways to measure the distance between relative Pareto Fronts, for example, the maximum of the distances between an element of $\mathscr{R}_k$ and the nearest element of $\mathscr{R}_{k+1}$. Another example is to use the Generational Distance between $\mathscr{R}_k$ and $\mathscr{R}_{k+1}$. Once a measure of distance has been chosen, calculate the distance between the $\mathscr{R}_k$s. Denote by $D(k, k+1)$ the distance between C_k and C_{k+1}.

Now, we combine the distances $D(k, k+1)$ with the evaluation of diversity $div(\mathscr{A})$ to generate a unique figure of merit. Let $G(\mathscr{A})$ denotes this figure of merit. For the approximations $\mathscr{A}$ in the worst class C_K, we set $G(\mathscr{A}) = div(\mathscr{A})$. For the approximations $\mathscr{A}_j$ in the second worst class C_{K-1}, we set $G(\mathscr{A}_j) = div(\mathscr{A}_j) + D(K-1, K) + \overset{max}{i \in C_K} G(\mathscr{A}_i)$. For the sets $\mathscr{A}_j$ in class C_k, $k < K$, we set the value $G(\mathscr{A}_j) = div(\mathscr{A}_j) + D(k, k+1) + \overset{max}{i \in C_{k+1}} G(\mathscr{A}_i)$.

The definition of $G(\mathscr{A})$ given above, guarantees that $G(\mathscr{A}_i) > G(\mathscr{A}_j)$ when $\mathscr{A}_i$ is in a better class than $\mathscr{A}_j$. If we rank the approximation based on the value of $G(\mathscr{A})$, we obtain the same ranking as that of the G–Ranker.

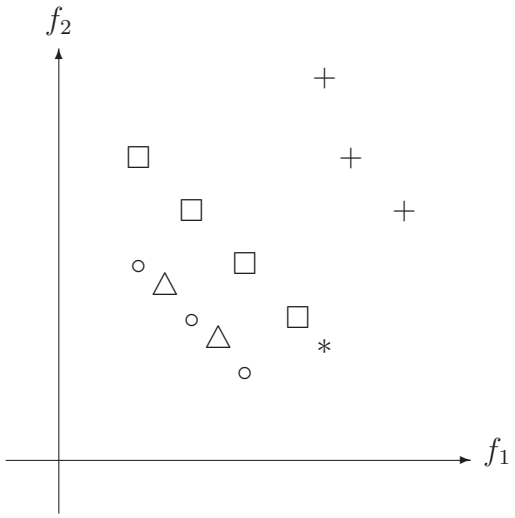

Figure 7.1: Three relative Pareto Fronts. $\mathscr{R}_1$ $(\triangle, \circ)$, $\mathscr{R}_2$ $(\square, *)$, $\mathscr{R}_3$ $(+)$.

7.3 Misleading Cases

The G–Ranker is not free of misleading cases. Consider the non–dominated sets in Figure 7.2 and suppose that the Pareto Front is the union of $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$ and $\mathscr{D}$. All the elements of $\mathscr{E}$ are dominated by the elements of the other approximations, thus the G–Ranker will put $\mathscr{E}$ in the second class and it will be considered as the worst approximation. But, according to intuition $\mathscr{E}$ could be the best approximation, because it is "almost" in the Pareto Front and has better diversity than the other sets. A DM may have a better idea of the possible attainable solutions from $\mathscr{E}$ than from any of the other sets. This is an atypical case where diversity may have more or equal relevance than convergence. G–Ranker considers $\mathscr{E}$ as the worst set because all its elements are dominated, so all the information in $\mathscr{E}$ is not optimal. But as a whole and considering the poor diversity of the other sets, $\mathscr{E}$ may be more informative than any of the other approximations.

A possible solution to this problem is to use a relaxed version of dom-

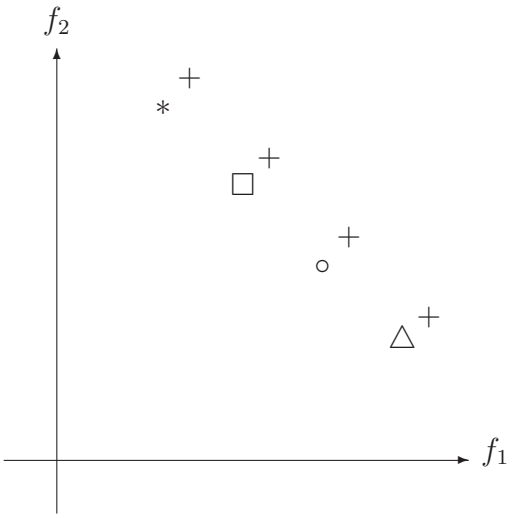

Figure 7.2: A misleading case for the G–Ranker. $\mathscr{A}$ ($\triangle$), $\mathscr{B}$ ($\circ$), $\mathscr{C}$ ($\square$), $\mathscr{D}$ ($*$), $\mathscr{E}$ ($+$).

inance, the negative ϵ–dominance. The negative ϵ–dominance is defined as follows:

Definition 35. *A vector $a \in \boldsymbol{R}^n$ dominates a vector $b \in \boldsymbol{R}^n$ according to the negative ϵ–dominance and a real value $\epsilon > 0$, if $a^{(i)} + \epsilon \leq b^{(i)}$, for all $i \in \{1, \ldots, n\}$ and there exists an index $j \in \{1, \ldots, n\}$ such that $a^{(j)} + \epsilon < b^{(j)}$.*

The negative ϵ–dominance considers a tolerance ϵ in the definition of dominance, so it is harder for a vector to dominate another one. As a consequence, some approximations with vectors that are nearly optimal can be included in better classes of dominance. The disadvantage is that we add a parameter to the G–Ranker that needs to be adjusted. Fortunately, the parameter ϵ has an intuitive meaning and the effects of increasing or decreasing this parameter are not hard to visualize.

7.4 Using Information Related to the Optimization Problem

The G–Ranker is designed to be used when there is no information available about the multi–objective problem. But, if we have information of the multi–objective problem, we can consider it when using the G–Ranker. For example, if the extreme values of the Pareto Front are known, we can make the normalization of the non–dominated sets using these values.

7.5 Using Other Geometric Objects

In Section 5, we explain how we decide to associate balls to the elements of a non–dominated set in order to evaluate its diversity. It is interesting to explore the use of other geometric figures. For example, instead of balls we may use boxes whose geometric center is located in the elements of the approximation sets and whose faces are aligned with the axis of the coordinate system. The advantage of using boxes is that there exists an exact algorithm to calculate the volume of the union of several boxes in any dimension [55], so we do not need to make an approximation with a Monte Carlo integration. The disadvantage is that using boxes makes the G–Ranker more sensitive to the orientation of the approximation sets.

Chapter 8

Conclusions

In this work we create a model of the analyst preferences for approximation sets. For some years, the experts in multi–objective optimization have had an intuitive idea of what characteristics are desirable in an approximation to the Pareto Front. Much effort has been invested in constructing a way to transform those characteristics into real numbers, or at least to create an order between the approximation sets according to their quality.

Our model has many properties that have been considered as desirable in the literature, like weak monotony, weak relativity and weak compatibility with the out–performance relations. The most important property of our approach is that it models very accurately the analyst preferences, as is shown in the benchmark we used to test it. It performs better than other comparison methods that are very popular in the literature.

Another contribution of this work is the introduction of a benchmark for quality indicators. The intention of these test cases is to check if a quality indicator is a good model or if it can be mislead. The study of quality indicators has been based, mainly, on theoretical aspects. But it is possible that methods that work well in theory, have a poor performance in practice. The importance of the test cases is that they allow one to study the behavior of the quality indicators in a more practical way.

We test some popular quality indicators using our benchmark, obtaining very interesting results. The S–metric proves to be one of the best performance measures, solving correctly most of the test cases. It fails the test cases related with the convexity of the approximations and the relative position of the approximations, confirming that the S–metric has some biases.

The ϵ–indicator is a relatively new quality indicator that is very popular in the literature. It is very easy to understand and to implement, and very efficient in computational complexity. But these advantages have a price. The behavior of the *epsilon*–indicator is below that of the S–metric. It shows the same biases as the S–metric and fails some other test cases. However, it correctly solve many of the test cases. Considering how fast this quality indicator is, it may be desirable to use the ϵ–indicator for fast tests, where we want to have an idea of the quality of the approximation and we do not need much precision.

The biggest surprise is related to the compatible and complete comparison method (CCCM). This comparison method has the most desirable theoretical properties, but when we test it in our benchmark, the results show that the CCCM has a very limited inference power. It solves correctly the test cases where there is a big difference in convergence, but gives no information when the approximations have similar convergence, even when they have huge differences in diversity. A extreme case is Test Case 8, where the approximations are mutually non comparable under the out–performance relations but they have big differences in the number of vectors. For example, in the two dimensional version of that test case, $|\mathscr{A}| = 4$ and $|\mathscr{E}| = 758$, but the CCCM considers these approximations as non comparable! This is no surprise, because when the CCCM was designed, the concept of diversity was not considered. The CCCM designed is based only on the concept of dominance between vectors, so its behavior in the benchmark is normal and it does not represent a defect in its design. The problem is that many researchers are not aware of this, and it is frequently assumed that the CCCM

can evaluate both diversity and convergence.

It is interesting to see that the comparison method with the most theoretical properties is the one with the most limited practical use, while the other quality indicators, even with less theoretical properties, have better behavior. This shows how important it is to use test cases to validate quality indicators. Another conclusion is that the most complicated and expensive quality indicators tested, are the most effective ones. This suggests that in order to evaluate the quality of an approximation in a more accurate way, one needs to pay the price in complexity. This is what we could expect by intuition, because the quality of an approximation is a very complex property to evaluate, and a quality indicator based on a very simple computation is not expected to capture properly that property.

It is important to know as much as possible about the behavior of the quality indicators before using them. A researcher that uses a quality indicator assuming that it has a certain behavior, but is not aware of its limitations, may arrive at wrong conclusions.

For the case of the G–Ranker, we see that it has the best performance, solving correctly the full benchmark. In this work, we have the hypothesis that it is possible to introduce more preferences, besides the ones derived from General Assumption 1, in a quality indicator to improve its evaluations, but without the introduction of important biases. Based on our experimentation we conclude that the hypothesis is correct. G–Ranker is the most informative of the indicators tested, and one of the least biased.

We can imagine a future where the quality indicators are designed, from the start, with a clear idea of what are the assumptions and biases introduced into the indicator. This way one can design better quality indicators, and indicators for specific preferences. In the case of the G–Ranker, one might wonder what other information could have been considered to improve it without introducing biases.

We also reviewed some of the most influential studies about comparison

methods, under the point of view of convergence and diversity, and one of the most important conclusions is that a very popular property known as "compatibility and completeness" is not only unnecessary but undesirable when constructing a model for the analyst preferences. A comparison method that is compatible and complete is too conservative and in many cases does not reflect the preferences of the analyst. An immediate consequence is that, contrary to general belief, unary comparison methods are not inferior to binary ones. Unary comparison methods are considered inferior because, at least theoretically, they can not be compatible and complete, but if "compatibility and completeness" is not a desirable property then this is not a reason to consider unary quality indicators as inferior. We must remark that the property defined by Hansen and Jaszkiewicz that they called "compatibility" is important and necessary for any comparison method that evaluates the total quality of an approximation set.

Another interesting question is if unary quality indicators can be compatible and complete under practical conditions. When it was demonstrated that unary QI cannot be compatible and complete, some theoretical assumptions were made, that are not met in practice. Under practical conditions, we arrive at a different result as can be seen in the Appendix A.

The G–Ranker is not perfect and we have made our most sincere effort to detect its weak points. We hope that the model we constructed becomes the basis for more accurate models, which are able to answer with the most precision possible when one approximation is better than another. Constructing a comparison method that introduces more preferences than General Assumption 1, opens the possibility to introduce biases in the evaluations. But, in order to be able to compare the quality of different approximations sets, there is a point where we need to take chances and make decisions.

Appendix A

The Cardinality of Some Sets of Non–dominated Sets

A.1 Introduction

In this appendix we make some demonstrations about the cardinality of some important sets of non–dominated sets. Based on these demonstrations, we review some claims about the limitations of unary quality indicators. Essentially, we show that under practical conditions we have no elements to affirm that we cannot construct a compatible and complete comparison method based on a finite number of unary quality indicators. In this thesis we show that being both compatible and complete is not a desirable property for a quality indicator, so this discussion may seem irrelevant. But, we must consider that in the future and under different conditions, the property of being both compatible and complete may become important. Also, this discussion is interesting from the theoretical point of view.

Next, we make a review of multi–objective optimization. Evolutionary multi–objective optimization (*EMO*) consists of maximizing or minimizing (or a mixture) a vector of objective functions $F(x) = \langle f_1(x), f_2(x), \ldots, f_n(x) \rangle$ subject to constraints using Evolutive Algorithms (EA). The objective func-

tions and constraints depend on a vector of variables $x \in \mathbf{R}^m$. We call X the set of all feasible vectors x, and Z the projection of X on objective functions space. Without loss of generality, we consider hereafter that we are minimizing the objective functions.

EMO uses the Pareto Optimality Criteria (POC). POC is defined through the relation between two vectors $z, w \in \mathbf{R}^n$ known as Pareto dominance. We have that z dominates w (z dom w) if $\forall i \in \{1, 2, \ldots, n\}$, $z^{(i)} \leq w^{(i)} \ \wedge \ \exists j \in \{1, 2, \ldots, d\} \mid z^{(j)} < w^{(j)}$, where $v^{(k)}$ stands for the k–th component of vector v. Otherwise, we say that z does not dominate w (z $\neg dom$ w). The goal is to find a set of vectors known as the Pareto Set (PS) defined as $PS = \{x \in X \mid \forall y \in X, F(y) \ \neg dom \ F(x)\}$. According to POC, all elements of PS are optimal, because they represent the different tradeoffs between the objective functions where it is not possible to improve one objective without degrading another. The projection of PS in objective functions space is called the Pareto Front ($\mathscr{PF}$), and is usually described as a surface that represents the best possible tradeoffs between the objective functions.

In recent years, many multi–objective evolutionary algorithms (MOEAs) based on POC have been developed. For a review of these algorithms consult [14] and [9]. Instead of generating a single solution, these algorithms generate a finite set A of vector solutions $x \in X$ that approximates the PS. These approximation sets have the characteristic that $\forall x, y \in A, F(x) \ \neg dom \ F(y) \wedge F(y) \ \neg dom \ F(x)$ and are usually called non–dominated sets (NS). We can consider these NSs as approximations of the Pareto Set. Also, we can consider these NSs as approximations to the Pareto Front, if we locate the sets in their projection in objective function space. Hereafter in this appendix we must locate points, sets, vectors, and solutions in the space of the objective functions.

The set Z is very important because all solutions generated by a MOEA are subsets of Z. Z can have many topologies, it can be discrete, finite, continuous, etc. In the rest of this appendix we assume the most extreme

case where $Z = \mathbf{R}^n$.

Define the set Ω as the set of all non–dominated sets we can generate with elements of Z. Ω is also a very interesting set, because all NSs generated by a MOEA are elements of Ω. One of the most important properties of a set is its cardinality. As we see later, the cardinality is related with the size of a set. The cardinality of Ω is especially important for performance measures, because many quality indicators are functions whose domain are subsets of Ω.

There are other sets of non–dominated sets that are important. Many MOEAs stores the non–dominated vectors they have found using an archive of fixed size k. This means that, once the value of k is decided for a MOEA, the set of all non–dominated sets we can obtain from that MOEA are those of size less than or equal to k. So, the set $\Omega_{\leq k}$ of all non–dominated sets of size less than or equal to k is also interesting.

Finally, we want to introduce another important set. Define $\Omega_{\leq a}$ as the set of all non–dominated sets of finite size. In practice, all MOEAs generate a finite number of explicit vector solutions for an evolutionary multi–objective problem. So, $\Omega_{\leq a}$ is a better representation of all possible outputs of a MOEA than Ω.

We find the cardinality of the sets described above and use this information to explain some properties of the performance measures described in [83]. The rest of the appendix is organized as follows: in Section A.2 we present some basic theorems from set theory, in Section A.3 we use set theory to find the cardinality of some important sets. Later, in Section A.4 we use the results of previous sections to make some demonstrations related to quality indicators for multi–objective algorithms.

A.2 Set Theory

In this section we present an overview of set theory and present all the definitions and results that are necessary to demonstrate the theorems and lemmas in the following sections. The concepts in this section are taken from [25] and [39]. The demonstrations of the results presented here can be consulted in [25] and [39].

Two sets, A and B, are equivalent ($A \sim B$) if it is possible to make a correspondence between the elements of both sets in such a way that to every element of A correspond one and only one element of B; and to every element of B correspond one and only one element of A. This kind of correspondence is called a one–to–one correspondence. The equivalence property is reflexive, transitive and symmetric. An injection from A to B is a one–to–one correspondence from A to a subset of B.

One of the most important concepts of set theory is that of cardinal numbers, or cardinality. Cardinal numbers are related with the size of a set. A cardinal number refers to an arbitrary member of a family of mutually equivalent sets. For example, the cardinal number 4 represents any set equivalent to {1,2,3,4}, like {a,b,c,d}, {"dog", "rat", "cat", "mouse"}, etc. We represent the cardinal number of a set A by $|A|$, for example $|\{a, b\}| = 2$. Infinite sets also have cardinal numbers. For positive integers $\mathbf{N} = \{1, 2, 3, \ldots\}$, we represent its cardinal number by a. For real numbers $\mathbf{R}$ we represent $|\mathbf{R}|$ by c. We represent the cardinal number of the set of functions defined in a continuous interval by f.

It is not possible to make an injection from A to B if $|A| > |B|$ because there are not enough different elements in B to be associated with the elements of A. If $A \subset B$ then $|A| \leq |B|$. An interesting result from set theory is that it is impossible to make an injection from $\mathbf{R}$ to $\mathbf{N}$, the set of natural numbers is somehow "smaller" than the set of real numbers. For two infinite sets A and B, $|A| < |B|$ if and only if there is an injection from A to B but there is no injection from B to A. If we can make an injection from A to B,

than $|A| \leq |B|$. It is proved that $a < c < f$.

A set A with cardinal number a is called countable and it is equivalent to the set of natural numbers. When listing its elements, a countable set is usually represented using "...", for example $A = \{a_1, a_2, \ldots\}$.

An interesting property of the infinite sets, like $\mathbf{N}$ and $\mathbf{R}$, is that it is possible to make a one–to–one correspondence between an infinite set and some of its subsets.

For two sets A and B, their union is represented by $A+B$. Their cartesian product is represented by $A \times B$. The cartesian product of a set with itself can be represented by an exponent. For example, $A \times A \times A = A^3$.

The power set of A, *P(A)*, is the set whose elements are all possible subsets of A and it is proved that $|A| < |P(A)|$. If $|A| = |B|$ then $|P(A)| = |P(B)|$.

The cardinal numbers of infinite sets are called transfinite numbers. The smallest transfinite number is a and all finite numbers are smaller than any transfinite number. We present a list of results of set theory, where $k > 0$ is a finite cardinal number and $m, n > 0$ are finite numbers.

a: $A \sim B \Longleftrightarrow |A| = |B|$. Two sets are equivalent if and only if they have the same cardinal number.

b: $c + k = c + a = c + c = c$. For a set A of cardinality c, if we add a finite number of elements to A, the resulting set has cardinality c. The same occurs if we add a–many elements to A or if we add c–many elements to A. For example $|\mathbf{R} + \{a, b, c\}| = |\mathbf{R}| = c$, $|(0, 1) + \mathbf{N}| = c$, $|[0, 1) + [1, 2)| = |[0, 1)| = c$. Similarly, we can extract a finite number of elements from a set of cardinality c and the resulting set has cardinality c.

c: $c \cdot k = c \cdot a = c \cdot c = c$. The cartesian product of a set A with cardinality c with a finite set results in a set with cardinality c. The same result is obtained if the cartesian product is evaluated with a set of cardinality a or c. For example $|\mathbf{R} \times \{1, 2, 3\}| = c$, $|\mathbf{R} \times \mathbf{N}| = c$, $|\mathbf{R} \times \mathbf{R}| = c$.

d: $c^m = c^a = c$. A set A with cardinality c elevated to a finite exponent results in an equivalent set. For example, $|\mathbf{R}^5| = |\mathbf{R}| = c$. The same result is obtained if A is elevated to a.

e: Let $|A| = c$. $|A^c| = |P(A)|$. If we elevate a set A of cardinality c to the exponent c, the result is a set with a bigger cardinality. The same cardinality of the power set of A. For example, $\mathbf{R}^c \sim P(\mathbf{R})$.

f: $c^m \times c^n = c^{(m+n)} = c$. For example, $\mathbf{R}^5 \times \mathbf{R}^2 \sim \mathbf{R}^7 \sim \mathbf{R}$.

g: $c + c + \ldots = c$. The sum of countable many sets, each of them with cardinality c, results in a set of cardinality c. For example $[0, 1) + [1, 2) + \ldots + [k, k + 1) + \ldots \sim \mathbf{R}$.

h: $|A| = |B| \Longrightarrow |P(A)| = |P(B)|$. If two sets have the same cardinality, then their power sets are equivalent.

i: $A \subset B \Longrightarrow |A| \leq |B|$. If A is a subset of B, then the cardinal number of A is less than or equal to the cardinal number of B.

j: Let $|A| = c$ and $|B| = a$: $f = |P(A)| > c = |P(B)| > a$.

k: If $C \subset B$ then: $A \sim C \Longrightarrow |A| \leq |B|$.

These results are used in the following sections.

A.3 The Cardinality of Some Important Sets

We introduce some demonstrations about the cardinality of some sets of non–dominated sets. These demonstrations are interesting by themselves from the theoretical point of view but they are also useful to prove the theorems we present later. We use extensively the Theorems (a)–(k) from Section A.2.

First we present the line S. The line S is a structure we borrow from [83] and we use it in the demonstration of some of the lemmas and theorems in

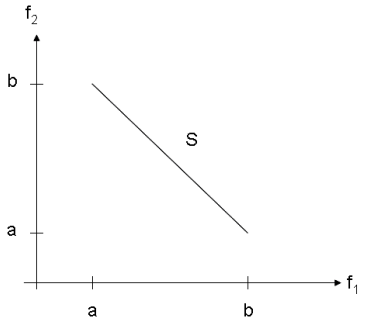

Figure A.1: A non–dominated set $S \subset \mathbf{R}^2$.

this section. It is a segment of line in $\mathbf{R}^n$, aligned in a direction so the points in the line are a non–dominated set. An example of the line S is shown in Figure A.1. The definition of S is the following:

Definition A.1. *Choose $a, b \in \boldsymbol{R}$ with $a < b$. The line S is defined as $S = \{\langle z^{(1)}, z^{(2)}, \ldots, z^{(n)} \rangle \in Z \mid a < z^{(1)} < b, z^{(2)} = b + a - z^{(1)}$ and $z^{(i)} = (a+b)/2$ for $i = 3$ to n $\}$. Also, define Ω_S as the set of all non–dominated sets we can generate from S.*

Note that the definition of $z^{(1)}$ and $z^{(2)}$ implies that all points in S are non–dominated between them. For simplicity, we give a fixed value to the rest of the $z^{(i)}$s. Ω_S is equivalent to the power set of S minus the empty set, because any subset of non–dominated set is also a non–dominated set with the exception of the empty set[1].

In Section A.1 we define the set Ω of all non–dominated sets we can create from Z. We are interested in the following question. How many non–dominated sets can be created from Z? In other words, what is the cardinality of Ω? To answer this question we present the following theorem:

[1]In order to make the demonstrations shorter, we consider that the empty set is not a non–dominated set. This makes no difference because our demonstrations hold even if we consider that the empty set is a non–dominated set.

Theorem A.1. *The cardinal number of* Ω *can be as big as* f*.*

Proof. Consider the line $S \subset Z$ described before and consider the most extreme case where $Z = \mathbf{R}^n$. The argumentation is the following (we denote by lowercase letters in parenthesis the theorems (a)–(k) from Section A.2):

A: $\Omega_S \subset \Omega \subset P(\mathbf{R}^n)$, because of the definition of S, Ω and Ω_S.

B: $|\Omega_S| \leq |\Omega| \leq |P(\mathbf{R}^n)|$, because of (i) and A.

C: $\Omega_S = P(S) - \{\emptyset\}$, because the subset of a NS is also a NS.

D: $|\Omega_S| = f$, because the power set of a set with cardinality c has cardinality f (j), S has cardinality c and C.

E: $P(\mathbf{R}^n) \sim P(\mathbf{R})$, because of (d) and (h).

F: $|P(\mathbf{R}^n)| = f$, because of E, (a) and (j).

G: $f \leq |\Omega| \leq f$, because of B, D and F.

H: $|\Omega| = f$, this is a direct consequence of G.

□

Note that the size of Ω is bounded by the size of $P(\mathbf{R}^n)$. $|P(\mathbf{R}^n)| = f$ so we have the following corollary:

Corollary A.1. *The cardinal number of* Ω *is at most* f*.*

Other demonstrations that will be useful in the rest of this work are about the cardinality of the sets of non–dominated sets of a fixed size. For example, define Ω_k as the set of all non–dominated sets in $\mathbf{R}^n$ of size k, for $k > 0$. What is the cardinality of Ω_1, Ω_2 and in general Ω_k? We respond to these questions in the following three lemmas.

Lemma A.1. *The cardinality of* Ω_1 *can be as big as* c*.*

Proof. By definition, Ω_1 is a set of the form $\{\{z\} \mid z \in Z\}$, so we can make a one to one correspondence between the elements $\{z\} \in \Omega_1$ with the corresponding $z \in Z$. This means that $\Omega_1 \sim Z$. The cardinality of Z can be as big as c, so recalling Theorem (a) from Section A we can conclude that $|\Omega_1|$ can be as big as c. □

Note that the size of Ω_1 is bounded by the size of Z. The cardinal number of Z can be at most c. So we have the following corollary:

Corollary A.2. *The cardinal number of Ω_1 is at most c.*

Lemma A.2. *The cardinality of Ω_2 can be as big as c.*

Proof. Consider the line S described before. Choose a point $s_1 \in S$ and define $S' = S - \{s_1\}$. Define Ω_2' as the set of all sets in the form $\{s_1, s\}$ for $s \in S'$. We have that all elements of Ω_2' are non–dominated sets of size two.

A: $|S'| = c$, because if we take a finite number of elements from a set of cardinality c, the resulting set has cardinality c (b). We define S' as $S - \{s_1\}$ and the cardinality of S is c, so the cardinal number of S' is c.

B: $|\Omega_2'| = c$, because we can make a one to one correspondence between Ω_2' and S'. For this, associate each element $\{s_1, s\} \in \Omega_2'$ with the corresponding element $s \in S'$. But the cardinality of S' is c (A), so the cardinal number of Ω_2' is c (a).

C: $\Omega_2' \subset \Omega_2$, because the elements of Ω_2' are non–dominated sets of size 2 and Ω_2 is the set of all non–dominated sets of size 2.

D: $|\Omega_2'| \leq |\Omega_2|$, because Ω_2' is a subset of Ω_2 (C), so its cardinality must be less than or equal to that of Ω_2 (i).

E: $c \leq |\Omega_2|$, because the cardinal number of Ω_2' is less than or equal to the cardinal number of Ω_2 (D), and the cardinal number of Ω_2' is c (B).

F: $|\Omega_2| \leq |\mathbf{R}^{2n}|$, because of (k) and because we can make an injection from Ω_2 to $\mathbf{R}^{2n}$. For this, sort the vectors $v = \left\langle v^{(1)}, v^{(2)}, ...v^{(n)} \right\rangle \in A$ for $A \in \Omega_2$ using the following rule: v precedes u, for $v, u \in A$, if $v^{(1)} < u^{(1)}$ or if $v^{(r)} < u^{(r)}$ when $v^{(i)} = u^{(i)}$ for $i = 1$ to $r - 1$. This way, every non–dominated set $A \in \Omega_2$ is associated with an unique pair of ordered vectors $v_1 = \left\langle v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(n)} \right\rangle$, $v_2 = \left\langle v_2^{(1)}, v_2^{(2)}, \ldots, v_2^{(n)} \right\rangle$. Associate each non–dominated set $A \in \Omega_2$ with the point $(v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(n)}, v_2^{(1)}, v_2^{(2)}, \ldots, v_2^{(n)}) \in \mathbf{R}^{2n}$ and we have the desired injection.

G: $|\Omega_2| \leq c$, because the cardinal number of Ω_2 is less than or equal to the cardinal number of $\mathbf{R}^{2n}$ (F) and the cardinal number of $\mathbf{R}^{2n}$ is c (d).

H: $c \leq |\Omega_2| \leq c$, because of E and G.

I: $|\Omega_2| = c$, this is a direct consequence of H.

□

Note that the size of Ω_2 is bounded by the size of $\mathbf{R}^{2n}$. $|\mathbf{R}^{2n}| = c$, so we have the following corollary:

Corollary A.3. *The cardinal number of Ω_2 is at most c.*

The proof of Lemma A.2 can be extended for Ω_k for $k > 2$, as is shown in the following demonstration.

Lemma A.3. *The cardinality of Ω_k, for $k > 2$, can be as big as c.*

Proof. Consider the line S described before. Choose $k - 1$ different points $s_1, s_2, \ldots, s_{k-1} \in S$ and define $S' = S - \{s_1, s_2, \ldots, s_{k-1}\}$. Define Ω'_k as the set of all sets in the form $\{s_1, s_2, \ldots, s_{k-1}, s\}$ for $s \in S'$. We have that all elements of Ω'_k are non–dominated sets of size k.

A: $|S'| = c$, because if we take a finite number of elements from a set of cardinality c, the resulting set has cardinality c (b). We define S' as $S - \{s_1, s_2, \ldots, s_{k-1}\}$ and the cardinality of S is c, so the cardinal number of S' is c.

B: $|\Omega'_k| = c$, because we can make a one–to–one correspondence between Ω'_k and S'. For this, associate each element $\{s_1, s_2, \ldots, s_{k-1}, s\} \in \Omega'_k$ with the corresponding element $s \in S'$. But the cardinality of S' is c (A), so the cardinal number of Ω'_k is c (a).

C: $\Omega'_k \subset \Omega_k$, because the elements of Ω'_k are non–dominated sets of size k and Ω_k is the set of all non–dominated sets of size k.

D: $|\Omega'_k| \leq |\Omega_k|$, because Ω'_k is a subset of Ω_k (C), so its cardinality must be less than or equal to that of Ω_k (i).

E: $c \leq |\Omega_k|$, because the cardinal number of Ω'_k is less than or equal to the cardinal number of Ω_k (D), and the cardinal number of Ω'_k is c (B).

F: $|\Omega_k| \leq |R^{kn}|$, because of (k) and because we can make an injection from Ω_k to R^{kn}. For this, sort the vectors $v = \left\langle v^{(1)}, v^{(2)}, ..., v^{(n)} \right\rangle \in A$ for $A \in \Omega_k$ using the following rule: v precedes u, for $v, u \in A$, if $v^{(1)} < u^{(1)}$ or if $v^{(r)} < u^{(r)}$ when $v^{(i)} = u^{(i)}$ for $i = 1$ to $r - 1$. This way, every non–dominated set $A \in \Omega_k$ is associated with a unique list of ordered vectors $v_1 = \left\langle v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(n)} \right\rangle, v_2 = \left\langle v_2^{(1)}, v_2^{(2)}, \ldots, v_2^{(n)} \right\rangle, \ldots, v_k = \left\langle v_k^{(1)}, v_k^{(2)}, \ldots, v_k^{(n)} \right\rangle$. Associate each non–dominated set $A \in \Omega_k$ with the point $(v_1^{(1)}, v_1^{(2)}, \ldots, v_1^{(n)}, v_2^{(1)}, v_2^{(2)}, \ldots, v_2^{(n)}, \ldots, v_k^{(1)}, v_k^{(2)}, \ldots, v_k^{(n)}) \in R^{kn}$ and we have the desired injection.

G: $|\Omega_k| \leq c$, because the cardinal number of Ω_k is less than or equal to the cardinal number of R^{kn} (F) and the cardinal number of R^{kn} is c (f).

H: $c \leq |\Omega_k| \leq c$, because of E and G.

I: $|\Omega_k| = c$, this is a direct consequence of H.

□

Note that the size of Ω_k is bounded by the size of $\mathbf{R}^{kn}$. $|\mathbf{R}^{kn}| = c$, so we have the following corollary:

Corollary A.4. *The cardinal number of Ω_k, for $k > 2$, is at most c.*

Based on the demonstrations presented before, we introduce the following theorem:

Theorem A.2. *The cardinal number of Ω_k, where k is a positive integer, is at most c.*

Theorem A.2 is useful in the demonstration of the two following theorems. These theorems are related with the cardinality of $\Omega_{\leq k}$ and $\Omega_{<a}$, defined in Section A.1.

Theorem A.3. *The cardinality of $\Omega_{\leq k}$ is at most c.*

Proof. We can represent $\Omega_{\leq k}$ with the following sum:

$$\Omega_{\leq k} = \Omega_1 + \Omega_2 + \ldots + \Omega_k \ . \tag{A.1}$$

the argumentation is as follows.

A: $|\Omega_i| = c$, for $i = 1$ to k, because of Theorem A.2 and we are considering the most extreme case.

B: $|\Omega_{\leq k}| = |\Omega_1 + \Omega_2 + \ldots + \Omega_k|$, because of Formula A.1.

C: $|\Omega_{\leq k}| = |\Omega_1| + |\Omega_2| + \ldots + |\Omega_k|$, because the different Ω_i are mutually disjoint.

D: $|\Omega_{\leq k}| = k \cdot c$, because of A and C.

E: $|\Omega_{\leq k}| = c$, because of D and because the product of c with a natural number $k > 0$ is equal to c (c).

□

Theorem A.4. *The cardinal number of $\Omega_{<a}$ is at most c.*

Proof. We can represent $\Omega_{<a}$ with the following sum:

$$\Omega_{<a} = \Omega_1 + \Omega_2 + \Omega_3 + \dots \quad . \tag{A.2}$$

Note that this sum has a–many elements, because we can make a one–to–one correspondence between the positive integers and the elements of the sum. For this we only need to associate each positive integer k with the corresponding Ω_k. The values of k increase without a limit, but they are always finite.

A: $|\Omega_i| = c$, for $i = 1, 2, \dots$, because of Theorem A.2 and we are considering the most extreme case.

B: $|\Omega_{\leq a}| = |\Omega_1 + \Omega_2 + \Omega_3 + \dots|$, because of Formula A.2.

C: $|\Omega_{\leq a}| = |\Omega_1| + |\Omega_2| + |\Omega_3| + \dots$, because the different Ω_k are mutually disjoint and B.

D: $|\Omega_{\leq a}| = c + c + c + \dots$, because of A and C.

E: $|\Omega_{\leq a}| = c$, because of D and (g).

□

A.4 Cardinalities and Quality Indicators

In this section we make an extensive use of the Theorems A.1– A.4 from Section A.3 and Theorems (a)–(k) from Section A.2.

An open problem in multi–objective optimization is how to evaluate the performance of different MOEAs. After every run, a MOEA generates a non–dominated set that approximates the Pareto Front. So, the evaluation of the MOEAs depends on the evaluation of the NSs it generates. A popular method to evaluate a non–dominated set is to use a unary quality indicator [83]. Define Ω_r as the set of all non–dominated sets that can be generated by a MOEA or the set of all non–dominated sets we are interested in. A unary quality indicator is a function $I : \Omega_r \rightarrow R$, that takes one NS as an argument (hence it is named unary) and returns a real number as an output. This real number is a measure of how good a NS is, and it is used to compare different NSs. For example, if we consider that the bigger the value of I the better, for two NSs $\mathscr{A}$ and $\mathscr{B}$, if $I(\mathscr{A}) > I(\mathscr{B})$ we can conclude that $\mathscr{A}$ is better than $\mathscr{B}$. It is also common to use several unary quality indicators in order to evaluate two non–dominated sets.

In [83] the concept of unary comparison method is defined . A unary comparison method $UC_{I,E}$ is formed by a combination $\mathbf{I} = (I_1, I_2, \ldots, I_k)$ of k unary quality indicators and a function $E : \mathbf{R}^k \times \mathbf{R}^k \rightarrow \{True, False\}$. For two NSs $\mathscr{A}$ and $\mathscr{B}$, if $UC_{I,E}(\mathscr{A}, \mathscr{B})$ is true, then the comparison method considers that $\mathscr{A}$ is better than $\mathscr{B}$. If $UC_{I,E}(\mathscr{A}, \mathscr{B})$ is false, then the comparison method considers that $\mathscr{A}$ is not better than $\mathscr{B}$.

Also, in [83] a property for comparison methods known as "compatibility and completeness" is introduced. Zitzler et al. [83] claim that "compatibility and completeness" is a very important property to have, because a compatible and complete comparison method is able to decide whether a NS $\mathscr{A}$ is better than a NS $\mathscr{B}$. It is proved [83] that a unary comparison method can not be compatible and complete. This is described in the following theorem:

Theorem A.5. *For multi–objective problems with 2 or more objectives, there exists no unary comparison method with a finite number k of quality indicators in* $\boldsymbol{I}$*, that is compatible and complete.*

Theorem A.5 has very important implications. It means that unary qual-

ity indicators have a limited capacity to evaluate whether a non–dominated set is better than another. In order to demonstrate this theorem, we need Lemma A.4.

Lemma A.4. *Let* $Z = \{\langle z^{(1)}, z^{(2)}, \ldots, z^{(n)}\rangle \in \boldsymbol{R}^n \mid \ a < z^{(i)} < b$ *, for* $i = 1$ *to* $n\}$*, be an open hypercube in* $\boldsymbol{R}^n$ *with* $n \geq 2$*,* $a, b \in \boldsymbol{R}$*. If there exists a compatible and complete unary comparison method with* $\boldsymbol{I} = (I_1, I_2, ..., I_k)$*, and an interpretation function* E*, then for all* $\mathscr{A}, \mathscr{B} \in \Omega$ *with* $\mathscr{A} \neq \mathscr{B}$ *there is at least one* I_j *in* $\boldsymbol{I}$ *such that* $I_j(\mathscr{A}) \neq I_j(\mathscr{B})$*.*

In other words, for a compatible and complete unary comparison method (CCUC), if $\mathscr{A} \neq \mathscr{B}$ then $\mathbf{I}(\mathscr{A}) \neq \mathbf{I}(\mathscr{B})$. Define Υ as the set of all different vectors $\mathbf{I}$, that we can generate with the outputs of k unary quality indicators. The cardinal number of Υ is c, because $\Upsilon = \mathbf{R}^k \sim \mathbf{R}$, (see (d) in Section A.2). Due to Lemma A.4 there must be an injection from Ω to $\mathbf{R}$. The demonstration of Lemma A.4 can be found in [83]. Next, we present a demonstration of Theorem A.5 that is equivalent to the demonstration presented in [83], but in our version it is easier to identify some details about why unary comparison methods cannot be compatible and complete. We use the Theorems (a)–(k) from Section A.2, denoted by lowercase letters in parenthesis, in the following proofs.

Proof of Theorem A.5. Consider the most extreme case where $Z = \mathbf{R}^n$ and let $\Omega_r = \Omega$.

A: $|\Omega_r| = f$, see Theorem A.1.

B: $|\Upsilon| = c$, because of the definition of Υ and (d).

C: $c < f$, because of (j).

D: We need to make an injection from Ω_r to Υ, because of Lemma A.4.

E: It is impossible to make an injection from Ω_r to Υ, because of A, B and C.

□

The conclusion is that no comparison method based on a finite number of unary quality indicators can be compatible and complete, because it leads to an absurd result. The central part of the demonstration is that we can not make an injection from Ω_r to Υ. This part of the demonstration is central for the rest of the study in this appendix, so we refer to it as the "cardinality contradiction".

Ziztler et al. [83] describe two conditions in which it is possible to construct a compatible and complete unary comparison method. The first condition is the use of an infinite number of unary quality indicators. They even mention the empirical attainment function [24] as a basis to construct a CCUC using an infinite number of quality indicators. An interesting question is why it is possible to construct such CCUC under the conditions just mentioned. The reason is very simple, and it is described in the following corollary.

Corollary A.5. *If we use c–many unary quality indicators, the cardinality contradiction vanishes.*

Proof. Redefine $\mathbf{I}$ as a combination of c–many unary quality indicators. Let $Z = R^n$ and $\Omega_r = \Omega$.

A: $|\Omega_r| = f$, because of Theorem A.1.

B: $|\Upsilon| = f$, because the number of different combinations of real numbers we can generate from $\mathbf{I}$ is $\mathbf{R}^c = f$ (e).

C: We need to make an injection from Ω_r to Υ, because of Lemma A.4.

D: It is possible to make an injection from Ω_r to Υ, because of A and B.

□

So, the cardinality contradiction vanishes if we use c–many unary quality indicators. Note that this is not true if we redefine $\mathbf{I}$ to contain a–many unary quality indicators, because in this case $|\Upsilon| = |\mathbf{R}^a| = c < f = |\Omega|$, so the cardinality contradiction holds.

The second special condition introduced in [83] is that there exists a comparison method that is compatible and complete if the size of the NSs to compare is equal or less than a fixed value k. They even describe that comparison method. Again, an interesting question is why it is possible to construct such a comparison method under that condition. Just as in the previous case, new conditions lead us to a new result. We state this in the following corollary:

Corollary A.6. *If we restrict the size of the non–dominated sets under comparison to be smaller than or equal to a fixed value $k > 0$ the cardinality contradiction vanishes.*

Proof. Let $Z = \mathbf{R}^n$ and $\Omega_r = \Omega_{\leq k}$.

A: $|\Omega_r| = c$, because of Theorem A.3.

B: $|\Upsilon| = c$, because the number of different vectors of real numbers we can generate from I with k unary quality indicators is $|\mathbf{R}^k| = c$ (d).

C: We need to make an injection from Ω_r to Υ, because of Lemma A.4.

D: It is possible to make an injection from Ω_r to Υ, because of A and B.

□

Unfortunately, the comparison method mentioned above is not useful in practice [83]. However, considering Corollary A.6, we wonder if other more useful comparison methods can be created.

Finally, we introduce a third special condition where the cardinality contradiction vanishes. This new special condition is to consider only non–dominated sets of arbitrary finite size. Note that this condition is less restrictive than the one mentioned in Corollary A.6, because we do not need to fix the size of the NSs to compare.

Corollary A.7. *If we only compare non–dominated sets $\mathscr{A} \in \Omega_{<a}$, the cardinality contradiction vanishes.*

Proof. Let $Z = \mathbf{R}^n$ and $\Omega_r = \Omega_{\leq a}$

A: $|\Omega_r| = c$, because of Theorem A.4.

B: $|\Upsilon| = c$, because of the definition of Υ and (d).

C: $\Omega_r \sim \Upsilon$, because of A, B and (a).

D: We need to make an injection from Ω_r to Υ, because of Lemma A.4.

E: It is possible to make an injection from Ω_r to Υ, because of C.

□

The consequence of Corollary A.7 is that we can not affirm that a compatible and complete comparison method based on a finite number of unary quality indicators is impossible to construct. All evolutionary multi–objective algorithms known so far generate a finite number of vectors as an output. An algorithm that generates an infinite number of explicit vector solutions, in finite time, is not possible. So, we have that the conditions of Corollary A.7 are met in practice.

Bibliography

[1] Russell L. Ackoff and Maurice W. Sasieni. *Fundamentals of Operations Research*. John Wiley and Sons, USA, 1968.

[2] Peter J. Angeline and Kenneth E. Kinnear, editors. *Advances in genetic programming*. MIT Press, Cambridge, MA, USA, 1996.

[3] Rutherford Aris. *Mathematical Modelling Techniques*. Pitman Publishing Limited, 39 Parker Street, London, 1978.

[4] David Avis, Binay K. Bhattacharya, and Hiroshi Imai. Computing the volume of the union of spheres. *The Visual Computer*, 3(6):323–328, 1988.

[5] Thomas Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.

[6] Nicola Beume and Günter Rudolph. Faster S-metric calculation by considering dominated hypervolume as Klee's measure problem. In Boris Kovalerchuk, editor, *Computational Intelligence*, pages 233–238. IASTED/ACTA Press, 2007.

[7] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding knees in multi-objective optimization. In *Eighth Conference on Parallel Problem Solving from Nature (PPSN VIII). Lecture Notes in Computer Science*, pages 722–731. Springer-Verlag, 2004.

[8] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Making, Theory and Methodology*. Elsevier Science, New York, 1983.

[9] Carlos A. Coello, D. Van Veldhuizen, and G. B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic/Plenum Publishers, New York, USA, 2002.

[10] Richard Courant. Variational methods for the solution of problems of equilibrium and vibrations. *Bulletin of the American Mathematical Society*, 49(1):1–23, 1943.

[11] George B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, New Yersey, 1963.

[12] Charles Darwin. *On the Origin of Species by Means of Natural Selection*. John Murray, London, 1859.

[13] Donald Dawson and Ronald G. Askin. Optimal new product design using quality function deployment with empirical value functions. *Quality and Reliability Engineering International*, 15:17–32, 1999.

[14] Kalyanmoy Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. John Wiley and Sons, Chichester, UK, 2001.

[15] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2002.

[16] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multi-Objective Optimization. Technical Report 112, Zurich, Switzerland, 2001.

[17] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 26(1):29–41, 1996.

[18] Russell C. Eberhart, Yuhui Shi, and James Kennedy. *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, USA, 2001.

[19] A. Farhang-Mehr and S. Azarm. Diversity assessment of pareto optimal solution sets: an entropy approach. *IEEE International Conference on E-Commerce Technology*, 1:723–728, 2002.

[20] Anthony V. Fiacco and Garth P. McCormick. *Nonlinear programming : sequential unconstrained minimization techniques*. John Wiley & Sons, New York, NY, 1968.

[21] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial Intelligence through Simulated Evolution*. Wiley, New York, 1966.

[22] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proc. 5th Int. Conf. Genetic Algorithms*, pages 416–423, San Mateo, CA, 1993. S. Forrest, Ed.

[23] C. M. Fonseca, L. Paquete, and M. López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2006)*, pages 1157–1163, 2006.

[24] Carlos M. Fonseca, Viviane Grunert da Fonseca, and Luis Paquete. Exploring the performance of stochastic multiobjective optimisers with the second-order attainment function. In *EMO*, pages 250–264, 2005.

[25] Abraham A. Fraenkel. *Set Theory and Logic*. Addison–Wesley Publishing Company, Reading, USA, 1966.

[26] K.R. Frisch. The logarithmic potential method for convex programming. Technical report, Institute of Economics, Univ. Oslo, 1955.

[27] U. M. Garcia-Palomares and O. L. Mangasarian. Superlinearly convergent quasi-newton algorithms for nonlinearly constrained optimization problems. *Mathematical Programming*, 11(1):1–13, 1976.

[28] A.M. Geoffrion, J.S. Dyer, and A. Feinberg. An interactive approach for multi–criterion optimization, with an application to the operation of an academic department. *Management Science*, 19(4):357–368, 1972.

[29] F. Glover. Tabu Search I. *ORSA Journal on Computing*, 1:190–206, 1989.

[30] F. Glover. Tabu Search II. *ORSA Journal on Computing*, 2:4–32, 1990.

[31] F. Glover and M. Laguna. *Tabu Search.* Kluwer Academic Publishers, Boston, 1997.

[32] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* ADDISON WESLEY, Boston, MA, 1989.

[33] Nicholas Ian Mark Gould. On the accurate determination of search directions for simple differentiable penalty functions. *IMA Journal of Numerical Analysis*, 6(3):357–372, 1986.

[34] Shih-Ping Han. *Superlinearly convergent variable metric algorithms for general nonlinear programming problems.* PhD thesis, 1974.

[35] Michael Pilegaard Hansen and Andrzej Jaszkiewicz. Evaluating the quality of approximations to the non-dominated set. Technical Report IMM-REP-1998-7, 1998.

[36] Magnus R. Hestenes. Multiplier and gradient methods. *Journal of Optimization Theory and Applications*, 4(5):303–320, 1969.

[37] John H. Holland. *Adaptation in natural and artificial systems.* MIT Press, Cambridge, MA, USA, 1992.

[38] J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Proc. 1st IEEE Conf. Evolutionary Computation*, pages 82–87, Piscataway, NJ, 1994.

[39] E. Kamke. *Theory of Sets*. Dover Publications, Inc., New York, USA, 1950.

[40] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proc. of the IEEE Int. Conf. on Neural Networks*, pages 1942–1948, Piscataway, NJ, 1995.

[41] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[42] Joshua Knowles and David Corne. On Metrics for Comparing Nondominated Sets. In *Proceedings of the 2002 Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 711–716.

[43] Joshua Knowles, Lothar Thiele, and Eckart Zitzler. A tutotial on the performance assessment of stochastic multiobjective optimizers. Technical Report TIK–Report No. 214, 2006.

[44] Joshua D. Knowles. *Local-Search and Hybrid Evolutionary Algorithms for Pareto Optimization*. PhD thesis, Massachusetts Institute of Technology, January 2002.

[45] X. Lai, M. Xie, and K. C. Tan. Dynamic programming for QFD optimization. *Quality and Reliability Engineering International*, 21:769–780, 2005.

[46] Pedro Larrañaga and Jose A. Lozano. *Estimation of distribution algorithms: A new tool for evolutionary computation*. Kluwer Academic Publishers, Boston, 2002.

[47] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multi-objective optimization. *Evolutionary Computation*, 10:263–282, 2002.

[48] A.F. Mehr and S. Azarm. Minimal sets of quality metrics. In *EMMO 2003, Lecture Notes in Computer Science 2632*, pages 405–417. C.M. Fonseca et al. (Eds.).

[49] Kaisa Miettinen. Some methods for nonlinear multi–objective optimization. In E. Zitzler et al., editor, *Evolutionary Multi–Criterion Optimization*, pages 1–20. Springer Berlin / Heidelberg, 2001.

[50] Miguel Angel Moreles and Salvador Botello. *Notas de Modelación y Métodos Numéricos I, Memorias del Taller de Métodos Numéricos en Ingeniería y Ciencias Aplicadas*. CIMAT-CIMNE, Guanajuato, Mexico, 2007.

[51] Boris Naujoks and Nicola Beume. Multi-objective optimisation using S-metric selection: application to three-dimensional solution spaces. In *In CEC2005*, pages 1282–1289. Press, 2005.

[52] G. L. Nemhauser, A.H.G. Rinnooy Kan, and M.J. Todd. *Optimization*. Elsevier Science Publishers B.V., Amsterdam, 1989.

[53] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer–Verlag, New York, 1999.

[54] Tatsuya Okabe, Yaochu Jin, and Bernhard Sendhoff. A Critical Survey of Performance Indices for Multi-Objective Optimisation. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC-2003)*, pages 878–885. IEEE Press, 2003.

[55] M. H. Overmars and Chee-Keng Yap. New upper bounds in Klee's measure problem. In *SFCS '88: Proceedings of the [Proceedings 1988]*

29th Annual Symposium on Foundations of Computer Science, pages 550–556, Washington, DC, USA, 1988. IEEE Computer Society.

[56] M. J. D. Powell. *A method for nonlinear constraints in minimization problems*. R. Fletcher, ed., Academic Press, New York, NY, 1969.

[57] M. J. D. Powell. *A fast algorithm for nonlinearly constrained optimization calculations*. Springer Berlin / Heidelberg, Berlin, 1978.

[58] Kenneth Price, Rainer M. Storn, and Jouni A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[59] Gregory J. E. Rawlins. *Foundations of Genetic Algorithms*. Morgan Kaufmann Publishers, Inc., San Mateo, CA., 1991.

[60] T. Runarsson and X. Yao. Stochastic ranking for constrained evolutionary optimization. *IEEE Transactions on Evolutionary Computation*, 4:284–294, 2000.

[61] J. D. Schaffer. *Some experiments in machine learning using Vector Evaluated Genetic Algorithms*. PhD thesis, Vanderbilt University, November 1984.

[62] J.R. Schott. *Fault tolerant design using single and multicriteria genetic algorithm optimization*. PhD thesis, Massachusetts Institute of Technology, May 1995.

[63] Hans-Paul Paul Schwefel. *Evolution and Optimum Seeking: The Sixth Generation*. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[64] David J. Sheskin. *Handbook of Parametric and Non-Parametric Statistical Procedures*. Chapman and Hall/CRC, USA, 2004.

[65] Vinicius L. S. Silva, Elizabeth F. Wanner, Sergio A. A. Cerqueira, and Ricardo H. C. Takahashi. A new performance metric for multiobjective optimization: the integrated sphere counting. In *IEEE Congress on Evolutionary Computation*, pages 3625–3630, 2007.

[66] P. G. Spirakis. The volume of the union of many spheres and point inclusion problems. In *Proceedings on STACS 85 2nd annual symposium on theoretical aspects of computer science*, pages 328–338, New York, NY, USA, 1985. Springer-Verlag New York, Inc.

[67] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. Technical report, 1993.

[68] R.E. Steuer. *Multiple Criteria Optimization: Theory, Computation and Applications.* John Wiley & Sons, Inc., 1986.

[69] Evangelos Triantaphyllou. *Multi-criteria decision making methods : a comparative study.* Kluwer Academic Pub., Boston, 2000.

[70] E.L. Ulungu. *Optimisation combinatoire multicritére: détermination de lensemble des solutions efficaces et méthodes interactives.* PhD thesis, Universit de Mons-Hainaut, 1993.

[71] David A. Veldhuizen. *Multiobjective Evolution Algorithms: Classifications, Analyses, and New Innovations.* PhD thesis, Force Institute Technology, Wright Patterson AFB, May 1999.

[72] Michael D. Vose. *The simple genetic algorithm: foundations and theory.* MIT Press, Cambridge, MA, 1999.

[73] R. B. Wilson. *A Simplicial Algorithm for Concave Programming.* PhD thesis, Harvard University, 1963.

[74] David H. Wolpert and William G. Macready. No free luch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[75] Stephen J. Wright. *Primal-Dual Interior-Point Methods*. SIAM Publications, Philadelphia, Pa, 1997.

[76] Leung Yiu-Wing and Wang Yupinge. U-measure: a quality measure for multiobjective programming. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 33(3):337–343, May 2003.

[77] Paul K. Yoon, Ching-Lai Hwang, and Kwangsun Yoon. *Multiple Attribute Decision Making: An Introduction (Quantitative Applications in the Social Sciences)*. Sage Pubn. Inc, March 1995.

[78] E. Zitzler, D. Brockhoff, and L. Thiele. The Hypervolume Indicator Revisited: On the Design of Pareto-compliant Indicators Via Weighted Integration. In S. Obayashi et al., editors, *Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, volume 4403 of *LNCS*, pages 862–876, Berlin, 2007. Springer.

[79] E. Zitzler, M. Laumanns, and L. Thiele. Spea 2: Improving the strength pareto evolutionary algorithm. Technical Report Technical report 103, 2001.

[80] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Technical Report Technical report 43, 1999.

[81] Eckart Zitzler. *Evolutionary Algorithms Multiobjective Optimization: Methods and Applications*. PhD thesis, Swiss Federal Institute of Technology (ETH), November 1999.

[82] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195, 2000.

[83] 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):529–533, May 2003.