2012-01-01

Development Of New Mathematical Methods For Post-Pareto Optimality

Victor Manuel Carrillo
University of Texas at El Paso, vmcsaucedo@gmail.com

DEVELOPMENT OF NEW MATHEMATICAL METHODS FOR POST-PARETO OPTIMALITY

VICTOR M. CARRILLO

Program in Computational Science

APPROVED:

_____
Heidi Taboada, Ph.D., Chair

_____
Jose F. Espiritu, Ph.D., Ph.D.

_____
Salvador Hernandez., Ph.D.

_____

Benjamin C. Flores, Ph.D.
Dean of the Graduate School

to my family and

to the loving memory

of my sister Rosario Elena

DEVELOPMENT OF NEW MATHEMATICAL METHODS FOR POST-PARETO OPTIMALITY

by

VICTOR M. CARRILLO

THESIS

Presented to the Faculty of the Graduate School of

The University of Texas at El Paso

in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE

Program in Computational Science

THE UNIVERSITY OF TEXAS AT EL PASO

December 2012

Acknowledgements

Thanks to my wife for supporting me in to succeed in this journey that has not ended yet.
Thanks to Dr Heidi Taboada my advisor for giving me the opportunity to start working on this research area. Besides for devising the correct strategy to become this work a reality.
I want to thank also my friends because when I needed their support they were always there for me to end this work.

Abstract

Many real-world applications of multi-objective optimization involve a large number of objectives. A multi-objective optimization task involving multiple conflicting objectives ideally demands finding a multi-dimensional Pareto-optimal front. Although the classical methods have dealt with finding one preferred solution with the help of a decision-maker, evolutionary multi-objective optimization (EMO) methods have been attempted to find a representative set of solutions in the Pareto-optimal front. Multiple objective evolutionary algorithms (MOEAs), which are biologically-inspired optimization methods, have become popular approaches to solve problems with multiple objective functions. With the use of MOEAs, multiple objective optimization becomes a two-part problem. First, the multiple objective optimization problem needs to be formulated and successfully solved using an MOEA. Then, a non- dominated set -also known as efficient or Pareto frontier- needs to be analyzed to select a solution to the problem. This can represent a challenging task to the decision-maker because this set can contain a large number of solutions. This decision-making stage is usually known as the post-Pareto analysis stage.

This thesis presents two different methods to perform post-Pareto analysis. The first method is the generalization of a method known as the non-numerical ranking preferences (NNRP) method. This method can help decision makers reduce the number of design possibilities to small subsets that clearly reflect their objective function preferences without having to provide specific weight values. Previous research has only presented the application of the NNRP method using three and four objective functions but had not been generalized to the case of *n* objective functions. The work presented in this thesis expands the NNRP method. The second method presented in this thesis uses a non-uniform weight generator method to reduce the size of the Pareto-optimal set. Both methods have been tested on different problem instances with successful results.

Table of Contents

Page

List of Tables

List of Figures

Chapter 1:

1. Introduction

Most real-world engineering optimization problems are implicitly or explicitly multi-objective (Mietinnen,1999). Multiple-objective optimization problems can be found in many areas and fields such as in economics, biology, engineering, etc. Almost all the optimization problems in real life involve more than one objective to be optimized, and normally those objectives are in conflict with each other. Such a situation is very easy to observe in real life. For instance, a high performance product is also a high cost item, and a customer always seeks merchandise with high performance, but at the lowest cost. Even though several methods for solving multi-objective optimization problems have been developed and studied, little work has been done on the evaluation of results obtained in multi-objective optimization. We can say that multiple objective optimization is a two phase problem; first we need to formulate and solve the multiple objective optimization problem and then the decision maker needs to select a solution for system implementation. This second stage is known as post-Pareto analysis and is the main focus of this thesis.

Generally, there are two primary approaches to solve multiple-objective optimization problems: mathematical methods and meta-heuristic methods. The first approach involves the aggregation of the attributes into a linear combination of the objective functions, also known as scalarization (Nakayama *et al.*, 2009). The second approach involves populating a number of feasible solutions along the Pareto frontier, and the final solution is a set of non-dominated solutions (Branke *et al.*, 2008).

The idea of the Pareto-frontier is to compare all solutions against each other, where the best fitted solutions dominate the weaker, which in turn are said to be dominated. The set of non-dominated solutions is known as the Pareto optimal set.

Without a loss of generality, a solution x_1 dominates a solution x_2 iff the two following conditions are true:

x_1 is no worse than x_2 in all objectives, i.e. $f(x_1) \leq f(x_2)\ \forall\ i,\ i \in \{1,2,\dots,n\}$

x_1 is strictly better than x_2 for at least one objective, i.e. $f(x_1) < f(x_2)$ for at least one i

Therefore, the solution of a multiple objective optimization is a set of non-dominated solutions. Sometimes, the Pareto set can contain a large number (in some cases, thousands) of solutions. From the decision-maker perspective, consideration of all the non-dominated solutions can be prohibitive and inefficient. The selection of solutions from the Pareto set is called post-Pareto optimality analysis**.** Some examples of mathematical methods used to solve multiple objective optimization problems are: goal programming, the lexicographic method, the weighted sum method and utility functions (Collete &Siarry, 2003). On the other hand, some meta-heuristic methods are: Tabu-search, neural networks, particle swarm, ant colony optimization and genetic algorithms, among others (Yang ,2008). Most of the meta-heuristic methods obtain as a solution a set of Pareto optimal solutions based in the Pareto dominance concept.

In this thesis, two methods are presented for post-Pareto analysis. The first method is a generalization of the non-numerical ranking preferences weight method proposed by Taboada & Coit (2006) and the second method is an approach based on a uniform random number generator developed by Carrillo and Taboada (2012) .

This thesis is divided into six chapters. Chapter 2introduces an overview of some of the classical multi-objective optimization methods. Chapter 3 presents an overview of how the post-Pareto optimality analysis has been addressed in the past, as well as some of the methodologies and objectives considered in previous research. Chapter 4 presents the development of the generalization of the non-numerical ranking preferences method. Chapter 5 shows the non-uniform weight generator and its development. Finally, Chapter 6 presents conclusions and future work.

Chapter 2:

2. Background & Previous Work

This chapter presents an introduction to multi-objective optimization and its solution methodologies.

2.1 Multi-objective Optimization

Optimization refers to finding one or more feasible solutions which correspond to extreme values of one or more objective functions. If an optimization problem involves only one objective function, the task of finding the optimal solution is called *single objective optimization*.

When an optimization problem involves more than one objective function, the task of finding a solution the problem *is known as multi-objective optimization* (Marler and Arora, 2004).

A Multi-objective optimization problem is defined as shown in Equation 1:

$$\begin{aligned} &\textit{Minimize / Maximize} \quad f(x) = \left(f_1(x), \ldots, f_k(x) \right) \\ &\textit{s.t.} \qquad g_j(x) \le 0 \;\; ; j = 1, \ldots, m \\ &\qquad\qquad x \in \mathbb{R}^n \end{aligned} \tag{1}$$

Where: n is the number of objective functions $f_n(x)$, m is the number of inequality constraints $g_j(x)$, $x \in \mathbb{R}^n$ is a vector of design variables, and $f(x) \in \mathbb{R}^k$ is a vector of n objective functions , with $f_i : \mathbb{R}^n \to \mathbb{R} \;\; for \; i = 1, ..., k$. The feasible or decision space is defined as $X = \left\{ x \in \mathbb{R}^n \mid g_j(x) \le 0 \;\; ; j = 1, \ldots, m \right\}$. The objective space is defined as the image of the feasible decision space X i.e. $Z = \{ f(x) \mid x \in X \}$. The idea of a solution for Equation 1 can be unclear, because a single point that minimizes all objectives simultaneously usually does not exist.

For example given the following two dimensional multi-objective optimization problem

$$\begin{aligned} &\textit{Minimize} \quad f_1(x) \\ &\textit{Minimize} \quad f_2(x) \\ &\textit{s.t.} \quad g_j(x) \le 0 \;\; ; j = 1, \ldots, m \quad x \in \mathbb{R}^3 \end{aligned} \tag{2}$$

The image of the decision space under the objective functions $f_1(x), f_2(x)$ is shown as in Figure 1. From this perspective it is hard, if not impossible to decide which is or are the optimum values to the problem. The yellow points in the purple region could be some of the best points to be considered as optimal points.

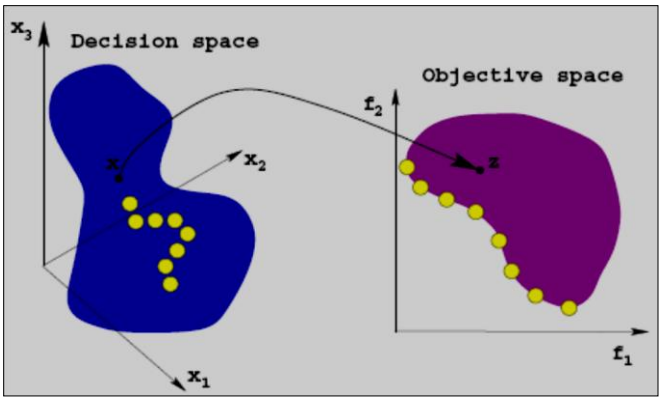

Figure 1:Pareto-optimal set and Pareto-front representations (yellow dots)

In a simple optimization problem, the notion of optimality is straightforward. The search is for the best (the minimum or the maximum) value of an objective function. In a multi-objective optimization problem, the notion of optimality is not as obvious as can be seen from Figure 1.

2.2 Pareto optimal set

A new definition of optimality must be found, a definition that accounts for all the criteria. That is when the concept of Pareto optimality arises. Although in general no solution is best with respect to all the criteria simultaneously, there exist a set of solutions that are strictly better than the remaining ones in the whole search space when optimizing all of the objectives simultaneously. This set of solutions is known as the Pareto optimal set or set of the non-dominated solutions. The complementary set of solutions is called the dominated solutions set. The image of the Pareto-optimal set under the objective functions is called the Pareto front (Mietinnen, 1999).

Example 1: Let us assume that each one of the objective spaces in Figure 2 corresponds to each one of the four multi-objective optimization problems. The red line corresponds to their respective Pareto fronts.

a) *Min* $f_1(x)$, *Min* $f_2(x)$ *s.t.* $g_j(x) \leq 0$ $j=1,..,m$ $x \in R^2$

b) *Min* $f_1(x)$, *Max* $f_2(x)$ *s.t.* $g_j(x) \leq 0$ $j=1,..,m$ $x \in R^2$

c)*Max* $f_1(x)$, *Min* $f_2(x)$ *s.t.* $g_j(x) \leq 0$ $j=1,..,m$ $x \in R^2$

d) *Max* $f_1(x)$, *Max* $f_2(x)$ *s.t.* $g_j(x) \leq 0$ $j=1,..,m$ $x \in R^2$

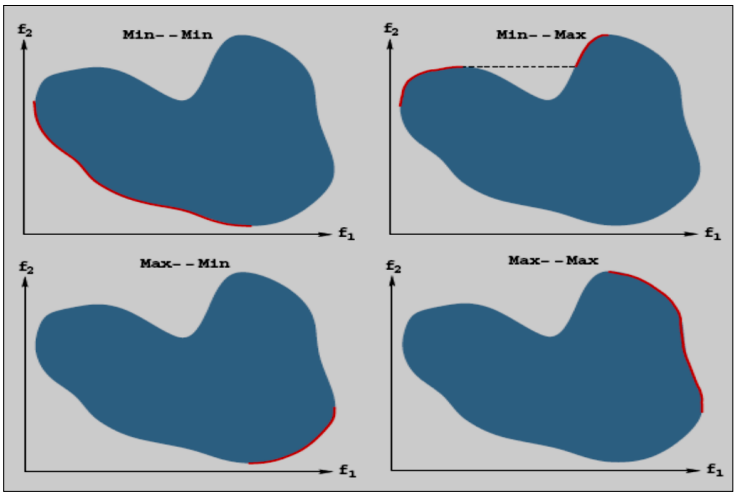

Figure 2: Pareto fronts represented as red lines.

Example 2: For a three objective minimization problem the Pareto front is shown in Figure 3

$$Min\ f_1(x), Min\ f_2(x), Min\ f_3(x)\ \ s.t.\ \ g_j(x) \leq 0\ j=1,..,m\ x \in R^2$$

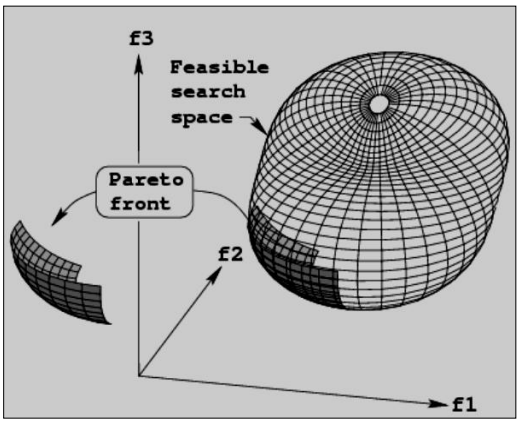

Figure 3: 3D Pareto front.

Consequently, the idea of Pareto optimality is used to describe solutions for MOP problems. A solution point is Pareto optimal if it is not possible to move from that point and improve at least one objective function without detriment to any other objective function. Alternatively, a point is weakly Pareto optimal if it is not possible to move from that point and improve (or worsen

depending either minimizing or maximizing) all objective functions simultaneously(souce of Fig 1 and Fig 2: *http://www.lania.mx/~ccoello/EMOO/* **)**

2.3 Pareto optimality and non-dominance definition

The formal definition of non-dominance is:

Definition 1 An $x^* \in X$ is called a Pareto optimal or non-dominated point if

$f_i(x^*) \leq f_i(x)$ *for all i and there is at least an index j such that* $f_j(x^*) < f_j(x)$

Example 2 Identify the non-dominated set from the following points where all objectives are to be minimized (Deb,2004):

Table 1: Data in R^3 to check non-dominance

Solution	f_1	f_2	f_3
A	2	3	1
B	5	1	10
C	3	4	10
D	2	2	2
E	3	3	2
F	4	4	4

Solution: Comparing all values of row A vs. row B in Table 1, A does not dominate B since 3 > 1 in the second column. Similarly, comparing row A against all the remaining rows we can see that A dominates points C, E and F as shown in Table 2. Continuing with the same procedure Tables 3 , 4 ,and 5 show the solutions that B, C, and D dominate.

Table 2: A dominance

Comparing A against the rest of the solutions				
A-B	✓	x	✓	Does not dominate B
A-C	✓	✓	✓	A dominates C
A-D	✓	x	✓	Does not dominate D
A-E	✓	✓	✓	A dominates E
A-F	✓	✓	✓	A dominates F

Table 3: B dominance

Comparing B against the rest of the solutions				
B-A	x	✓	x	Does not dominates A
B-C	x	✓	✓	Does not dominates C
B-D	x	✓	x	Does not dominates D
B-E	x	x	x	Does not dominates E
B-F	x	✓	x	Does not dominates F

Table 4: C dominance

Comparing C against the rest of the solutions				
C-A	x	x	x	Does not dominates A
C-B	✓	x	x	Does not dominates B
C-D	x	x	x	Does not dominates D
C-E	✓	x	x	Does not dominates E
C-F	✓	✓	x	Does not dominates F

Table 5: D dominance

Comparing D against the rest of the solutions				
D-A	✓	✓	x	Does not dominates A
D-B	✓	x	✓	Does not dominates B
D-C	✓	✓	✓	D dominates C
D-E	✓	✓	✓	D dominates E
D-F	✓	✓	✓	D dominates F

From the tables above, we find that A , B and D are non-dominated, thus the Pareto-front is the set of points {A,B,D}.

Definition 2 An $x^* \in X$ is called a weak Pareto solution if $f_i(x^*) < f_i(x)$ *for all* $i = 1,...,n$.

In Figure 3(Nakayama *et al*, 2009) is an example of an infinity set of weak Pareto solutions, which poses a dilemma for a decision maker.

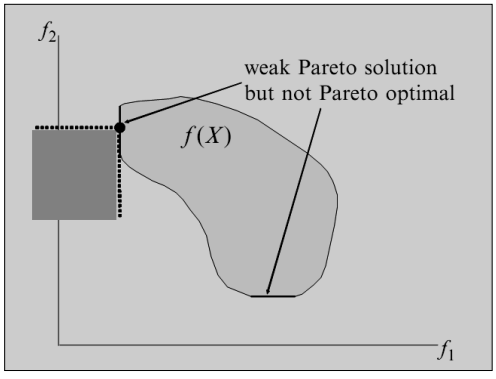

Figure 4: Flat borders are weak Pareto solutions

Generating Pareto optimal solutions plays an important role in Multi-objective optimization and mathematically the problem is considered to be solved when the Pareto optimal set is found. However, this is not always enough because the Pareto-optimal set can potentially contain a very large number of solution. It is desirable to select one solution or a small subset of solutions to present to the decision-maker for system implementation.

2.4 Conditions for Pareto-Optimality

In this section the Pareto-optimality conditions in a multi-objective optimization problem are presented. Consider Equation 1 below:

$$\begin{aligned} &\textit{Minimize / Maximize} \quad f(x) = \left(f_1(x), \ldots, f_k(x) \right) \\ &\qquad\qquad s.t. \qquad g_j(x) \le 0 \;\; ; j = 1, \ldots, m \\ &\qquad\qquad\qquad\quad x \in \mathbb{R}^n \end{aligned}$$

It is assumed that all objective and constraint functions are continuously differentiable.

The following statement is known as the necessary condition for Pareto-optimality:

Theorem 1 (Fritz-John necessary condition) A necessary condition for x^* to be

Pareto-optimal is that there exist vectors $\lambda \ge 0$ and $\mu \ge 0$, (where $\lambda \in \mathbb{R}^k$, $\mu \in \mathbb{R}^m$ and $\lambda \neq 0$, $u \neq 0$) such that the following conditions are true:

$$1. \quad \sum_{i=1}^{k} \lambda_i \nabla f_i(x^*) + \sum_{j=1}^{m} \mu_i \nabla g_j(x^*) = 0, \quad \textit{and} \tag{3}$$

$$2. \quad \mu_j g_j(x^*) = 0 \; \textit{for all} \;\; j = 1, \ldots, m$$

For an unconstrained multi-objective optimization problem the above necessary conditions are reduced to $\sum_{i=1}^{k} \lambda_i \nabla f_i(x^*) = 0$, or in matrix form:

$$\begin{bmatrix} \frac{\partial f_1(x^*)}{\partial x_1} & \cdots & \frac{\partial f_k(x^*)}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_1(x^*)}{\partial x_n} & \cdots & \frac{\partial f_k(x^*)}{\partial x_n} \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \vdots \\ \lambda_k \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} \tag{4}$$

If the above matrix is square n x n and x^* is a Pareto-optimal point then it is enough to compute the Jacobian matrix determinant:

$$\det\begin{bmatrix} \frac{\partial f_1(x^*)}{\partial x_1} & \cdots & \frac{\partial f_n(x^*)}{\partial x_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial f_1(x^*)}{\partial x_n} & \cdots & \frac{\partial f_n(x^*)}{\partial x_n} \end{bmatrix} = 0 \qquad (5)$$

Example 3 (Deb, 2004) Identify the candidate Pareto-optimal points

solutions and the candidate Pareto-front for the following problem:

$$Min\ f_1(x_1, x_2) = x_1^4 - 4x_1x_2$$
$$Min\ f_2(x_1, x_2) = x_1 + 2x_2^2$$

Solution: Note that there are no constraints, therefore the determinant of the

Jacobian matrix must be calculated and equated to zero to obtain the candidate Pareto-optimal points.

$$\text{Indeed, } \begin{vmatrix} \frac{\partial f_1(x^*)}{\partial x_1} & \frac{\partial f_1(x^*)}{\partial x_2} \\ \frac{\partial f_2(x^*)}{\partial x_1} & \frac{\partial f_2(x^*)}{\partial x_2} \end{vmatrix} = \begin{vmatrix} 4x_1^3 - 4x_2 & -4x_1 \\ 1 & 4x_2 \end{vmatrix} = 16x_1^3 - 416x_2^2 + 4x_1 = 0$$

Solving for x_1 and x_2 the candidate Pareto-optimal points is the set

$$\left\{ (x_1, x_2) : x_2 = \frac{x_1 \pm \sqrt{x_1^6 + x_1}}{2} \right\}$$

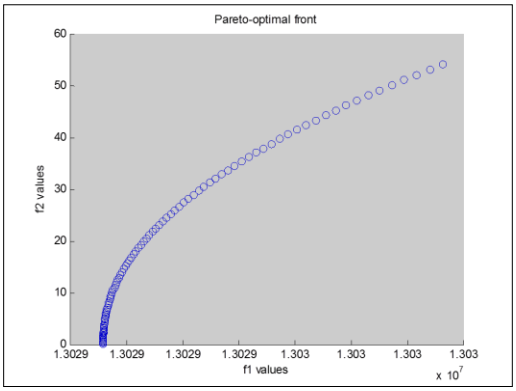

Figure 5: Pareto front solutions from example 3

The following theorem states sufficient conditions for a solution to be Pareto-optimal for convex functions.

Theorem 2 (Karush-Kuhn-Tucker (KKT) sufficient condition for Pareto-optimality)

Let the objective functions of the MOOP in equation (1) be convex and the constraint functions concave. Let the objective and constraint functions be continuously differentiable at a feasible solution x^*. A sufficient condition for x^* to be a Pareto-optimal solution is that there exist vectors $\lambda \geq 0$ and $u \geq 0$, (where $\lambda \in \mathbb{R}^k$, $\mu \in \mathbb{R}^m$) such that the following equations are true:

1. $\sum_{i=1}^{k} \lambda_i \nabla f_i\left(x^*\right) + \sum_{j=1}^{m} \mu_i \nabla g_j\left(x^*\right) = 0, \quad and$ (6)

2. $\mu_j g_j(x^*) = 0 \; for \; all \quad j = 1, \ldots, m$

Example 4 (Deb, 2004) For the following problem

$$
\begin{aligned}
&Min \;\; f_1(x_1, x_2) = x_1 \\
&Min \; f_2(x_1, x_2) = x_2 \\
&s.t. \;\; g(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 2)^2 \leq 4
\end{aligned}
$$

find the Pareto-optimal region.

The KKT conditions are: $\sum_{i=1}^{k} \lambda_i \nabla f_i\left(x^*\right) - \sum_{j=1}^{m} u_i \nabla g_j\left(x^*\right) = \lambda_1 \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \lambda_2 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \mu \begin{bmatrix} 2(x_1 - 2) \\ 2(x_1 - 2) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

Thus, solving the system of equations

$$
\begin{aligned}
&\lambda_1 + 2\mu(x_1 - 2) = 0 \\
&\lambda_2 + 2\mu(x_2 - 2) = 0 \\
&\mu\left[(x_1 - 2)^2 + (x_2 - 2)^2 - 4\right] = 0
\end{aligned}
$$

the Pareto-optimal region is $\left\{ (x_1, x_2) : x_1 = \frac{16\mu^2 - \lambda_1^2}{4\mu^2} , x_2 = \frac{\lambda_1^2}{4\mu^2} \;\; , \;\; with \, 0 < \lambda_1 \leq 4\mu, \; \mu > 0 \right\}$

A subset of the Pareto-optimal front is

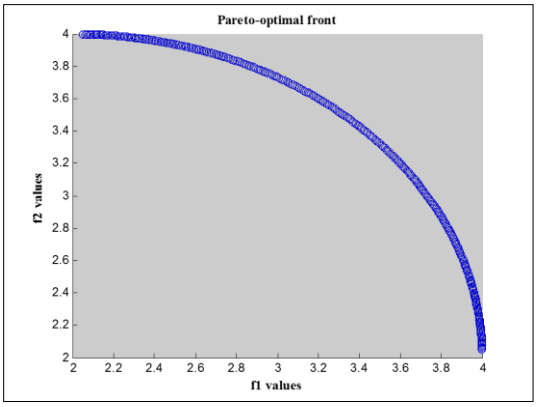

Figure 6: Subset of Pareto-front for example 4

Note that it is hard for the decision maker to select a solution from that large set of alternatives in the Pareto front.

2.5 Classical methods for Pareto-optimality(*)

(*)*Classical methods is the terminology used by Kalyanmoy Deb(2004) to distinguish from evolutionary optimization methods.*

A multi-objective optimization problem with more than two objective functions is hard to solve with just the optimality conditions presented in the last section. In general, multi-objective optimization

problems were solved by using aggregating approaches or by scalarization.

Scalarization means that the objective functions of the multi-objective problem are transformed into a single (scalar) function or a real valued function

i.e.: a transformation $F:\mathbb{R}^k \to \mathbb{R}$ is applied to $f(x)$ such that problem (1) is now written as:

$$\begin{aligned} &\textit{Minimize} \quad F\left(f(x)\right) \\ &\textit{s.t.} \quad g_j(x) \le 0 \;;\; j=1,\ldots,m \\ &\qquad x \in \mathbb{R}^n F\left(f(x)\right) \end{aligned} \qquad (7)$$

After the multi-objective optimization problem has been scalarized, the single nonlinear optimization methods can be applied. One of the most used scalarization methods is the weighted sum method.

2.5.1 Weighted Sum Method

In this method, the weights of the objective functions are usually chosen in proportion to its relative importance in the problem. Otherwise they are selected at random.

$$\begin{aligned} \textit{Minimize} \qquad & F(x) = \sum_{i=1}^{k} w_i f_i(x) \\ s.t. \quad & g_j(x) \le 0 \;\; ; j = 1, \ldots, m \\ & x \in \mathbb{R}^n \;\; , \;\; \sum_{i=1}^{k} w_i = 1 \;\; , \;\; w_i > 0 \end{aligned} \qquad (8)$$

The optimal points obtained when solving the above problem with single nonlinear optimization are Pareto-optimal points due to the following theorem:

Theorem 3 The solution to the problem by equation (8) is Pareto-optimal if the weight is positive for all objectives.

Figures 6 and 7 show the search for Pareto points using the weighted sum method. (The source of all figures in this chapter is: "Classic methods for multi-objective optimization" by Guizeppe Narzizi)

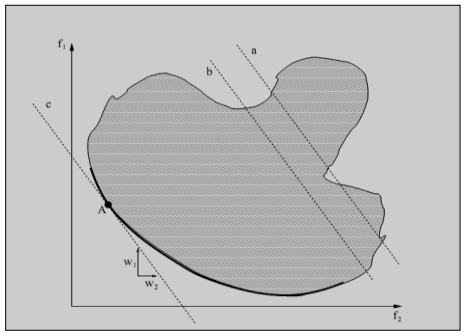

Figure 7: Weighted sum method on a convex Pareto-front

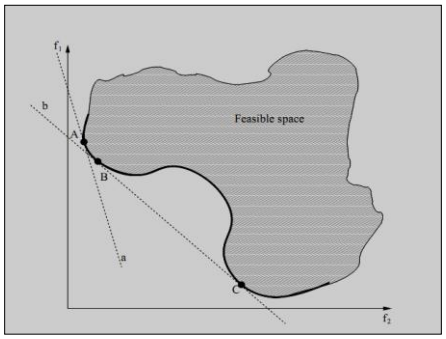

Figure 8: Disadvantage of the weighted method for non-convexity

Example 5: Given the problem:

$$\begin{aligned} &Min \;\; f_1(x, y) = x^3 + y^2 \\ &Min \; f_2(x_1, x_2) = y^2 - 4x \end{aligned}$$

, find the Pareto-optimal set using the weights w , $1 - w$ where $w > 0$. Solution: restating the problem as:

$Min\ f(x,y) = w\left(x^3 + y^2\right) + (1-w)\left(y^2 - 4x\right)$, calculating the stability points

$$\frac{\partial f}{\partial x} = 3x^2 w - 4(1-w)$$
$$\frac{\partial f}{\partial y} = 2yw + 2y(1-w)$$

and solving the equations $\quad 3x^2 w - 4(1-w) = 0$, $\quad 2yw + 2y(1-w) = 0$

its corresponding Pareto-optimal set is : $\left\{ x = 2\sqrt{\frac{1-w}{3w}} \quad y = 0 : w > 0 \right\}$ with graph in Figure 9

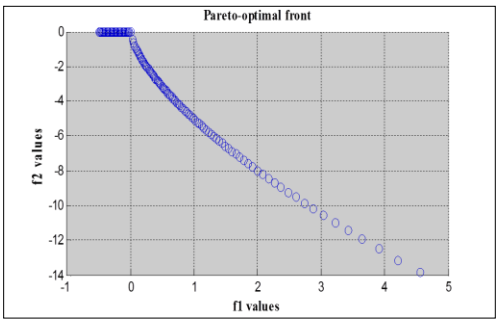

Figure 9: Pareto front for example 5

This method is the simplest way to solve a MOP. For problems having a convex Pareto-optimal front this method guarantees finding solutions on the entire Pareto-optimal set. However if the Pareto front is non-convex it cannot guarantee finding all the Pareto-optimal solutions.

2.5.2 ϵ-Constraint Method

This method alleviates the difficulties faced by the weighted sum method when solving problems with non-convex objective spaces. The method is:

$$\begin{aligned} &Minimize \quad f_\mu(x) \\ &s.t.\ f_i(x) \le \varepsilon_i \quad i = 1,\ldots,k \quad i \neq \mu \\ &\quad g_j(x) \le 0 \quad j = 1,\ldots,m \\ &\quad h_l(x) = 0 \quad l = 1,\ldots,r \quad x \in \mathbb{R}^3 \end{aligned} \tag{9}$$

For convex or non-convex objective space problems the following theorem supports the utility of the ϵ-constraint method:

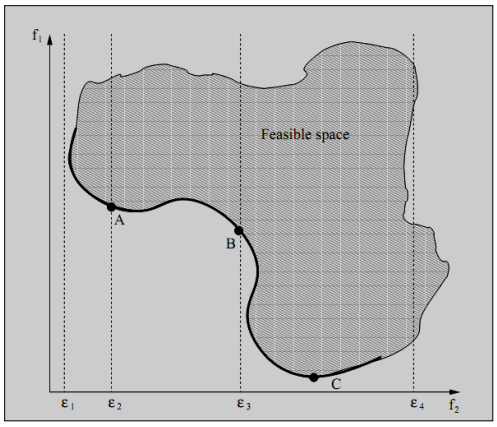

Figure 10**:** Pareto front points A, B and C for the ϵ- constraint method

Theorem 4: The unique solution of the ϵ - constraint problem stated in Equation (9) is a Pareto-optimal point for any given upper bound vector $\varepsilon = (\varepsilon_1, ..., \varepsilon_{\mu-1}, \varepsilon_{\mu+1}, ..., \varepsilon_k)$

Distinct Pareto-optimal solutions can be found by using different ε_μ values. The method can be used for problems having convex or non convex objective spaces, but has the disadvantage that solutions depend on the chosen vector ε_μ .It must be chosen in the interval

$\min f_\mu \le \varepsilon_\mu \le \max f_\mu$.

Example 6 : Find the optimal solution for the function in terms of ε_1 using Karush-Kuhn-Tucker optimality conditions:

$$Min \ \ f(x,y) = y^2 - 4x$$
$$s.t. \ \ g(x,y) = x^3 + y^2 \le \varepsilon_1$$

Solution: Let $L(x,y) = f(x,y) + \mu g(x,y)$ be the Lagrangian function associated. Let us find the stationary points of the Lagrangian function:

$$\nabla L(x,y) = \nabla f(x,y) + \mu \nabla g(x,y)$$
$$= \begin{bmatrix} -4 \\ 2y \end{bmatrix} + \mu \begin{bmatrix} 3x^2 \\ 2y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad (10)$$

Solving equation (10) we get the Lagrangian stationary points : $x = \pm \frac{2}{\sqrt{3\mu}}$, $y = 0$,

$\mu = \frac{4}{3\sqrt[3]{\varepsilon_1^2}}$ which implies that there is only one stationary point to consider: $x = \sqrt[3]{\varepsilon_1}$, $y = 0$

Let us check the point $(\sqrt[3]{\varepsilon_1}, 0)$ is candidate to be optimum points. Let us check the Lagrangian function:

$$z^T \nabla_X^2 L(\sqrt[3]{\varepsilon_1}, y0) z = \begin{bmatrix} z_1 & z_2 \end{bmatrix} \begin{bmatrix} 6\sqrt[3]{\varepsilon_1} & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = 4z_2^2 > 0 \ \ \forall \ z_2 \neq 0$$

Therefore the optimal solutions are all the points $(\sqrt[3]{\varepsilon_1}, 0)$ for any $\varepsilon_1 > 0$ value selected.

2.5.3 Weighted Metric Methods

Instead of using a weighted sum of the objectives, other means of combining multiple objectives into a single objective function can also be used. For this purpose, weighted metrics such as l_p *and* l_∞ distance metrics are used. For non-negative weights, the weighted l_p metric from the ideal solution z^* can be minimized as follows:

$$\begin{aligned} \textit{Minimize} \quad & l_p(x) = \left(\sum_{i=1}^{k} w_i \left| f_i(x) - z_i^* \right|^p \right)^{1/p} \\ \textit{s.t.} \quad & g_j(x) \leq 0 \quad j = 1, \ldots, m \\ & h_l(x) = 0 \quad l = 1, ..., r \ ; \ x \in \mathbb{R}^n \end{aligned} \tag{11}$$

For $p = 1$ $l_1(x) = \sum_{i=1}^{k} w_i \left| f_i(x) - z_i^* \right|$ the scalarized function is equivalent to the weighted sum approach.

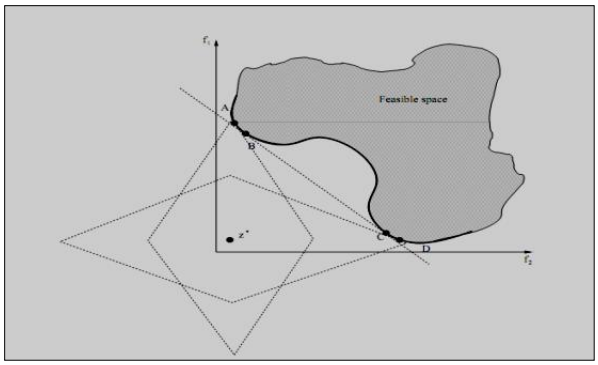

Figure 11:Weighted metric method with p = 1

For $p = 2$ $l_2(x) = \left(\sum_{i=1}^{k} w_i \left| f_i(x) - z_i^* \right|^2 \right)^{1/2}$ the scalarized function is a weighted distance to the ideal solution z^*.

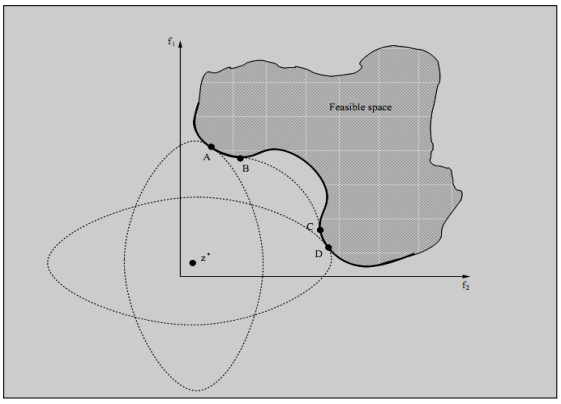

Figure 12: Weighted metric method with p = 2

Example 7: Consider the following problem:

$Min \ \ f_1(x, y) = x^3 + y^2$
$Min \ \ f_2(x, y) = 5(y^2 - x)$

Using the weighted l_2 distance metric, find the Pareto-optimal solutions corresponding to the weight vector $(w_1, w_2) = (1, 0)$.

Solution: for the weights proposed and taking $z^* = (0, 0)$ as ideal point, and restating the problem as $Min \ \ f_1(x, y) = \left(x^3 + y^2\right)^2$ *where* $(x, y) \in \mathbb{R}^2$, its critical points are:

$\left\{\left(x,\sqrt{-x^3}\right):\ x<0\right\}$ .The Hessian is $H(x,y)=\begin{bmatrix}30x^4+12xy^2 & 12xy^2\\ 12xy^2 & 4x^3+12y^2\end{bmatrix}$ therefore

$H(x,\sqrt{-x^3})=\begin{bmatrix}18x^4 & -12x^4\\ -12x^4 & -8x^3\end{bmatrix}$

since $\det\left[H_{11}(x,\sqrt{-x^3})\right]=18x^4>0 \quad for \quad x\neq 0$ and

$\det\left[H(x,\sqrt{-x^3})\right]=-144x^7(x+1)>0 \Leftrightarrow -1<x<0$ the Pareto-optimal solutions is

$\left\{\left(x,\sqrt{-x^3}\right):\ x<0\right\}$ and the corresponding Pareto-front subset is

$\left\{(f_1,f_2): f_1=x^3+(-x)^{\frac{3}{2}}\ ,\ f_2=5(-x)^{\frac{3}{2}}-5x<0 \quad for\ -1<x<0\right\}$, with plot in Figure 13

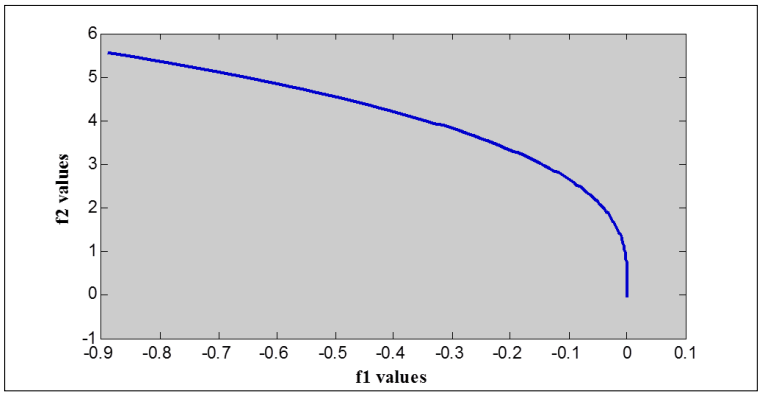

Figure 13: Pareto front subset for example 7

For $p=\infty$ $l_2(x)$ can be reduced to the Weighted Tchebycheff problem, Fig 13:

$$\begin{aligned} &Minimize\ \ l_\infty(x)=\underset{1\le i\le k}{Max}\left\{w_i\left|f_i(x)-z_i^*\right|\right\}\\ &s.t.\quad g_j(x)\le 0 \qquad j=1,\ldots,m\\ &\qquad h_l(x)=0 \qquad l=1,\ldots,r\\ &\qquad x\in\mathbb{R}^n \end{aligned} \tag{12}$$

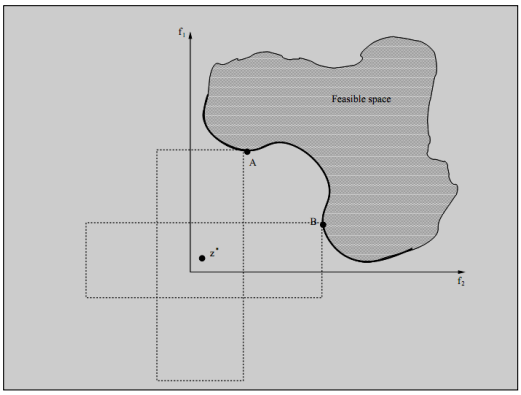

Figure 14:Weighted metric method with p = ∞

The weighted Tchebycheff method guarantees finding each and every Pareto-optimal solution when z^* is a utopian objective vector (Mietinen,1999).To apply the method, objective functions must be normalized. A negative aspect is that it requires knowledge of the maximum and minimum values of the objective functions.

2.5.4 Rotated Weighted Metric Method

Instead of directly using the l_p metric as it is stated in last section, the l_p metric can be applied with an arbitrary rotation from the ideal point. Let us assume that R is the square rotating axis matrix, therefore the system after rotation can be expressed without loss of generality equally to the original formulation.

$$\begin{aligned} &\textit{Minimize} \quad l_p(x) = \left(\sum_{i=1}^{k} w_i \left| f_i(x) - z_i^* \right|^p \right)^{1/p} \\ &\textit{s.t.} \quad g_j(x) \le 0 \quad j = 1, \ldots, m \\ &\qquad h_l(x) = 0 \quad l = 1, \ldots, r \;\; ; x \in \mathbb{R}^n \end{aligned} \tag{13}$$

By using different rotation matrices, the above functions can be minimized. For case *p*=2 the metric can be written as: $l_p(x) = \left[\left(f_i(x) - z_i^* \right)^t C \left(f_i(x) - z_i^* \right) \right]^{1/2}$

where the rotation matrix is $R = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$ and

$C = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}^T \begin{bmatrix} w & 0 \\ 0 & w_2 \end{bmatrix} \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}$.The rotation matrix will transform the objective axis into another set of axes. With this procedure it is easier to attain the sunken parts

of the Pareto-optimal front. Unfortunately, this approach has many parameters to cope with and in certain problems it could not reach dominated points so there is no guarantee of getting always Pareto-optimal points, Fig 15.

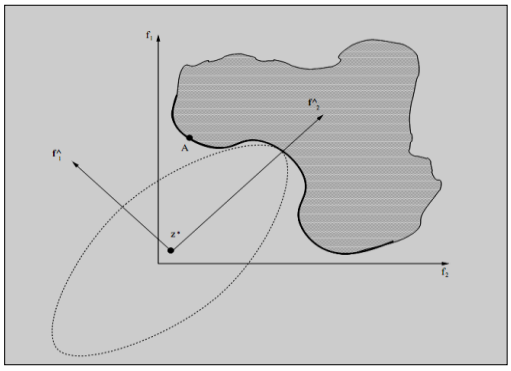

Figure 15: Rotated weight method with p = 2

2.5.5 Utility or Value Function Method

To use this method, the user provides a mathematical function $U:\mathbb{R}^m \to \mathbb{R}$ which relates all the objectives functions. The task is to maximize the utility function as follows:

$$\begin{aligned} &\textit{Maximize } U\left(f(x)\right) \\ &\quad s.t. \quad g_j(x) \le 0 \quad j = 1,\ldots,m \\ &\qquad\quad h_l(x) = 0 \quad l = 1,...,r \\ &\qquad\quad x \in \mathbb{R}^n \end{aligned} \tag{13}$$

Rosenthal (1985) stated that the utility function must be strictly decreasing before it can be used in multi-objective optimization. Miettinen (1999) proved the following theorem which makes this method meaningful, Fig 16:

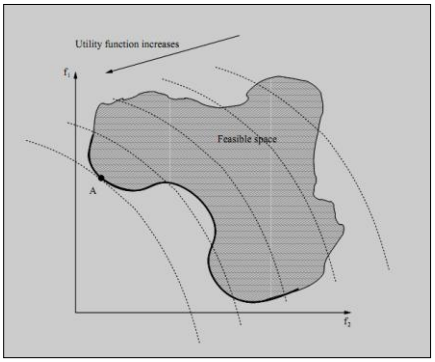

Figure 16: Contours of a utility function

Theorem 5: Let the utility function $U:\mathbb{R}^m \to \mathbb{R}$ be strictly decreasing. If U attains its maximum at $f^* = f(x^*)$.Then f^* is Pareto-optimal

This method is simple and ideal, if the user has at hand a utility function. Unfortunately, it requires users to come up with an adequate utility function and avoid using over-simplified utility functions.

Example 8: For the following two-objective problem

$$\begin{aligned} &Min\ f_1(x,y) = x^2 + y^2 \\ &Min\ f_2(x_1,x_2) = 5 + y^2 - x \\ &s.t.\quad g_1(x_1,x_2) = x + 5 \geq 0 \\ &\qquad g_2(x_1,x_2) = 5 - y \geq 0 \end{aligned}$$

use the utility function $U = 50 - f_1 - f_2$ and find the Pareto-optimal solution.

Solution: Calculate the reduced Lagrangian from the K.K.T sufficiency conditions. Since the utility function U must to maximized, without loss of generality it can be written as $F = 50 - f_1 - f_2$ for the restated problem:

$$\begin{aligned} &Max\ U = 45 - x^2 - 2y^2 + x \\ &s.t.\quad g_1(x_1,x_2) = x + 5 \geq 0 \\ &\qquad g_2(x_1,x_2) = 5 - y \geq 0 \end{aligned}$$

Which has critical points $(x^*, y^*) = (-5,5)$ *and lagragian values* $\mu_1 = 6$, $\mu_2 = 10$;

$\nabla g_1(-5,5) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ $\nabla g_2(-5,5) = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$ *are linearly independent* and the Hessian

$z^t \nabla^2_{xx} L(x,y) z = z^t \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} z = -(z_1^2 + z_2^2) < 0 \quad \forall\ (z_1, z_2) \neq (0,0)$ and $z^t \nabla g_i(-5,5) = 0$,thereby

$(x^*, y^*) = (-5,5)$ is a Pareto-optimal of the original problem.

To close this section, it is worth to mention that other very important methods are the goal programming method, the lexicographic method, the weighted goal programming method, and the min-max method, which are not described here.

One common characteristic that multi-objective optimization algorithms share is that the Pareto-optimal points search is through updating a single solution per iteration and that mainly use a deterministic transition rule to approach the optimum solution.

Every classical optimization algorithm although convergent is designed to solve a specific type of problem (Reklaitis *et al,*1983). The geometric method is designed to solve only constrained polynomial-type of objective functions, but cannot be applied easily to solve other types of problems. The conjugate gradient methods work well in problems having one optimal solution, but they are not expected to work well in problems having multiple optimal solutions. The Frank-Wolfe's successive linear programming method (Reklaitis *et al,* 1983) works efficiently on a constrained linear function, but on nonlinear objective functions its performance depends mainly on the initial conditions. Thus , one algorithm may be best suited for one class of problems but not for those problems outside that class. This requires users to know a number of optimization algorithms to solve different optimization problems.

The second general approach to solve multiple objective optimization problems is to obtain Pareto-optimal solutions when using evolutionary algorithms (EA's). Evolutionary algorithms generate sets of Pareto-optimal points or non-dominated points.

The next section presents an introduction to Evolutionary Algorithms.

2.6 Evolutionary Algorithms

In the 1960s, several researchers independently suggested adopting the principles of natural evolution, in particular Darwin's theory of the survival of the fittest, for optimization. These pioneers were Lawrence Fogel, John H. Holland, Ingo Rechenberg, and Hans-Paul Schwefel. One distinguishing feature of these so-called evolutionary algorithms (EAs) is that they work with a population of solutions. This is of particular advantage in the case of multi-objective optimization, as they can search for several Pareto optimal solutions simultaneously in one run as shown in Fig 17, providing the DM with a set of alternatives to choose from.

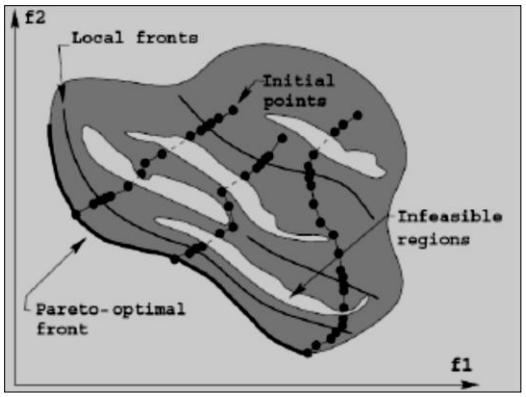

Figure 17: Evolutionary algorithm optimal search

Evolutionary algorithms mimic natural evolutionary principles in the search and optimization procedures as: evolution, reproduction, mutation and selection. Some Evolutionary Algorithms are: Genetic algorithms (GA), evolutionary expression programming (EP),Gene expression programming(GEP), evolution strategy (ES), and neuro-evolution. Some related techniques are swarm algorithms as: ant colony, bee swarm, cuckoo search etc. Analyzing their differences is a common technique to better understand the characteristics of classical optimization and evolutionary algorithms. Indeed let us start with the classical optimization methods (COM) features:

COM are classified as:

a) direct methods, and

b) gradient based methods

In direct search methods only the objective functions and constraints are used to guide the search strategy; whereas gradient methods require first and second order derivatives for the objective functions and constraints.

Common difficulties with direct and gradient methods are:

Convergence to an optimal solution depends on the chosen initial solution; most algorithms tend to get stuck to suboptimal solutions; an algorithm solving one optimization problem may not be solving a different one; every classical optimization algorithm is designed to solve a specific type of problem: geometric programming solves only polynomial problems. Most classical optimization methods use the point to point approach where only one solution gets updated to a new solution in one iteration so the advantages of parallel systems cannot be fully exploited.

On the other hand we have that :

Evolutionary algorithms (EAs) work with a population of solutions instead of a single solution; EAs do not require any auxiliary information except the objective functions values; EAs use probabilistic rules to guide their search; EAs are robust since they can be applied to a wide variety of problems, and finally as early mentioned EAs can be easily used in parallel systems. One striking difference between classical optimization methods (COM) and EAs is in their convergence theory development. Whereas in COM there is plenty of convergence results that supports it. In EAs until recently in the 2011 Yu Chen, Xiufen Zou , Weicheng Xie proved necessary and sufficient conditions on EAs convergence with Markov's chain theory.

Next the working principle of EAs is described using a Genetic Algorithm example for a can design.

Example 9: The minimization problem of a can design cost material is:

$$\min \ f(d,h) = c\left(\frac{\pi d^2}{2} + \pi dh\right)$$

$$s.t. \ \ g(d,h) = \frac{\pi d^2 h}{4} \geq 300$$

$$0 \leq d \leq 31 \ cm \ \ , 0 \leq h \leq 31 \ cm$$

To use a Genetic Algorithm, the objective function and constraint *g* must be represented as binary strings. For the sake of simplicity it is assumed that parameters *d* and *h* are integers. However GA's are not restricted to use only integer values.

Five bits are used to code each of the decision variables *d* and *h*, which makes the overall string made out of (d, h) 10 bites long as it is shown below:

If $(d, h)=(8, 6)_{10}$ = 01000 0101 where each one of the 0 and 1 values are called "genes" and the whole zeros and ones string is known as "chromosome" . Afterwards choose a $(d, h)=(8, 10)_{10}$ chromosome and calculate its fitness. In most cases the fitness is calculated with the objective function as $f(8,6)=16$. The first step of the GA is to generate randomly a population of feasible points and calculate its fitness as shown before. Assume that six chromosomes are generated and its fitness calculated as in the next table :

Table 6: Chromosomes Mating pool

Values	Chromosomes Matting Pool	Fitness $f(d,h)$
(9,11)	1001 1011	28
(13,9)	1101 1001	46
(12,6)	1100 0110	29
(5,8)	0101 1000	11
(7,4)	0111 0100	11
(6,13)	0110 1101	20

Now the reproduction or selection operator is applied to the chromosomes in Table 6.

The primary objective of the reproduction operator is to make duplicates of good solutions and eliminate bad solutions in a population, while keeping the population size constant. Some common methods for selection are: Tournament selection, proportionate selection and ranking selection. According to Goldberg, D. E.(1991) tournament selection has better or equivalent convergence and computational time complexity properties compared to any other selection operator. For this example a tournament selection is applied. Tournament selection consists on selecting at random a pair of chromosomes and compare their fitness. The best of each pair is kept in the matting poll discarding the worst and duplicating the best chromosomes in order to keep the population size. In the following table are the results of the selection:

Table 7: Mating pool after selection

Values	Chromosomes Matting Pool	Fitness $f(d,h)$
(5,8)	0101 1000	11
(7,4)	0111 0100	11
(6,13)	0110 1101	20
(5,8)	0101 1000	11
(7,4)	0111 0100	11
(6,13)	0110 1101	20

Know the crossover operator is applied picking at random a pair of strings from the matting pool and some portions of the strings are exchanged between them in order to create two new strings.

Let us select two strings and do a crossover exchanging 7 bits from the right side from each one of the chromosomes

(5, 8) 01 01 1000

(6, 13) 01 10 1101

Figure 17: Chromosomes before crossover

01 10 1000 = (6,8)

01 01 1101 = (5,13)

Figure 18: Chromosomes after crossover

A next pair is selected and crossover until a new population of the same size is obtained.

Table 8: Mating pool after crossover

Values	Chromosomes Matting Pool	Fitness $f(d,h)$
(6,8)	0110 1000	13
(7,4)	0111 0100	11
(5,13)	0101 1101	16
(6,8)	0110 1000	13
(7,4)	0111 0100	11
(5,13)	0101 1101	16

For the next step the mutation operator is applied to each one of the strings in the matting pool exchanging one bit into a zero or one and finally an stopping criterion is applied . If it is satisfied the process stop or continue starting all over as showed starting with calculating the fitness of the new population. Mutating a pair of strings is obtained:

(6,8) = 01 1[0] 1000 ⟶ 01 1[1] 1000= (7,8)

(5,13) = 01 0[1] 1101 ⟶ 01 0[0] 1101= (4,13)

Figure 18: Chromosomes before mutation

Figure 19: Chromosomes after mutation

Table 9: Mating pool after mutation

Values	Chromosomes Matting Pool	Fitness $f(d,h)$
(7,8)	0111 1000	16
(7,4)	0111 0100	11
(4,13)	0100 1101	12

The procedure could stops or the cycle starts all over if a termination condition is not satisfied.

A summary of a GA procedure is:

1. Choose a string representation scheme for the variable values,
2. Create a population of strings at random,
3. Assign a fitness value to each one of the solutions or strings,
4. Check a termination condition. If it is not satisfied go to the next step .Otherwise end the procedure
5. Modify the population solutions by the main operators : crossover, mutation, selection and go to step 4 to check the termination condition.

For one string or "off-spring" obtained after passing under a complete iteration from tits fitness value we can see that instead of improving, the value increased.

(6,8) = 01 1[0] 1000 ⟶ 01 1[1] 1000= (7,8)

Table 10: Mating pool after mutation

Values	Chromosomes Matting Pool	Fitness $f(d,h)$
(7,8)	0111 1000	16

To avoid that the fitness of the population-best solution does not deteriorate an elite-preserving operator was introduced to favor the elites of a population giving a opportunity to be carried over the next generations. Rudolph 1996 proved that GAs converge to the global optimal solution of some functions in the presence of elitism(K. Deb, Multi-objective using Evolutionary Algorithms ,Wiley 2001).

Some of the more common EAs mentioned in the literature are:

- Strength Pareto Evolutionary Algorithm (SPEA) by Zitzler and Thiele (1998)
- Strength Pareto Evolutionary Algorithm (SPEA 2) by Zitzler (2001)
- Pareto Archived Evolution Strategy (PAES) by Knowles (2000)
- Non-dominated Sorting Genetic Algorithm (NSGA II). by K. Deb (2001)
- Adaptive Pareto Algorithm (APA). Dumitrescu (2001)

At this point it is worth mentioning that the goal of this thesis is to analyze two Pareto-optimal sets. One was obtained with the Elitist Non-Dominated Sorting Genetic Algorithm II (NSGA-II) , Kalyanmoy ,Deb(2001) and the second one from the SPSS sample data bases.

In the next chapter will be shown some methods developed to the further analysis of Pareto optimal sets obtained with Evolutionary algorithms.

Chapter 3:

3 Post Pareto Optimality Methods

In chapter 2 are described two strategies to solve a MOOP: The classic and the evolutionary algorithm (EA) approaches. In the last approach, a set of non dominated points or Pareto-optimal set is obtained. A non dominated set can be formed by numerous solutions and are equally acceptable for selection, but it can be difficult for a decision maker that choice. In this chapter are presented some methods for selecting optimal solutions from a Pareto-optimal set and comment about their advantages and disadvantages.

Post-Pareto Optimality Techniques

Once a Pareto-optimal set is obtained some decision-maker considerations are used to choose a solution. Some of the following methods are frequently used in classical multi-objective over the search space.

3.1 Compromise Programming Approach:

This method picks a solution that is located at a minimum distance from a reference point or ideal point " A " ,Yu(1973), Zeleny(1973). Usually it is the vector of each of the best objective functions values $A=\left(f_1^*, f_2^*,..., f_n^*\right)$ *where* $f_i^* = \min/\max\{f_i : f_i \in Pareto \quad front\}$ where S is the search space .Some of the metrics to calculate distances are:

$$d\left(f,z\right)=\left(\sum_{i=1}^{n}\left|f_i-z_i\right|\right)^{1/p} \qquad l_p \ metric \tag{14}$$

$$d\left(f,z\right)=\max_{1\le i\le n}\frac{f_i-z_i}{\max_{x\in S}\{f_i-z_i\}} \qquad Tchebycheff \ \ metric,\ f=\left(f_1,...,f_n\right)\in Pareto\ front \tag{15}$$

Black dots on Figure 19 are optimum points.

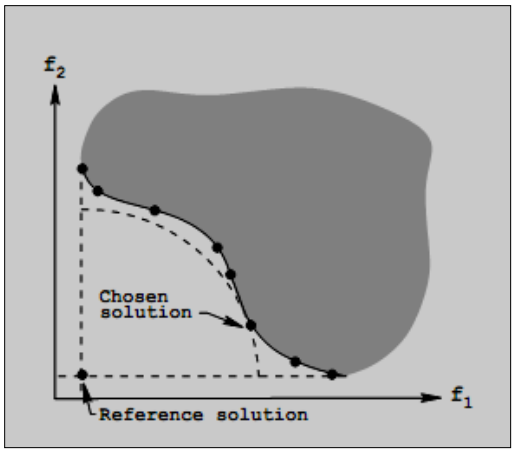

Figure 18:Ideal point and selected optimal solution

3.2 Marginal Rate of Substitution Approach

The marginal rate of substitution is the amount of improvement in one objective function which can be obtained by sacrificing an unit decrement in any other objective function, Kaisa Miettinen (1999) The solution having the maximum marginal rate of substitution or "knee point" is the one chosen by this method. Since pair-wise comparisons have to be made with all m objectives for each Pareto-Optimal solution, this method may be computational expensively Kalyanmoy, D. (2004).

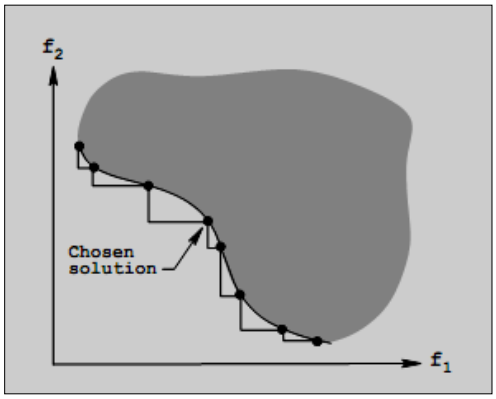

Figure 19: Chosen solution or "knee point"
with maximum marginal rate of substitution

3.3 Pseudo-Weight Vector Approach

In this approach, a pseudo-weight vector is calculated for each obtained solution.
From the Pareto-Optimal solutions obtained, the weights are calculated as:

$$w_i = \frac{\frac{\left(f_i^{\max} - f_i\right)}{\left(f_i^{\max} - f_i^{\min}\right)}}{\sum_{i=1}^{m} \frac{\left(f_i^{\max} - f_i\right)}{\left(f_i^{\max} - f_i^{\min}\right)}} \qquad \text{by definition is clear that } \sum_{i=1}^{m} w_i = 1 \qquad (16)$$

This equation calculates the relative distance of the solution from the worst value in each objective function and finds the solution, as shown in the next figure. The interesting part is that equation (16) allows a way to compute a relative trade off weight vector for a solution situated in the non convex region of the Pareto-optimal region. Once the weight vectors are calculated, a simple strategy would be to choose the solution closer to a DM preferred Kalyanmoy, D. (2004).

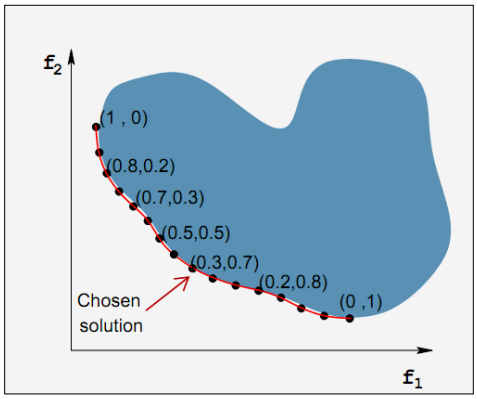

Figure 20:Chosen solution for pseudo weight approach

3.4 Non-numerical Ranking Preferences Method

This is a post-Pareto optimality method proposed and developed by Taboada *et al* (2006) to obtain a finite sequence of values for a weighting function which is used in a pruning algorithm, that reduces a Pareto-optimal set (previously obtained with an evolutionary algorithm) into an acceptable and manageable size for the decision maker**.** One example of ranking objective functions, was demonstrated by Carrillo *et al* (2011). Let us assume that, for a decision-maker the objective function $f_1(x)$ is more important than objective $f_2(x)$; objective $f_2(x)$ is more important than objective $f_3(x)$ and so on ending up with the ranking: $f_1 \succ f_2 \succ \cdots \succ f_n$. The relation "$\succ$" is a binary relation, and is called *preference relation*, or *preference order*. Weights

are generated in a numerical order $w_1 > w_2 > \cdots > w_n$ suggested by the ranked functions, and a weight function

$f(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x)$ *where* $w_1 + w_2 + \cdots + w_n = 1$ *and* $0 < w_i < 1 \; \forall i$ is created. This procedure is clearly advantageous because there is no need to provide specific weight values; the only requirement is to have ranked previously the collection of objective functions.

In a broader sense, this method is a pseudo-ranking scheme that accommodates preferences, but it is different from assigning preselected weights or utility functions. The weight values are sampled from the region shown in Fig. 21

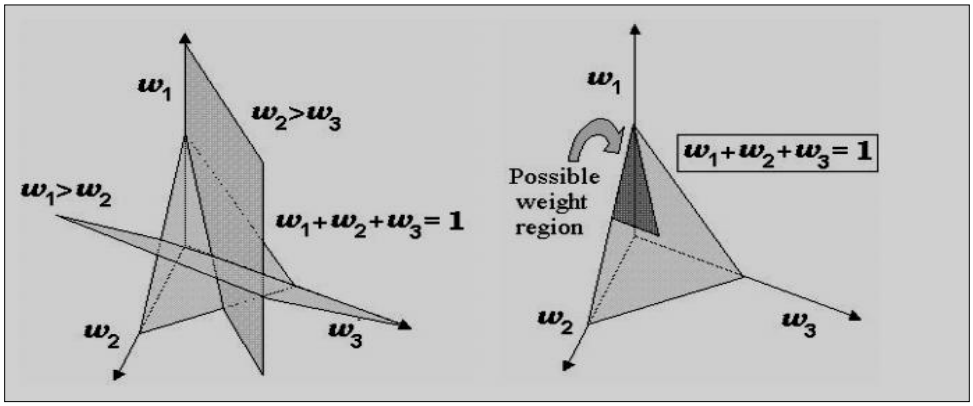

Figure 21:Weight region geometric representation

with the help of the $f_W(x)$ probability density function(p.d.f.) or also called weighting function.

An application of this technique is shown below for a multi-objective optimization problem with three component functions. Let Eq. 17 be the three dimensional p.d.f which will provide the w_i values

$$f_W(w_1,w_2,w_3)=\begin{cases}c, & w_1>w_2>w_3\\ 0, & otherwise\end{cases} \qquad where \quad w_1+w_2+w_3=1 \ and \quad 0\le w_i\le 1 \tag{17}$$

The above p.d.f. can be reduced to a two dimensional case Eq. 18 simplifying the calculations to get

$$f_W(w_1,w_2)=\begin{cases}c, & w_1>w_2>1-w_1-w_2\\ 0, & otherwise\end{cases} \tag{18}$$

its p.d.f. is Eq. 19 $$f_W(w_1,w_2)=\begin{cases}0 & ,0\le w_1<\frac{1}{3}\\ 12 & ,\frac{1}{2}-\frac{1}{2}w_1\le w_2\le w_1 \quad \& \frac{1}{3}\le w_1<\frac{1}{2}\\ 12 & ,\frac{1}{2}-\frac{1}{2}w_1\le w_2\le 1-w_1 \quad \& \frac{1}{2}\le w_1<1\end{cases} \tag{19}$$

Moreover its cumulative distribution function (c.d.f) is Eq. 20:

$$F_{w_1}(w_1)=\int_{-\infty}^{w_1} f_{w_1}(w_1)dw_1=\begin{cases}0 & , \quad 0\le w_1<\frac{1}{3}\\ 9w_1^2-6w_1+1 & , \quad \frac{1}{3}\le w_1<\frac{1}{2}\\ -3w_1^2+6w_1-2 & , \quad \frac{1}{2}\le w_1<1\end{cases} \tag{20}$$

Finally we are ready to calculate w_1 using the probability integral transformation Eq. 21

$$w_1=F_{w_1}^{-1}(u)=\begin{cases}0 & ,otherwise\\ \frac{\sqrt{u}+1}{3} & ,0\le u<\frac{1}{4}\\ 1-\sqrt{\frac{1-u}{3}} & ,\frac{1}{4}\le u<1\end{cases} \tag{21}$$

Similarly to calculate w_2 , derive and use the inverse probability transformation of its (c.d.f) to obtain Eq. 22

$$w_2=F_{w_2}^{-1}(u|w_1)=\begin{cases}0 & , \quad 0\le w_1<\frac{1}{3}\\ \frac{(3w_1-1)u+1-w_1}{2} & ,0\le u\le 1 \quad \frac{1}{3}\le w_1<\frac{1}{2}\\ \frac{(1-w_1)(u+1)}{2} & ,0\le u\le 1 \quad \frac{1}{2}\le w_1<1\end{cases} \tag{22}$$

To finalize the process solve for w_3 from the equation $w_1 + w_2 + w_3 = 1$. Once the weights are obtained, the algorithm used to prune Pareto optimal solutions is shown below:

3.4.1 Pseudo code

Step 1.Rank the objective component functions:

$$f_1 \succ f_2 \succ \cdots \succ f_n \Rightarrow w_1 > w_2 > \cdots > w_n$$

Step 2.Convert the original problem into a minimization problem and scale the $f_i's$

Step 3.Randomly generates weights based on the following scheme:

(i) randomly generate an $u \in [0,1]$

(ii) generate $w_1 = F_{w_1}^{-1}(u) = \begin{cases} 0 & ,otherwise \\ \dfrac{\sqrt{u}+1}{3} & ,0 \le u < \dfrac{1}{4} \\ 1-\sqrt{\dfrac{1-u}{3}} & ,\dfrac{1}{4} \le u < 1 \end{cases}$

(iii) then calculate $w_1 = \dfrac{\sqrt{u}+1}{3}$ if $0 \le u < \dfrac{1}{4}$ and get $w_2 = \dfrac{(3w_1 - 1)u + (1 - w_1)}{2}$;

(iv) otherwise calculate $w_1 = 1 - \sqrt{\dfrac{1-u}{3}}$ if $\dfrac{1}{4} \le u \le 1$ and get $w_2 = \dfrac{(u+1)(1-w_1)}{2}$

(v) whatever path is selected above calculate, calculate $w_3 = 1 - w_1 - w_2$

(vi) repeat all steps until *n* iterations to generate sets of weights $w_1, w_2 \, and \, w_3$.

Step 4.Sum weighted objectives to form:

$$f_{composite} = w_1 f_1 + \cdots + w_n f_n$$

Step 5.Record the best solution from the weight combination.

Step 6. Repeat Steps 2-5, 10,000 times (user defined) and the best solution for that combination is identified

The final result is a ‘‘pruned’’ Pareto optimal set. This method has been observed to achieve a 90% reduction of the entire Pareto optimal set, as reported by Coit & Baheranwala (2006).

3.5 Pruning using *k-means clustering*

This approach is more useful for users that find difficult in specifying any objective function preference. The result is a pruned Pareto set from which the decision maker only needs to consider *k* particular solutions. The *k-means clustering* algorithm is well known for its efficiency in clustering data sets. The grouping is done by calculating the centroid for each cluster, and assigning each observation to the group with the closest centroid. A recurrent problem that many clustering algorithms encounter is the choice of the number of clusters.
In order to determine the optimal number of clusters, k, Rousseeuw (1987), Rousseeuw *et al*(1989) suggested a cluster validity technique, the silhouette plot, to evaluate the quality of a clustering allocation, independently of the clustering technique that is used. An approach to prune Pareto-optimal set was proposed by Taboada and Coit (2006) using this technique.

Proposed Approach:

1. Obtain the entire Pareto-optimal set or sub-set of solutions by using a multiple-objective evolutionary algorithm (MOEA) or by another means.
2. Apply the *k*-means algorithm to form clusters on the solutions contained in the Pareto set.
3. To determine the "optimal" number of clusters, *k*, in this set, silhouette plots are used. A value of the silhouette width, $s(i)$, is obtained for several values of *k*. The clustering with the highest average silhouette width is selected as the "optimal" number of clusters in the Pareto-optimal set.
4. For each cluster, select a representative solution. To do this, the solution that is closest to its respective cluster centroid is chosen as a good representative solution.
5. Analyze the results. At this point, the decision-maker can either:
5.1 Analyze the "knee" cluster. The suggestion is to focus on the cluster that has solutions that conform to the "knee" region. The "knee" is formed by those solutions of the Pareto-optimal front where a small improvement in one objective would lead to a large deterioration in at least one other objective. Moreover, from this "knee" cluster the decision maker can select a promising solution for system implementation. This would be the solution closest to the ideal, or utopian solution of the multiple objective problem, in a standardized space.
5.2 Analyze the *k* representative solutions and/or select the most promising solutions among this *k* set, selecting the solution closest to the ideal point.

An advantage of this approach is that the user does not have previously selected the number of cluster since the silhouette plot gives a number of clusters to be selected. It reduced in a 96% a Pareto optimal set; nevertheless more tests must be done to make sure such rate pattern continues.

3.6 Greedy Reduction (GR) algorithm

The Greedy Reduction (GR) algorithm is useful for obtaining subsets of large, Pareto-optimal sets. Selection of these subsets is based on maximizing a scalarizing function of the vector of percentile ordinal rankings of the Pareto optima within the larger set. Nevertheless, if the percentile vectors are non-dominated, the Greedy algorithm is not always optimal. On the other hand, the GR algorithm executes in linear time in the worst case.

3.7 Local Search with ASF (Achievement Scalarizing function)

Padhye, *et al (*2009), proposed a mutation driven, hill climbing *local search* to refine the solutions from a Pareto-optimal set previously obtained with a evolutionary algorithm. Local search usually considers a non-dominated solution and tries to improve it by utilizing a construction of some single objective function. In their study of the following achievement scalarizing function (ASF) was used and considered a starting point $z = f(y)$ usually known as a reference point for local search.

$$\min_{x \in S \subset \mathbb{R}^n} \max_{i=1}^{M} \frac{f_i - z_i}{f_i^{\max} - f_i^{\min}} + \rho \sum_{j=1}^{M} \frac{f_j - z_j}{f_j^{\max} - f_j^{\min}} \qquad (23)$$

By minimizing ASF solutions are projected on the Pareto-front and convergence can be guaranteed. Improvement was obtained, assuring that the solutions found are good estimates for true Pareto-solutions. Improvement was obtained, assuring that the solutions found are good estimates for true Pareto-solutions.

3.8 A Two Stage Approach Pareto Analysis

In 2008, Li *et al* proposed a two stage approach for selecting optimal solutions from a Pareto optimal set, previously obtained with any of the two NSGA or NSGA-II evolutionary algorithms. At the first stage, Pareto-optimal solutions are classified into several clusters by applying an artificial neural network method called self-organizing map (SOM). In the second stage, non-efficient solutions are eliminated from each cluster, and representative efficient solutions are identified through the data envelopment analysis (DEA) method, which is a special multiple objective selection optimization (MOSO) approach. The self organizing map, is a clustering method that has several advantages, compared to the k-means classification method.
Unlike the k-means method that only minimizes the mean squared error, in terms of the Euclidian distance, the SOM approach measures the similarity by the Euclidian distance as well as the angle between the input vectors by updating the weight vectors iteratively . Such training process results in the topological preservation from the input vectors to the output lattice map. Because of those advantages, SOM is utilized in the proposed two-stage approach to classify a Pareto optimal solution set. Once the Pareto set has been clustered with SOM, the data envelopment method (DEA) is applied to reduce the size of each cluster to a manageable collection of optimal points for the decision maker. DEA is a linear programming technique for measuring the relative performance of the decision making units. In a Multi-objective optimization problem, the decision variables can be considered as decision making units. Such performance is measured with relative efficiency formula:

$$RE = \frac{weighted\ \ sum\ of\ \ inputs}{weighted\ \ sum\ of\ \ outputs} \tag{24}$$

For a problem involving *l* decision making units, each of which has *m* inputs and *n* outputs, the relative efficiency of the *k*th DMU can be expressed as:

$$RE_k = \frac{\sum_{j=1}^{n} u_j y_{jk}}{\sum_{i=1}^{m} v_i x_{ik}} \quad k = 1, 2, ..., l \ and \ \ u_j, v_i \geq 0 \tag{25}$$

where u_j and v_i are the weights for the outputs and inputs respectively .
The strategy proposed for pruning the clusters obtained in the first stage is maximizing the relative efficiency of the DMU's as shown in formula(26):

$$\max RE_{k_0} = \sum_{j=1}^{n} u_j y_{jk_0}$$

$$s.t \quad \sum_{i=1}^{m} v_i x_{ik} = 1, \qquad (26)$$

$$\sum_{j=1}^{n} u_j y_{jk_0} - \sum_{i=1}^{m} v_i x_{ik} \le 1$$

$$u_j, v_i \ge \varepsilon > 0, \; i = 1,2,..,m \quad j = 1,2,...,n$$

ε is a small positive value. To obtain the performance of the entire set of variables, it is necessary to solve n times the above linear problem. Finally, the best DMU's are those with relative efficiency equal to one, and the Pareto optimal vectors with relative efficiency equal to one for all its entries are the selected optimal solutions from the clustered Pareto-optimal set by the SOM method.

3.9 A Clustering Method Based on Dynamic Self Organizing Trees for Post-Pareto Optimality Analysis

This algorithm offers two main advantages: there is no need to provide an initial number of clusters, and at each hierarchical level**,** the algorithm optimizes the number of clusters, and can reassign data from previous hierarchical levels in order to rearrange missed clustered data. The clustering method is the dynamically growing self organizing tree (DGSOT) algorithm. DGSOT is a hierarchical clustering method than has some advantages over other well-known clustering methods. The DGSOT algorithm constructs a hierarchical tree from top to bottom by division. At each hierarchical level, the algorithm optimizes the number of clusters, and can reassign data from previous hierarchical levels in order to rearrange misclustered data. Each leaf of the tree represents a cluster, each cluster is a subset of non-dominated solutions from the original set of solutions. Therefore, each leaf in the final tree is a non-dominated solution.
Hierarchical algorithms find successive clusters using previously established clusters. These algorithms usually are either agglomerative ("bottom-up") or divisive ("top-down"). Agglomerative algorithms begin with each element as a separate cluster and merge the element into successively larger clusters. Divisive algorithms begin with the whole set, and proceed to divide it into successively smaller clusters (like DGSOT).A more detailed description of these methods is presented by Fung (2001). Some of the most important disadvantages of both methods are presented in Table 9.

Table 9: Disadvantages for *k*-means and hierarchical algorithms

K-Means	Hierarchical clustering
Number of *k* clusters have to be predefined	It does not provide a discrete number of clusters. Clusters have to be defined with cut-offs
Fixed number of clusters can make it difficult to predict what *k* should be	Cannot return to previously hierarchical level to reassign misclustered data
Different initial partitions can result in different final clusters. It is helpful to rerun the program using different values of *k* to compare results	Selection of split points is critical

3.10 Visualization technique for Pareto Front

Efficient visualization techniques for Pareto Front and Set analyses are needed for helping decision makers in the selection task. The Pareto front supplies a set of solutions where the designer has to look for the best choice according to his preferences. Visualization techniques often play a key role in helping DM, but they have important restrictions for more than two-dimensional Pareto fronts. A graphical representation, called Level Diagrams, for n-dimensional Pareto front analysis was proposed by Blasco *et al.* (2008). Level Diagrams consists of representing each objective and design parameter on separate diagrams. This new technique is based on two key points: classification of Pareto front points according to their proximity to ideal points, measured with a specific norm of normalized objectives (several norms can be used); and synchronization of objective and parameter diagrams. Level Diagrams can be colored, so establishing a visual representation of preferences that can help the decision-maker.

The Level Diagrams tool is based on the classification of the Pareto front approximation according to the proximity to the ideal point. For this classification a norm is applied to evaluate the distance to the ideal point. Different norms could be used to obtain different characteristics of the diagrams, the most common are (a) Block Norm : $\|f(x)\|_1 = \sum_{i=1}^{n} |f_i(x)|$ (b) Euclidean Norm : $\|f(x)\|_2 = \sum_{i=1}^{n} \{f_i(x)\}^2$ and (c)

Maximum Norm : $\|f(x)\|_\infty = max\{f_i(x) | i = 1, \dots, n\}$.Each norm gives a different point of view of the Pareto front shape, for instance: Euclidean norm supply an accurate evaluation of the conventional geometrical distance to the ideal point, and then offer a better view of the true shape. Maximum norm can supply information about the worst objective for a specific point, and is useful for trade-off analysis between different objectives. An increment in this norm directly reveals a worsening of at least one of the objectives. To plot Level diagrams, the points

of the Pareto front are sorted in ascending order of the value of the norm of the objective function vector $\|f(x)\|_x$ whatever norm is used. Afterwards $f_i(x)$ and decision variable x_j has are graphed against $\|f(x)\|_x$. On the vertical axis $\|f(x)\|_x$ is drawn vs $f_i(x)$ on the horizontal axis. The same procedure is used to graph $\|f(x)\|_x$ vs x_j .

The fundamental idea is classification by layers, and synchronous representation of all objectives and parameters. It is shown that this Level Diagrams representation enables a good analysis of the Pareto front, and so provides an excellent tool to help decision-making.

Chapter 4:

4. Non-Numerical Ranking Preferences Method

In chapter three, several methods to analyze Pareto-optimal sets were presented, and discussed its advantages and disadvantages. In this chapter the fourth method described in the previous chapter as the non-numerical ranking preferences method, will be generalized for any number of weights for a multi-objective optimization problem with n objective functions and demonstrated with an example. The method proposed in this chapter Taboada,et al (2007), is based on a non-numerical ranking of the objective functions based on their relative importance. The strength of this method is precisely that the decision-maker only ranks non-numerically (in order of relative importance) the objective functions but does not have to select specific weight values. The pruning method uses threshold boundaries to help the decision-maker selecting in a flexible way solutions that reflect his/her preferences. In the next section are explained all the details of the proposed algorithm and how the pruning of solution is performed.

4.1 Generalization to *n* weights

In section 3.4 was shown the procedure to obtain the algorithm for the three weights case Carrillo,*et al* (2011) , based on the weight function besides of a series of conditional cumulative distribution functions developed for such goal. Since this is an iterative procedure, the general method proposed by Carrillo,Taboada. (2012) for the Non-Numerical Ranking Preferences Method is:

Iterative method for n weights case

1. Find the normalization constant c for the probability density function Eq. 27

$$f(w_1,\ldots,w_n)=\begin{cases} c, & w_1 > w_2 > \cdots > w_n \\ 0, & otherwise \end{cases} \qquad where \ \sum_{i=1}^{n} w_i = 1 \ and \ \ 0 < w_i < 1 \ for \ all \ i \qquad (27)$$

$$W=(w_1,\ldots,w_n)$$

2. Get the marginal p.d.f. for $f(w_n)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} c\, dw_1 \cdots dw_{n-1}$ (28)

3. Obtain its corresponding cumulative distribution function

$F_n(w_n) = \int_{-\infty}^{w_n} f_{w_n}(s)ds$ and solve for w_n in $u = F_n(w_n)$

4. To calculate w_{n-1} obtain its conditional probability density function

$f(w_{n-1}|w_n) = \frac{f(w_{n-1}, w_n)}{f(w_n)}$ and from its conditional cumulative distribution function(c.d.f.)

$F(w_{n-1}|w_n)$ solve for $w_{n-1} = F^{-1}(u|w_n)$, where

$u = F(w_{n-1}|w_n)$ *is uniformly distributed on* (0,1)

5. Continue up to $f(w_2|w_3 \cdots w_n) = \frac{f(w_2, w_4, \ldots, w_n)}{f(w_3, \ldots, w_n)}$ and from its conditional

$F(w_2|w_3 \cdots w_n)$ solve for $w_2 = F^{-1}(u \mid w_3 \cdots w_n)$ where

$u = F(w_2|w_3 \cdots w_n)$ *is uniformly distributed in* (0,1)

6. Finally solve for $w_1 = 1 - \sum_{i=2}^{n} w_i$

Remark: The integration order to obtain the weights is determined by the integration region given by the constraint $w_1 > w_2 > \cdots > w_n$, $\sum_{i=1}^{n} w_i = 1$ *and* $0 < w_i < 1$ *for all* i . Since this region can be restated in several algebraic forms, the number of constants C for each of the p.d.f. in Equation (27) are as many as algebraic forms for the constraint region . Therefore probability density functions are not unique. Moreover, the constant C must have a value such that the integral probability theorem can be applied to solve for all of the weights values searched. Hence it is very unlikely obtaining a general algorithm to produce any desired number of weights. Thus for each case must be developed an algorithm as is shown in the next section for the case of five weights.

4.2 Development of a non-numerical ranking five weights algorithm.

Similarly to the cases three and four, for the five weights case, a probability density function (p.d.f.) is developed, but the order of the weights are obtained varies depending on the integration region form as mentioned before, as is shown below.

Let be a probability density function for the five weights to be obtained as shown in Eq. 29:

$$f(W)=\begin{cases} c, & w_1>w_2>w_3>w_4>w_5 \\ 0, & otherwise \end{cases} \quad p.d.f$$
$$where \quad w_1+w_2+w_3+w_4+w_5=1 \ and \quad 0\le w_i\le 1 \qquad (29)$$
$$and \quad W=(w_1,w_2,w_3,w_4,w_5)$$

One solution for the set

$$w_1>w_2>w_3>w_4>w_5$$
$$where \quad w_1+w_2+w_3+w_4+w_5=1 \ and \quad 1>w_1>0$$

is as shown in Eq. 30

$$0<w_5<\frac{1}{5} \quad ; w_5<w_4<\frac{1-w_5}{4}$$
$$\frac{1}{2}(1-2w_4-w_5)\le w_1<1-3w_4-w_5$$
$$\frac{1}{2}(1-w_1-w_4-w_5)<w_2<1-w_1-2w_4-w_5 \qquad (30)$$
$$w_3=1-w_1-w_2-w_4-w_5$$

To find the normalization constant c let us calculate the multiple integral

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_W(w_1,w_2,w_4,w_5)dw_1dw_2dw_4dw_5 =$$
$$\int_0^{1/5}\int_{w_5}^{\frac{1-w_5}{4}}\int_{\frac{1}{2}(1-2w_4-w_5)}^{1-3w_4-w_5}\int_{\frac{1}{2}(1-w_1-w_4-w_5)}^{1-w_1-2w_4-w_5} 1\,dw_2dw_1dw_4dw_5=\frac{1}{3840}$$
$$Thus \quad c=3840$$

Thus its probability density function can be written as Eq. 31:

$$f(w_1,w_2,w_4,w_5)=\begin{cases} & 0<w_5<\frac{1}{5}\ , \quad w_5<w_4<\frac{1-w_5}{4} \\ 3840, & \frac{1}{2}(1-2w_4-w_5)\le w_1<1-3w_4-w_5 \\ & \frac{1}{2}(1-w_1-w_4-w_5)<w_2<1-w_1-2w_4-w_5 \\ 0, & otherwise \end{cases} \qquad (31)$$

According to the integration region obtained to generate the weights one must start at w_5

The marginal $p.d.f.$ of f_{w_5} is $f_{w_5}(w_5) = \int_{w_5}^{\frac{1-w_5}{4}} \int_{\frac{1}{2}(1-2w_4-w_5)}^{1-3w_4-w_5} \int_{\frac{1}{2}(1-w_1-w_4-w_5)}^{1-w_1-2w_4-w_5} 3840\, dw_2 dw_1 dw_4$

The corresponding p.d.f. is Eq. 32 :

$$f_{w_5}(w_5) = \begin{cases} 20(1-5w_5)^3 & ,0 < w_5 < \frac{1}{5} \\ 0 & ,otherwise \end{cases} \tag{32}$$

And its corresponding *c.d.f* is Eq.33:

$$F_{w_5}(w_5) = \int_{-\infty}^{w_5} f_{w_5}(s)ds = \begin{cases} 1-(1-5w_5)^4 & ,0 < w_1 < \frac{1}{5} \\ 0 & ,otherwise \end{cases} \tag{33}$$

Solving for w_5 by the inverse probability transformation theorem :

$$w_5 = \frac{1-(1-u)^{\frac{1}{4}}}{5} = F_{w_5}^{-1}(u) \quad where\ u\ is\ a\ uniform\ random\ variable\ U \sim (0,1)$$

To generate w_4 $f(w_4, w_5) = 240(4w + m - 1)^2$ was derived.

To calculate w_4 we need to obtain its conditional p.d.f given w_5 and finally get its cumulative distribution function Eq. 34

$$f(w_4|w_5) = \frac{f(w_4, w_5)}{f(w_5)} \quad which\ is$$

$$f(w_4|w_5) = \begin{cases} \frac{12(4w_4 + w_5 - 1)^2}{(1-5w_5)^3} & ,0 < w_5 < \frac{1}{5}\ , \quad w_5 < w_4 < \frac{1-w_5}{4} \\ 0 & ,otherwise \end{cases} \tag{34}$$

The w_4 cumulative distribution given that w_5 is Eq. 35:

$$F(w_4|w_5) = \begin{cases} 1 + \frac{(4w_4 + w_5 - 1)^3}{(1-5w_5)^3} & ,0 < w_5 < \frac{1}{5} \\ & w_5 < w_4 < \frac{1-w_5}{4} \\ 0 & ,otherwise \end{cases} \tag{35}$$

Thus $w_4 = \dfrac{\sqrt[3]{u-1}\left(1-5w_5\right)+1-w_5}{4} = F^{-1}_{w_4|w_5}\left(u\right)$ *where u is a uniform random variable* $U \sim \left(0,1\right)$

To calculate w_1 we need to obtain its conditional p.d.f given w_5 *and* w_4 Eq. 36

$$f\left(w_1|w_5,w_4\right) = \frac{f\left(w_1,w_4,w_5\right)}{f\left(w_4,w_5\right)}$$

$$f\left(w_1,w_4,w_5\right) = \begin{cases} 1920\left(1-w_1 3w_4-w_5\right) & ,0<w_5<\dfrac{1}{5}\ , \qquad w_5<w_4<\dfrac{1-w_5}{4} \\ & \dfrac{1}{2}\left(1-2w_4-w_5\right)\le w_1<1-3w_4-w_5 \\ 0 & ,otherwise \end{cases} \tag{36}$$

$$f\left(w_1|w_4,w_5\right) = \begin{cases} \dfrac{8\left(1-w_1 3w_4-w_5\right)}{\left(4w+m-1\right)^2} & ,0<w_5<\dfrac{1}{5}\ , \qquad w_5<w_4<\dfrac{1-w_5}{4} \\ & \dfrac{1}{2}\left(1-2w_4-w_5\right)\le w_1<1-3w_4-w_5 \\ 0 & ,otherwise \end{cases}$$

The corresponding cumulative distribution function is Eq. 37:

$$F_{w_1}\left(w_1|w_4,w_5\right) = \begin{cases} 1-\dfrac{\left(1-w_1-3w_4-w_5\right)^2}{\left(4w_4+w_5-1\right)^2} & ,0<w_5<\dfrac{1}{5}\ , \qquad w_5<w_4<\dfrac{1-w_5}{4} \\ & \dfrac{1}{2}\left(1-2w_4-w_5\right)\le w_1<1-3w_4-w_5 \\ 0 & ,otherwise \end{cases} \tag{37}$$

By the inverse probability transformation $u = F_{w_1}\left(w_1|w_4,w_5\right)$ is a uniform random variable on $\left(0,1\right)$, thus solving for w_1 we obtain Eq. (38)

$$w_1 = F^{-1}_{w_1|w_4,w_5}\left(u\right) = 1-w_5-3w_4-\left(4w_4+w_5-1\right)\sqrt{1-u} \tag{38}$$

To find w_2 the following conditional probability density function is calculated Eq. 39:

$$f\left(w_2|w_1,w_4,w_5\right) = \frac{f\left(w_1,w_2,w_4,w_5\right)}{f\left(w_1,w_4,w_5\right)} \tag{39}$$

obtaining the conditional p.d.f. as in Eq. 40

$$f\left(w_2 \middle| w_1, w_4, w_5\right) = \begin{cases} & , 0 < w_5 < \frac{1}{5} \ , \qquad w_5 < w_4 < \frac{1-w_5}{4} \\ \frac{2}{\left(1-w_1-w_5-3w_4\right)} & \frac{1}{2}\left(1-2w_4-w_5\right) \le w_1 < 1-3w_4-w_5 \\ & \frac{1}{2}\left(1-w_1-w_4-w_5\right) < w_2 < 1-w_1-2w_4-w_5 \\ 0 & , otherwise \end{cases} \tag{40}$$

Finally its conditional c.d.f. is Eq. 41

$$F\left(w_2 \middle| w_1, w_4, w_5\right) = \int_{\frac{1}{2}\left(1-w_1-w_4-w_5\right)}^{y} \frac{2ds}{\left(1-w_1-w_5-3w_4\right)}$$

$$\begin{cases} & 0 < w_5 < \frac{1}{5} \ , \qquad w_5 < w_4 < \frac{1-w_5}{4} \\ \frac{1-w_1-2w_2-w_4-w_5}{w_1+3w_4+w_5-1} & \frac{1}{2}\left(1-2w_4-w_5\right) \le w_1 < 1-3w_4-w_5 \\ & \frac{1}{2}\left(1-w_1-w_4-w_5\right) < w_2 < 1-w_1-2w_4-w_5 \\ 0 & , otherwise \end{cases} \tag{41}$$

Solving for w_2 in the equation (42)

$$u = \frac{1-w_1-2w_2-w_4-w_5}{w_1+3w_4+w_5-1} \tag{42}$$

$$obtaining: w_2 = \frac{\left(u+1\right)\left(1-w_1-w_4-w_5\right)-2uw_4}{2}$$

obtaining the last weight from the equation : $w_3 = 1-w_1-w_2-w_4-w_5$.

Finally the five weights algorithm can be written in the following pseudo code:

4.2.1 Non Numerical ranking five weights pseudo code

To obtain the $w_1, w_2, w_3, w_4 \, and \, w_5$ values follow the steps:

1) Randomly generate an $u \in (0,1)$

2) Compute $w_5 = \frac{1-(1-u)^{\frac{1}{4}}}{5}$

3) Randomly generate another $u \in (0,1)$

compute $w_4 = \frac{\sqrt[3]{u-1}(1-5w_5)^3 - m + 1}{4}$

4) Randomly generate another $u \in (0,1)$

compute $w_1 = 1 - w_5 - 3w_4 - (4w_4 + w_5 - 1)\sqrt{1-u}$

5) Randomly generate another $u \in (0,1)$

Calculate $w_2 = \frac{(u+1)(1-w_1-w_4-w_5) - 2uw_4}{2}$

6) Calculate $w_3 = 1 - w_1 - w_2 - w_4 - w_5$

7) Repeat all steps until n iterations to generate a n weights set of w_1, w_2, w_3, w_4 *and* w_5 .

Example 4.1

The Pareto set to test the preferences algorithm was obtained from car-sales file provided by the SPSS software program; which consist of 115 brand cars with eight features for each one as: price, engine size, horse-power, fuel capacity and miles per gallon, etc. The original file was reduced to 42 records of non-dominates values or Pareto optimal values ,which were pruned using three threshold values. Data was reduced choosing the upper bounded f values by each one of the alpha threshold values, *i.e.* $f_{composite} = w_1 f_1(x) + \cdots + w_k f_k(x) < \alpha \quad threshold$. The results obtained with the five weights algorithm are shown in tables 10 , 11 and 12.

Table 10: Reduced Pareto set with 5 weight and α=0.4

car	pruned for alpha=0.4		price	engines	horsepw	fuelcap	mpg
19	Chevrolet	Malibu	16.535	3.1	170	15	25
25	Chevrolet	Metro	9.235	1	55	10.3	45
29	Chrysler	Concorde	22.245	2.7	200	17	26
36	Dodge	Intrepid	22.505	2.7	202	17	25
40	Ford	Contour	17.035	2.5	170	15	25
44	Honda	Civic	12.885	1.6	106	11.9	32
46	Hyundai	Accent	9.699	1.5	92	11.9	31
47	Hyundai	Elantra	11.799	2	140	14.5	27
48	Hyundai	Sonata	14.999	2.4	148	17.2	25
58	Mitsubishi	Eclipse	19.047	2.4	154	15.9	24
74	Nissan	Sentra	20.39	2.4	155	15.9	25
75	Nissan	Altima	26.249	3	222	18.5	25
79	Oldsmobile	Alero	18.27	2.4	150	15	27
94	Saturn	SL	10.685	1.9	100	12.1	33
100	Toyota	Corolla	13.108	1.8	120	13.2	33
103	Toyota	Celica	16.875	1.8	140	14.5	31
106	Volkswagen	Passat	21.2	1.8	150	16.4	27
110	Volvo	S40	23.4	1.9	160	15.8	25

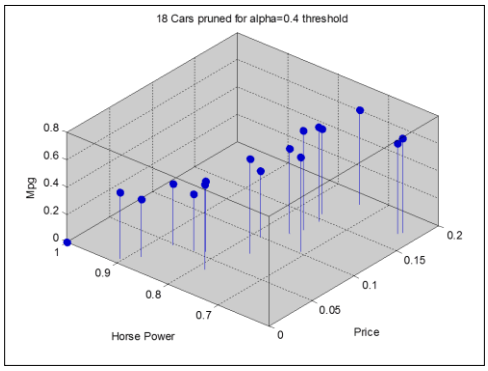

Figure 22: Graph of pruned values in table 10

Table 11: Reduced Pareto set with 5 weight and α=0.3

car	pruned for alpha=0.3		price	engines	horsepw	fuelcap	mpg
25	Chevrolet	Metro	9.235	1	55	10.3	45
44	Honda	Civic	12.885	1.6	106	11.9	32
46	Hyundai	Accent	9.699	1.5	92	11.9	31
74	Nissan	Sentra	13.499	1.8	126	13.2	30
94	Saturn	SL	10.685	1.9	100	12.1	33
100	Toyota	Corolla	13.108	1.8	120	13.2	33

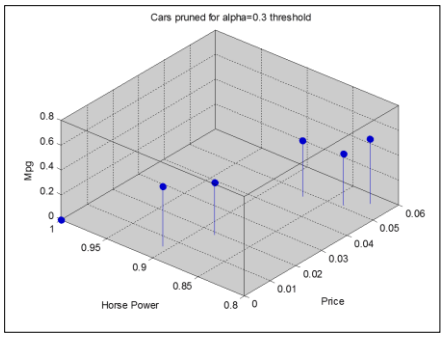

Figure 21: Graph of pruned values in table 11

Table 12: Reduced Pareto set with 5 weight and α=0.2

car	pruned for alpha=0.2		price	engines	horsepw	fuelcap	mpg
25	Chevrolet	Metro	9.235	1	55	10.3	45

According to the results showed in table 1 the optimum solution is for the cheapest car and most gas saving among all. For the second pruning there are several options for the decision maker to select; but for the last one for a 0.4 threshold the selection includes expensive an cheap automobiles as well ,this gives a wide range to decide and select.

4.3 Development of a non-numerical ranking seven weights algorithm

In a similar form as in the previous section, a probability density function for seven weights case is derived and demonstrated the weighting algorithm generator. Let be $f(W)$ the probability density function we are interested on stated in Eq. 43.

$$f(W)=\begin{cases} c, & w_1 > w_2 > w_3 > w_4 > w_5 > w_6 > w_7 \quad p.d.f \\ 0, & otherwise \end{cases} \tag{43}$$

$$where \quad w_1+w_2+w_3+w_4+w_5+w_6+w_7=1 \quad and \quad 0<w_i<1; \quad W=(w_1,w_2,w_3,w_4,w_5,w_6,w_7)$$

One solution for the set $w_1 > w_2 > w_3 > w_4 > w_5 > w_6 > w_7$ *where* $\sum_{i=1}^{6} w_i = 1$ is the region formed by the inequalities shown below in Eq. 44 :

$$\frac{1}{7} < w_1 \le \frac{1}{6} \;;\; \frac{1-w_1}{6} < w_2 < w_1; \frac{1}{4}\left(1-w_1-w_2\right) < w_3 < w_2$$
$$\frac{1}{4}\left(1-w_1-w_2-w_3\right) < w_4 < w_3 \;; \frac{1}{3}\left(1-w_1-w_2-w_3-w_4\right) < w_5 < w_4 \qquad (44)$$
$$\frac{1}{2}\left(1-w_1-w_2-w_3-w_4-w_5\right) < w_6 < w_5$$
$$w_7 = 1-w_1-w_2-w_3-w_4-w_5-w_6$$

To find the normalization constant c we calculate

Therefore the corresponding p.d.f can be written as Eq. 45:

$$f\left(w_1,w_2,w_4,w_5,w_6\right) = \begin{cases} & \frac{1}{7} < w_1 \le \frac{1}{6} \;;\; \frac{1-w_1}{6} < w_2 < w_1; \frac{1}{4}\left(1-w_1-w_2\right) < w_3 < w_2 \\ c, & \frac{1}{4}\left(1-w_1-w_2-w_3\right) < w_4 < w_3 \;; \frac{1}{3}\left(1-w_1-w_2-w_3-w_4\right) < w_5 < w_4 \\ & \frac{1}{2}\left(1-w_1-w_2-w_3-w_4-w_5\right) < w_6 < w_5 \\ 0, & otherwise \end{cases} \qquad (45)$$

$$\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_W\left(W\right) dw_6 dw_5 dw_4 dw_3 dw_2 dw_1 =$$

$$\int_{1/7}^{1/6}\int_{\frac{1}{6}(1-w_1)}^{w_1}\int_{\frac{1}{5}(1-w_1-w_2)}^{w_2}\int_{\frac{1}{4}(1-w_1-w_2-w_3)}^{w_3}\int_{\frac{1}{3}(1-w_1-w_2-w_3-w_4)}^{w_4}\int_{\frac{1}{2}(1-w_1-w_2-w_3-w_4-w_5)}^{w_5} c\; dw_6 dw_5 dw_4 dw_3 dw_2 dw_1 = \frac{c}{90720(6)^6} = 1$$

Thus $c = 90720(6)^6$

To generate all of the weights we will start with :

The marginal $p.d.f.$ *of* f_{w_1} *is*

$$f\left(w_1\right) = \int_{\frac{1}{6}(1-w_1)}^{w_1}\int_{\frac{1}{5}(1-w_1-w_2)}^{w_2}\int_{\frac{1}{4}(1-w_1-w_2-w_3)}^{w_3}\int_{\frac{1}{3}(1-w_1-w_2-w_3-w_4)}^{w_4}\int_{\frac{1}{2}(1-w_1-w_2-w_3-w_4-w_5)}^{w_5} c\; dw_6 dw_5 dw_4 dw_3 dw_2$$

The $p.d.f.$ *is*:

$$f\left(w_1\right) = \begin{cases} \frac{c}{2160}\left(7w_1-1\right)^5 & ,\frac{1}{7} \le w_1 < \frac{1}{6} \\ 0 & ,otherwise \end{cases} \qquad (46)$$

And its corresponding c.d.f is Eq. 47

$$F(w_1)=\int_{-\infty}^{w_1} f_{w_1}(s)ds=\begin{cases}\dfrac{c}{6(7)(2160)}(7w_1-1)^6 & ,\dfrac{1}{7}\le w_1<\dfrac{1}{6}\\ 0 & ,otherwise\end{cases} \tag{47}$$

By the probability Integral transformation theorem:
"$F(w_1)=u$ *is a uniform random variable* $U\sim(0,1)$".

Solving for w_1 Eq. 48:

$$w_1=\frac{6+\sqrt[6]{u}}{42}=F_{w_1}^{-1}(u) \tag{48}$$

To generate w_2 $f(w_1,w_2)=\dfrac{c(6w_2+w_1-1)^4}{360}$ was derived.

To calculate w_6 we need to obtain its conditional p.d.f given w_6 Eq. 49

$$f(w_2|w_1)=\frac{f(w_1,w_2)}{f(w_1)} \quad which \quad is$$

$$f(w_2|w_1)=\begin{cases}\dfrac{c(6w_2+w_1-1)^4}{(7w_1-1)^5} & ,\dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1\\ 0 & ,otherwise\end{cases} \tag{49}$$

The w_2 cumulative distribution given that w_1 is Eq. 50:

$$F(w_2|w_1)=\begin{cases}\dfrac{(6w_2+w_1-1)^5}{5(7w_1-1)^5} & ,\dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1\\ 0 & ,otherwise\end{cases} \tag{50}$$

Thus $w_2=\dfrac{1-5w_1+(7w_1-1)\sqrt[5]{5u}}{6}=F_{w_2|w_1}^{-1}(u)$ *where* u *is a uniform random variable* $U\sim(0,1)$

To calculate w_3 we need to obtain its conditional probability density function given w_1 *and* w_2 in Eq. 51

$$f\left(w_3\middle|w_1,w_2\right)=\frac{f\left(w_1,w_2,w_3\right)}{f\left(w_1,w_2\right)}$$

$$f\left(w_1,w_2,w_3\right)=\begin{cases}\dfrac{c}{18}\left(5w_3+w_1+w_2-1\right)^3 & ,\dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1 \\ & ;\ \dfrac{1}{4}\left(1-w_1-w_2\right)<w_3<w_2 \\ 0 & ,otherwise\end{cases}$$

$$f\left(w_3\middle|w_1,w_2\right)=\begin{cases}\dfrac{20\left(5w_3+w_1+w_2-1\right)^3}{\left(6w_2+w_1-1\right)^4} & ,\dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1 \\ & ;\ \dfrac{1}{4}\left(1-w_1-w_2\right)<w_3<w_2 \\ 0 & ,otherwise\end{cases} \tag{51}$$

The corresponding c.d.f is Eq. 52:

$$F\left(w_3\middle|w_1,w_2\right)=\begin{cases}\dfrac{\left(5w_3+w_1+w_2-1\right)^4}{\left(6w_2+w_1-1\right)^4} & \dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1 \\ & ;\ \dfrac{1}{4}\left(1-w_1-w_2\right)<w_3<w_2 \\ 0 & ,otherwise\end{cases} \tag{52}$$

By the inverse probability transformation $u=F\left(w_3\middle|w_1,w_2\right)$ is a uniform random variable on $(0,1)$, thus solving for w_4 we obtain Eq. 53:

$$w_3=F^{-1}_{w_3|w_1,w_2}\left(u\right)=\frac{1-w_1-w_2+u\left(6w_2+w_1-1\right)}{5} \tag{53}$$

The next weight to find is w_4 . Its p.d.f. is Eq. 54

$$f\left(w_4 \middle| w_1, w_2, w_3\right) = \frac{f\left(w_1, w_2, w_3, w_4\right)}{f\left(w_1, w_2, w_3\right)}$$

$$= \begin{cases} & , \frac{1}{7} < w_1 \le \frac{1}{6} \ ; \ \frac{1-w_1}{6} < w_2 < w_1 \\ \frac{3\left(4w_4 + w_1 + w_2 + w_3 - 1\right)^2}{\left(5w_3 + w_2 + w_1 - 1\right)^3} & ; \frac{1}{4}\left(1 - w_1 - w_2\right) < w_3 < w_2 \\ & ; \frac{1}{4}\left(1 - w_1 - w_2 - w_3\right) < w_4 < w_3 \\ 0 & , otherwise \end{cases} \tag{54}$$

Finally the corresponding c.d.f is Eq. 55

$$F\left(w_4 \middle| w_1, w_2, w_3\right) = \begin{cases} & , \frac{1}{7} < w_1 \le \frac{1}{6} \ ; \ \frac{1-w_1}{6} < w_2 < w_1 \\ \frac{\left(4w_4 + w_1 + w_2 + w_3 - 1\right)^3}{4\left(5w_3 + w_2 + w_1 - 1\right)^3} & ; \frac{1}{4}\left(1 - w_1 - w_2\right) < w_3 < w_2 \\ & \frac{1}{4}\left(1 - w_1 - w_2 - w_3\right) < w_4 < w_3 \\ 0 & , otherwise \end{cases} \tag{55}$$

Solving for w_4 in Eq. 56 $u = \frac{\left(4w_4 + w_1 + w_2 + w_3 - 1\right)^3}{4\left(5w_3 + w_2 + w_1 - 1\right)^3}$ (56)

$$we \quad get : w_4 = \frac{\left(1 - w_1 - w_2 - w_3\right) + \sqrt[3]{4u}\left(5w_3 + w_2 + w_1 - 1\right)}{4} \tag{57}$$

The next weight to find is w_5 ,thus we need to calculate its conditional probability function given that four previous weights are known. In equation 58 is written the conditional probability distribution .

$$f\left(w_5 \middle| w_1, w_2, w_3, w_4\right) = \frac{f\left(w_1, w_2, w_3, w_4, w_5\right)}{f\left(w_1, w_2, w_3, w_4\right)}$$

$$= \begin{cases} \dfrac{3\left(3w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)}{\left(4w_4 + w_3 + w_2 + w_1 - 1\right)^2} & \begin{array}{l} ,\dfrac{1}{7} < w_1 \le \dfrac{1}{6} \;;\; \dfrac{1-w_1}{6} < w_2 < w_1 \\ ;\dfrac{1}{4}\left(1-w_1-w_2\right) < w_3 < w_2 \\ \dfrac{1}{4}\left(1-w_1-w_2-w_3\right) < w_4 < w_3 \\ ;\dfrac{1}{3}\left(1-w_1-w_2-w_3-w_4\right) < w_5 < w_4 \end{array} \\ 0 & ,othwerwise \end{cases} \tag{58}$$

Thus its cumulative distribution function is in equation 59:

$$F\left(w_5 \middle| w_1, w_2, w_3, w_4\right) = \begin{cases} \dfrac{\left(3w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)^2}{2\left(4w_4 + w_3 + w_2 + w_1 - 1\right)^2} & \begin{array}{l} ,,\dfrac{1}{7} < w_1 \le \dfrac{1}{6} \;;\; \dfrac{1-w_1}{6} < w_2 < w_1 \\ ;\dfrac{1}{4}\left(1-w_1-w_2\right) < w_3 < w_2 \\ \dfrac{1}{4}\left(1-w_1-w_2-w_3\right) < w_4 < w_3 \\ ;\dfrac{1}{3}\left(1-w_1-w_2-w_3-w_4\right) < w_5 < w_4 \end{array} \\ 0 & ,otherwise \end{cases} \tag{59}$$

Solving for w_5 in the equation 60

$$u = \frac{\left(3w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)^2}{2\left(4w_4 + w_3 + w_2 + w_1 - 1\right)^2} \tag{60}$$

and obtain the fifth weight in Eq. 61

$$w_5 = \frac{\sqrt{2u}\left(4w_4 + w_3 + w_2 + w_1 - 1\right) + 1 - w_4 - w_3 - w_2 - w_1}{3} \tag{61}$$

The next step is to find weight w_6. Similarly to last procedure its conditional cumulative probability function is Eq. 62:

$$f\left(w_6\middle|w_1,w_2,w_3,w_4,w_5\right)=\frac{f\left(w_1,w_2,w_3,w_4,w_5,w_6\right)}{f\left(w_1,w_2,w_3,w_4,w_5\right)}$$

$$=\begin{cases}\dfrac{2}{\left(3w_5+w_4+w_3+w_2+w_1-1\right)} & ,\dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1;\\ & \dfrac{1}{4}\left(1-w_1-w_2\right)<w_3<w_2\\ & \dfrac{1}{4}\left(1-w_1-w_2-w_3\right)<w_4<w_3\\ & ;\dfrac{1}{3}\left(1-w_1-w_2-w_3-w_4\right)<w_5<w_4\\ & \dfrac{1}{2}\left(1-w_1-w_2-w_3-w_4-w_5\right)<w_6<w_5\\ 0 & ,othwerwise\end{cases} \tag{62}$$

Its cumulative distribution function is Eq. 63:

$$F\left(w_6\middle|w_1,w_2,w_3,w_4,w_5\right)=\begin{cases}\dfrac{\left(2w_6+w_5+w_4+w_3+w_2+w_1-1\right)}{\left(3w_5+w_4+w_3+w_2+w_1-1\right)} & ,\dfrac{1}{7}<w_1\le\dfrac{1}{6}\ ;\ \dfrac{1-w_1}{6}<w_2<w_1;\\ & \dfrac{1}{4}\left(1-w_1-w_2\right)<w_3<w_2\\ & \dfrac{1}{4}\left(1-w_1-w_2-w_3\right)<w_4<w_3\\ & ;\dfrac{1}{3}\left(1-w_1-w_2-w_3-w_4\right)<w_5<w_4\\ & \dfrac{1}{2}\left(1-w_1-w_2-w_3-w_4-w_5\right)<w_6<w_5\\ 0 & ,othwerwise\end{cases} \tag{63}$$

to conclude solving for w_6 in the Equation 64

$$u = \frac{\left(2w_6 + w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)}{\left(3w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)} \tag{64}$$

$$we \quad get : w_6 = \frac{2uw_5 + \left(w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)\left(u - 1\right)}{2}$$

and the last weight is obtained from : $w_7 = 1 - w_1 - w_2 - w_3 - w_4 - w_5 - w_6$

4.3.1 Seven Weights algorithm Pseudo code

Finally the algorithm's pseudo code to generate the $w_1, w_2, w_3, w_4, w_5, w_6$ *and* w_7 is as follows:

1) Randomly generate an $u \in (0,1)$

2) Compute $w_1 = \frac{6 + \sqrt[6]{u}}{42}$

3) Compute

$$w_2 = \frac{1 - 5w_1 + \left(7w_1 - 1\right)\sqrt[5]{5u}}{6}$$

4) Compute

$$w_3 = \frac{1 - w_1 - w_2 + u\left(6w_2 + w_1 - 1\right)}{5}$$

5) Compute $w_4 = \frac{\left(1 - w_1 - w_2 - w_3\right) + \sqrt[3]{4u}\left(5w_3 + w_2 + w_1 - 1\right)}{4}$

6) Compute $w_5 = \frac{\sqrt{2u}\left(4w_4 + w_3 + w_2 + w_1 - 1\right) + 1 - w_4 - w_3 - w_2 - w_1}{3}$

7) Compute $w_6 = \frac{2uw_5 + \left(w_5 + w_4 + w_3 + w_2 + w_1 - 1\right)\left(u - 1\right)}{2}$

finally calculate $w_7 = 1 - w_1 - w_2 - w_3 - w_4 - w_5 - w_6$

8) Repeat all steps until n iterations to generate a weights set of w_1, w_2, w_3, w_4, w_5 *and* w_6 .

Example 4.2

As in the first Example 4.1 for the five weights case, the same cars-sale file will be tested adding two additional features. Indeed it has been aggregated the car variables width and weight. The pruned subsets for three threshold values are:

Table 13: Reduced Pareto set with 7 weights and α=0.2

Car	pruned for alpha=0.2		price	engines	horsepow	fuelcap	mpg	width	curb_wgt
25	Chevrolet	Metro	9.235	1	55	10.3	45	62.6	1.895

Again the cheapest and most gas saving car is the Chevy metro

Table 14: Reduced Pareto set with 7 weights and α=0.3

Car	pruned for alpha=0.3		price	engines	horsepow	fuelcap	mpg	width	curb_wgt
25	Chevrolet	Metro	9.235	1	55	10.3	45	62.6	1.895
46	Hyundai	Accent	9.699	1.5	92	11.9	31	65.7	2.24

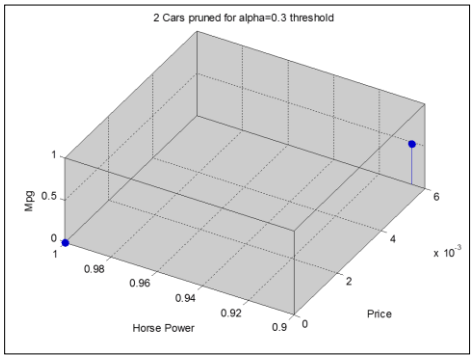

Figure 23: Graph of pruned values in table 14

For the second threshold value α=0.3 there is a difference in the number of cars compared with the five weights case such that simplifies the decision maker car selection. Finally for the last threshold value α=0.4 a considerable car reduction is observed comparing to the five case.

Table 15: Reduced Pareto set with 7 weights and α=0.4

car	pruned for alpha=0.4		price	engines	horsepow	fuelcap	mpg	width	curb_wgt
25	Chevrolet	Metro	9.235	1	55	10.3	45	62.6	1.895
44	Honda	Civic	12.885	1.6	106	11.9	32	67.1	2.339
46	Hyundai	Accent	9.699	1.5	92	11.9	31	65.7	2.24
47	Hyundai	Elantra	11.799	2	140	14.5	27	66.9	2.626
74	Nissan	Sentra	13.499	1.8	126	13.2	30	67.3	2.593
93	Saab	3-Sep	12.535	1.9	100	12.1	33	66.4	2.367
94	Saturn	SL	10.685	1.9	100	12.1	33	66.4	2.332
103	Toyota	Celica	16.875	1.8	140	14.5	31	68.3	2.425

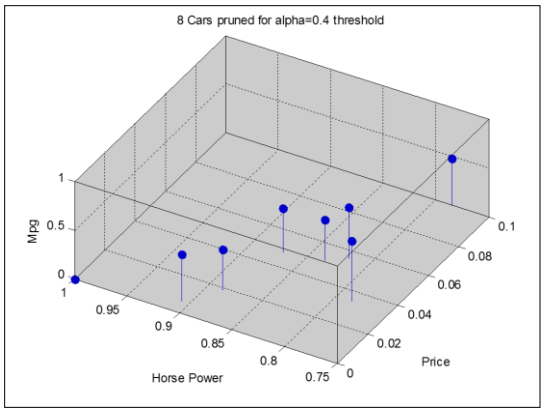

Figure 24: Graph of pruned values in table 15

Conclusions: This weight generator method has the advantage of providing the decision maker a linear computational time complexity algorithm for the ranked weights search. However the larger the number of objective functions in a non-numerical ranking multi-objective optimization problem, the larger the mathematical complexity to obtain the algorithm to generate the weights required.

In the next chapter a post-Pareto approach using non-uniform weight generator proposed and developed by Carrillo, Taboada (2012) will be presented and compared its performance with a pair of examples.

Chapter 5:

5. A Post-Pareto Approach Using a Non-Uniform Weight Generator Method

So far there exist two general approaches to solve multiple objective problems. The first approach involves the aggregation of all the objective functions into a single composite objective function. Mathematical methods such as the weighted sum method, goal programming, or utility functions are methods that pertain to this general approach. The output of this method is a single solution. On the other hand, there is the multiple objective evolutionary algorithm approach that offer the decision maker a set of trade off solutions usually called non dominated solutions or, Pareto-optimal solutions. This set is usually very large and the decision maker faces the problem of reducing the size of this set to have a manageable number of solutions to analyze. This paper presents an additional post-Pareto approach to prune the non-dominated set of solutions obtained by multiple objective evolutionary algorithms due to Carrillo & Taboada(2012). The proposed approach uses a non-uniform weight generator method to reduce the size of the Pareto-optimal set. A pair of examples is presented to show the performance of the method.

5.1 A general method to create n-increasingly ordered non-uniform distributed random numbers

In this section is developed the algorithm that generates n uniformly distributed weights that are the basis for the non-numerical ranking weights needed for the scalar function

$$f_{composite} = w_1 f_1 + w_2 f_2 + w_n f_n \ .$$

To obtain the weights first a finite sequence of $x_1 < x_2 < \cdots < x_n$ positive values is produced using $X_1,\ldots,X_n$ *random variables uniformly distributed* $X_i \sim U(i-1,i)$ afterwards calculating theratio $w_k = \frac{x_k}{M}$ *for* $k = 1,\ldots,n$ where $M = \sum_{k=1}^{n} x_k$ gives the weights desired. Method development: Let be $X_1,\ldots,X_n$ *random variables uniformly distributed* $X_i \sim U(i-1,i)$ thus its corresponding probability density functions are as shown in Eq. 65:

$$f(x_i)=\begin{cases}1 & if\ \ x_i\in(i-1,i)\\ 0 & otherwise\end{cases}\qquad for\ \ i=1,\ldots,n \tag{65}$$

Since $X_1,\ldots,X_n$ are independent, its joint probability density function is as shown in Eq. 66 :

$$f(x_1,x_2,...,x_n)=\prod_{i=1}^{n}f(x_i)=\begin{cases}1 & if\ x_i\in(i-1,i)\ \ for\, i=1,...,n\\ 0 & otherwise\end{cases} \tag{66}$$

Indeed , by multiple integration is proven that the joint function in Eq. 66 is a probability density function as follows $\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}f(x_1,x_2,...,x_n)dx_1dx_2\cdots dx_n=\int_{n-1}^{n}\cdots\int_{1}^{2}\int_{0}^{1}dx_1dx_2\cdots dx_n=1$

To generate each one of the X_i values such that $X_1<X_2<\cdots<X_n$ starting with X_1 is enough to calculate its cumulative distribution and apply the Probability Integration Theorem is applied as follows:

According to Eq. 65 the p.d.f. for the first random variable X_1 is as Eq.67:

$$f(x_1)=\begin{cases}1 & if\ \ x_1\in(0,1)\\ 0 & otherwise\end{cases} \tag{67}$$

and its cumulative distribution function is :

$F(x_1)=\int_0^x ds=x_1$ .By the Integral probability Theorem $F(x_1)=u_1$ has a uniform distribution $u\sim U(0,1)$ .Therefore $x_1=u_1$. To obtain the second value x_2 given that the first value x_1 is known, calculate the conditional probability density function Eq. 68

$$f(x_2\mid x_1)=\frac{f(x_1,x_2)}{f(x_1)}=\begin{cases}1 & if\ \ x_i\in(i-1,i)\ \ for\ i=1,2\\ 0 & otherwise\end{cases} \tag{68}$$

Similarly to the previous x_1 we obtain its cumulative distribution function:

$F(x_2\mid x_1)=\int_1^{x_2}ds=x_2-1=u_2\ \ where\ u_2\in U(0,1)$ .Solving for x_2 we get $x_2=u_2+1$.

Since this is an iterative procedure for the k-th case the c.d.f is

$$F(x_k\mid x_1,x_2,...x_{k-1})=\int_2^{x_k}ds=x_k-(k-1)=u_k\ \ where\ u_k\in U(0,1)$$

Solving for x_k we get the general iterative formula Eq. 69 for each one of the x_k values:

$$x_k = u_k + (k-1) \tag{69}$$

For the values k=1, 2, 3,…, n the x_k sequence obtained is:

$$x_1 = u_1 < x_2 = u_2 + 1 < x_3 = u_3 + 2 < \cdots < x_k = u_k + (k-1) < \cdots < x_k = u_n + (n-1)$$

Finally scaling the x_i values previously obtained, a sequence of weights in (0,1) is generated :

Let be $M = \sum_{k=1}^{n} x_k$, then a non-uniformly distributed increasing sequence is obtained

$\{w_k\}_{k=1}^{n}$ *such that* $0 < w_1 < w_2 < \cdots < w_n < 1$ *and* $\sum_{k=1}^{n} w_k = 1$ as defined in Eq. 70

$$w_k = \frac{x_k}{M} \quad for \quad k = 1,\ldots,n \tag{70}$$

5.2 Non-uniform weights pseudo code Algorithm

To obtain the $w_1, < w_2 < w_3, < \ldots < w_n$ values ,follow the steps:

1. Randomly generate an $u_1 \in (0,1)$

2. calculate $x_1 = u_1 + (1-1)$

3. Randomly generate an $u_2 \in (0,1)$

4. calculate $x_2 = u_2 + (2-1)$

5. Continue the iteration according to the formula $x_k = u_k + (k-1)$ *for* $k = 1,\ldots,n$

6. Finally calculate $M = \sum_{k=1}^{n} x_k$ and $w_k = \frac{x_k}{M}$ *for* $k = 1,\ldots,n$

Example 5.1 To show the performance of the non-uniform weight generator an algorithm presented by Carrillo & Taboada (2012) will be applied into a set of non dominated solutions. The Pareto set was obtained from the work presented by Taboada & Coit (2007) . The problem solved in that paper is a well-known problem called redundancy allocation problem (RAP). The

RAP refers to a system of s subsystems in series. For each subsystem, there are m_i functionally equivalent components, with different levels of cost, weight, reliability and other characteristics, which may be selected. There is an unlimited supply of each of the m_i choices. The objective of the problem is to find how many components to set in parallel in each subsystem and of which supplier in order to optimize three different objectives. For this specific case the objectives considered were: reliability cost and weight. Table 16 shows the Pareto-set of solutions obtained in . The Pareto sets consists of 75 solutions.

Table 16: Non dominated Solutions

Solution	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Reliability	0.6820	0.7204	0.7571	0.7886	0.8432	0.8596	0.8604	0.8645	0.8753	0.8771	0.8821	0.8833	0.8884	0.8932	0.9012
Cost	13	16	19	17	19	21	21	23	31	23	25	27	29	33	34
Weight	13	16	19	17	19	21	21	23	31	23	25	27	29	33	34
Solution	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
Reliability	0.9016	0.9156	0.9172	0.9196	0.9262	0.9342	0.9360	0.9426	0.9447	0.9451	0.9619	0.9623	0.9636	0.9684	0.9702
Cost	35	33	34	36	37	35	36	36	38	39	38	39	39	40	41
Weight	22	28	27	28	26	32	31	32	29	30	36	34	35	40	39
Solution	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
Reliability	0.9730	0.9734	0.9746	0.9797	0.9808	0.9822	0.9824	0.9826	0.9828	0.9835	0.9844	0.9851	0.9860	0.9864	0.9905
Cost	42	43	42	44	43	52	45	53	54	46	50	48	54	55	56
Weight	34	32	41	38	47	35	45	33	39	46	42	44	43	41	37
Solution	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
Reliability	0.9911	0.9921	0.9923	0.9943	0.9949	0.9980	0.9981	0.9981	0.9984	0.9988	0.9990	0.9990	0.9991	0.9991	0.9994
Cost	58	59	57	58	60	68	69	73	75	72	77	79	80	82	62
Weight	41	37	50	45	49	45	43	55	47	67	51	69	69	73	82
Solution	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
Reliability	0.9996	0.9996	0.9997	0.9997	0.9998	0.9998	0.9999	0.9999	0.9999	0.999962	0.999977	0.999982	0.999994	0.999998	0.999999
Cost	81	64	70	74	68	76	85	87	88	90	94	119	121	125	143
Weight	77	85	89	91	92	94	81	85	85	89	105	110	113	121	120

The algorithm applied to prune the data above in Table 16 once the weights were generated is as follows: Calculate $f_{composite} = w_1 f_1 + w_2 f_2 + w_n f_n$ for k=100,000 weights triplets and bound the points that satisfy f <alpha threshold .

Several thresholds were used with values from [0.063 to 0.1] as shown in Table 17.

Table 17: Pruned solutions

Thresholds	Solutions obtained with the non-uniform weights generator and several threshold values								
Th<0.1	32	34	36	38	45	47	51	52	54
Th<0.08	38	45	47	51	52	54			
Th<0.075	45	47	51	52	54				
Th<0.07	45	47	51	52					
Th<.065	47	52							
Th<.064	47								
Th<.063	0								

It is clear that if the alpha threshold is increased , the optimal-Pareto solutions subset enlarges ,which complicates the solutions selection for the decision maker. On the other hand the results

obtained by Coit &Taboada [3] (2007) uses a different weight generator, and values selection was done calculating the minimum value of each set of f values calculated for every 10,000 random vectors $w = (w_1, w_2, ..., w_n)$. Is important to mention that these weights were produced following the same non-numerical preferences procedure . Results are in table 18:

Table 18: Pruned solutions

Solutions obtained with the non-numerical ranking preferences method			
32	34	37	42
49	60	62	65
70			

When comparing results, both sets share a 22% of solutions according to table 16 .

Conclusions The non-uniform weight generator was successfully applied in the RAP problem using a ranking preferences method. The results obtained by comparing the two Pareto-optimal pruning methods gives the decision maker workable sized collections of values to select, considering that in the pruning stage the non uniform method uses a boundary method procedure compared with the statistical one proposed by Taboada ,Coit, (2007).

Chapter 6:

Conclusion and future work

6.1 Conclusions

Reducing the size of a Pareto front is yet a problem to be addressed in the future. A general overview of the latest research done to reduce the Pareto front size has been presented to familiarize the reader. Some of these procedures are that recent that has not been checked for their advantages and disadvantages but by their developers. In this thesis has been added to the list two contributions to prune the Pareto front in chapters 4 and 5. The first one a general iterative method to obtain the weights needed for every non-numerical ranked objective functions in a MOOP for a composite utility function known as weighting function used to reduce Pareto fronts with the aid of a boundary technique. The strength of this first procedure is that the probability properties of each of the weights obtained are known through their probability density functions and cumulative distribution functions developed to end up with each weight value.

For the second approach an algorithm was demonstrated to calculate any number of ranked weights to be used on a weighting composite function for a MOOP problem of any size. This second weight generator technique is simpler than the first one, linear in time complexity but due to its non-uniform properties the probability properties of each of the weights are not obtained in this thesis such issue is a mathematical problem worth to be addressed in a next research paper.

Both approaches were successfully applied to show their performance.

6.2 Future Research

A future research proposal for Pareto-optimal analysis is the application of linear programming interior point methods over the constraint region obtained from a MOP previously adapted to be analyzed with the non-numerical preferences method and some statistical procedure to reduce to an acceptable size the Pareto front. To support this proposal instead of just applying the simplex method is that regardless large classes of problems are best solved with the simplex method, a drawback of the simplex method is its exponential time complexity (Tanksley &

Latriece 2009). It is possible that all of the vertices of the feasible polyhedron have to be visited before an optimal solution is reached. Interior point methods are well-suited for solving very large scale optimization problems and has been proved to have computational polynomial complexity. For pruning large Pareto-fronts it seems to be a better choice. Karmarkar (1984) introduced an Interior-Point Method (IPM) for linear programming, combining the desirable theoretical properties of the ellipsoid method proposed by Leonid Khaciyan in 1979 and the practical advantages of the simplex method. Interior point methods do not pass from vertex to vertex along the edges of the feasible region, which is the main feature of the simplex algorithm; they follow the central path in the interior of the feasible region. In Figure 26 is shown a geometric description of the Interior point method (Colombo & Marco.,2007)

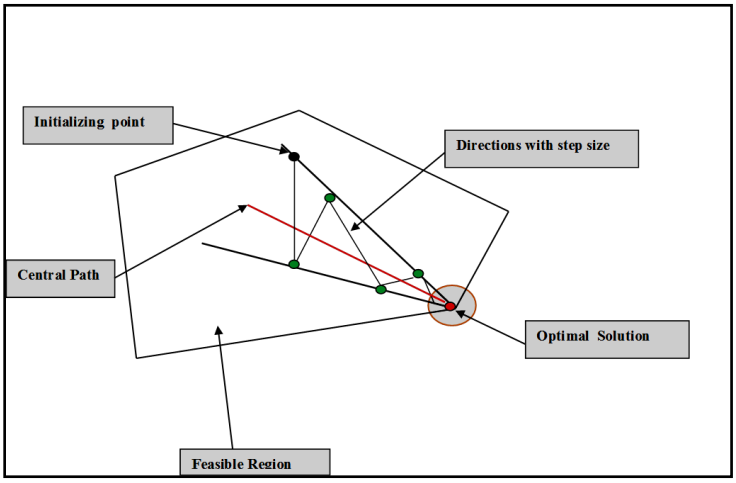

Figure 25: Interior point method

6.2.1 Interior Point Method for post Pareto Optimality

In this section is shown some preliminary results after being applied Interior point method for reducing the Pareto front of 75 triplets obtained with an Evolutionary algorithm and the non-numerical ranking method to solve a redundancy allocation problem in chapter 5. The weighting $f_{composite} = x_1 f_1 + x_2 f_2 + \cdots + x f_n$ is minimized under the constraints shown below

in Equation 71

$$\begin{aligned} &\min\ x_1 f_1 + x_2 f_2 + x_3 f_3 \\ &s.t. \qquad 1 > x_1 > x_2 > x_3 > 0 \end{aligned} \tag{71}$$

, the triplet (f_1, f_2, f_3) $belongs\,to\,the\ Pareto-front$ set of 75 triplets such that f_1 is the reliability of a redundant system, f_2 is its cost and f_3 its weight . The *x's* are the weights to search for the minimizing problem. This MOOP is equivalent to:

$$\min\ f_1x_1 + f_2x_2 + f_3x_3$$

$$\begin{aligned} s.t. \quad & x_1 && < 1 \\ & -x_1 + x_2 && < -1 \\ & && -x_2 + x_3 < 0 \end{aligned}$$

The weighting function and constraints can be expressed in matricial form as

$$f = [f_1 \quad f_2 \quad f_3]$$

$$A = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

After applying the LIPSOL interior point algorithm some of the optimal values obtained are shown in the following tables:

Reliability was considered the main preference after cost and weight.

The selection is shown in Table 19.

Table 19: Interior point optimal rated values

count	x1r	x2r	x3r
15	0.94	0.05	0.01
16	0.93	0.06	0.01
17	0.88	0.08	0.04
18	0.85	0.11	0.04
19	0.95	0.05	0.01
60	0.77	0.19	0.04
64	0.65	0.34	0.01
63	0.70	0.28	0.02

Whose real values correspond are in Table :

Table 20: Interior point true optimal values

Solution	Reliability	Cost	Weight
15	0.901223	34	24
16	0.901588	35	22
17	0.915556	33	28
18	0.917232	34	27
19	0.919619	36	28
60	0.999363	62	82
63	0.999721	70	89
64	0.999732	74	91

Is important to note that the results obtained comprises optimal components from lower values for cost and reliability up to highest values for cost and reliability. Is precisely this property which makes worth to continue in this research direction.

6.2.2 Sweeping Cones Post-Pareto Analysis

A second proposal for future research is to develop an approach for post Pareto analysis consisting into deepen the idea of using sweeping cones to prune a Pareto front. The core of this procedure stems in normalizing the vectors of the Pareto front and the weights obtained to be used in the non-numerical ranking preferences method. Afterwards a collection of sweep cones must be generated to capture those optimal points that best satisfy the ranking preferences.

Geometrically the sweep cones proposal allocates all of the function values and the weights on a sphere of radius one centered in the origin of coordinates. Afterward the sweep cones collects optimal points onto the sphere as shown in Figures 26,27 for the redundancy allocation problem presented in Chapter 4.

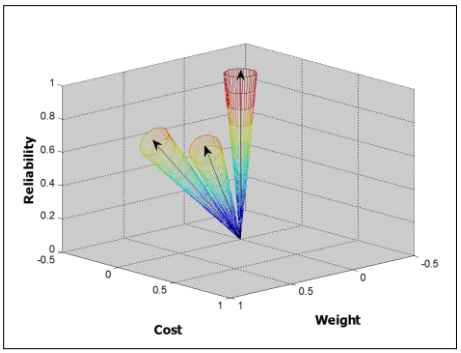

Figure 26: Sweeping cones

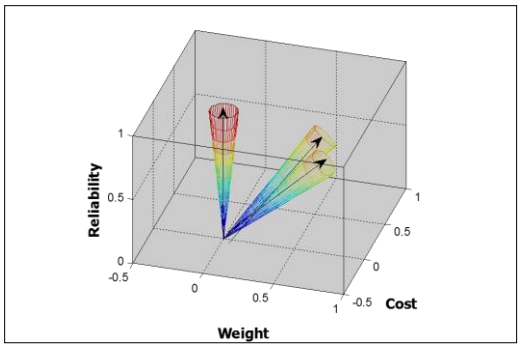

Figure 27: Sweeping cones

After having applied the sweep cones proposal to the 75 Pareto front in the redundancy allocation optimization problem in chapter 4 the following preliminary results in Table X were obtained after a pruning process for several alpha threshold values which is equal to the cosine of the theta angle shown in Table 21: *i. e.* $\cos(\theta) = \alpha$

Table 21: Pruned values with the sweeping cones approach

1,000,000 Iterations for each angle										
Running time (secs)		34.7822	34.1318	34.2003	31.0877	33.627	33.4596	33.2643	33.2854	33.5499
Angle	θ	42.0268	42.0979	41.926	41.755	41.582	41.409	41.236	41.062	40.9746
Threshold	$\alpha<\Gamma$	0.74<F	0.742<F	0.744<F	0.746<F	0.748<F	0.750<F	0.752<F	0.754<F	0.755<F
	Count									
1.33%	1	9	9	9	9	9	9	9	9	
2.66%	2	14	14	14	14	16	16	16		
4%	3	15	15	15	15	38	38			
5.33%	4	16	16	16	16	47				
	5	18	19	20	20					
	6	19	20	36	36					
	7	20	24	38	38					
	8	24	25	45	45					
	9	25	32	47	47					
	10	32	36	51	52					
14.66%	11	36	38	52	54					
	12	38	39	54						
17.33%	13	39	45	56						
	14	44	46							
	15	45	47							
	16	46	51							
22.66%	17	47	52							
	18	51	54							
25.33%	19	52	56							
	20	54								
28%	21	56								
	.									
	.									
	48									
	49									
	50									
	.									
	.									
	73									
	74									
	75									

References

1. Blasco, X. *, J.M. Herrero, J. Sanchis, M. Martínez(2008) "*A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization*", Information Sciences 178 (2008) 3908–3924

2. . Branke, J.; Deb, K.; Miettinen, K.;Slowinski, R. (Eds.) (2008).“*Multiobjective Optimization Interactive and Evolutionary Approaches*”, Springer

3. Carrillo,V.,Aguirre,O.,Taboada, H.(2011),"*Applications and performance of the non-numerical ranking preferences method for post-Pareto optimality*",Procedia Computer Science vol 6,243-248.

4 . Carrillo, Taboada (2012). "*General Iterative procedure for the Non-numerical ranking preferences method in Multi-objective Optimization*", Complex Adaptive Systems, Publication 2,Washington D.C.

5. Carrillo, Taboada (2012)." *A Post-Pareto Approach for Multi-Objective Decision Making Using a Non-Uniform Weight Generator Method*", Complex Adaptive Systems, Publication 2,Washington D.C.

6. Colombo,Marco.,2007. "Advances in Interior Point Methods for Large-Scale Linear Programming", Phd Thesis, University of Edinburgh

7. Collete,Y,Siarry,P. (Eds.) (2003).“*Multiobjective Optimization Interactive Principles and Case Studies* ”, Springer

8. Deb,Kalyanmoy (2004). Multi-objective Optimization using evolutionary algorithms.Wiley.

9. Dreo *et all* ,2006. " Metaheuristics for Hard Optimization :Simulated Annealing, Tabu Search, Evolutionary and Genetic Algorithms, Ant Colonies, …", Springer

10. Fung,G. (2001). A comprehensive overview of basic clustering algorithms

11. Goldberg,D.E.(1991)."*Real-coded genetic algorithms,virtual alphabeths,and blocking Complex Systems*" 5(2),139-168

12. Kaisa, M. (1999), ”*Nonlinear Multiobjective Optimization*”.Kluer Academic Publishers

13. Knowles,J.D.,Watson,R.A.(2000).Approximating the non-dominated front using the Pareto archived evolution strategy."*Evolutionary Computation Journal* "8(2),149-172

13. Luo, F., Khan, L., Bastani, F., Yen, I-L., and Zhou, J. 2004. A Dynamically Growing Self-Organizing Tree (DGSOT) for Hierarchical Clustering Gene Expression Profiles. Bioinformatics vol 20, 2605-2617.

14. Marler, R.T. and Arora, J.S. "Survey of multi-objective optimization methods for engineering". Struct Multidisc Optim 26, 369–395 (2004)

15. Mietinen,K.(1999).Nonlinear Multi-objective Optimization. Boston: Kluer Academic Publishers

16. Nakayama ,H., Yun,Y., Yoon,Min. 2009" Sequential Approximate Multiobjective Optimization Using Computational Intelligence", Springer

17. Nikhil Padhye, Subodh Kalia and Kalyanmoy Deb, (2009).Multi-objective Optimization and Multi-criteria Decision Making in SLS, Department of Mechanical Engineering Indian Institute of Technology Kanpur, Kanpur-208016, U.P., India November 22.

18. Oswaldo Aguirre MSa,*, Heidi Taboada PhDb ."*A Clustering Method Based on Dynamic Selforganizing Trees for Post-Pareto Optimaliyy Designs*".Complex Adaptive Systems, Volume 1 Cihan H. Dagli, Editor in Chief Conference Organized by Missouri University of Science and Technology 2011- Chicago, IL

19. Padhye, et al.(2009). "*Multi-objective Optimization and Multi-criteria Decision Making in SLS Performance* ", Computational Optimization and Applications, 28, 357–372, 2004 2004 Kluwer Academic Publishers. Manufactured in The Netherlands 4

20. Rosseew Peter J. (1987)."*Shilouettes:A graphical aid to the interpretation of the validation of cluster analysis*".Journal of computational and applied mathematics,20,53-65

21. Rousseeuw P.&, Trauwaert E. and Kaufman L. (1989):" *Some silhouette-based graphics for clustering interpretation*". Belgian Journal of Operations Research, Statistics and Computer Science, 29(3).

22. Taboada, Coit ,Baheranuala,Wattanapongsakorn.(2007). "*Practical solutions for multi-objective optimization: An application to system reliability design problems* ",Reliability Engineering and System Safety ,92, 314-322

23. Taboada, H**.** & Coit, D. (2006). "*Data Mining Techniques to Facilitate the Analysis of the Pareto-Optimal Set for Multiple Objective Problems*". In Proceedings of the Industrial Engineering Research Conference (IERC), Orlando, Florida.

24. Tanksley,Latriece (2009). "*Interior Point Methods and kernel functions of a linear programming problem*". Georgia Southern University, Master thesis.

25. Vandana ,V.,Jacobson , S. ,and Stori, J. (2004) A Post-Optimality Analysis Algorithm for Multi-Objective Optimization,Computational Optimization and Applications,28,357-372.

26. Yang , Xin-She. 2008,"Introduction to Mathematical : From Linear Programming to Metaheuristics Cambridge International Science Publishing 2008

27. Yu, P. L. and Leitmann, G., 1974, "Compromise solutions, dominations structures, and Salukvadze's solution," Journal of the Optimization Theory and Applications, Vol. 13, pp.362-378.

28. Yu Chen, Xiufen Zou , Weicheng Xie.(2001)" Conditions for the convergence of evolutionary algorithms". Journal of Systems Architecture.vol 47,601-612

29. Zitzler,E. and Thiele,L."*An evolutionary algorithm for multi-objective optimization. The strenght Pareto approach*".Technical Report 43,Zurich,SwitzerlandComputer Engineering and Networks Laboratory(TIK),Swiss Federal Institute of Technology (ETH)

30. Zeleny, M. 1973, "*Compromise programming, in: Multiple Criteria Decision Making*", eds: J. L. Cochrane and M. Zeleny, University of South Carolina Press, Columbia, SC, pp. 262-301.

31. Zhaojun Li a, Haitao Liao b,_, David W. Coit,"*A two-stage approach for multi-objective decision making with applications to system reliability optimization* ",Reliability Engineering and System Safety 94 (2009) 1585–1592

Curriculum Vita

Victor was born in Lerdo, Durango, Mexico. The first of six offspring of Victor Carrillo and Maria Elena Saucedo. He received her bachelor degree in the Mexico City National Polytechnic Institute in fall 1977. Then he started his professional career as lecturer in the Metropolitan Autonomous University in Mexico City from 1977 to 1992.Actually he is lecturer in the Juarez Autonomous University in Juarez, Chihuahua since 1993. He entered The University of Texas at El Paso in fall 1988. While pursuing a master's degree in Statistics, he worked with Dr. Joanne Peeples as her Teaching Assistant. In May 1990 he graduate and obtain his Master's. He was lecturer in the Community College at El Paso Texas for the Summer term in 1990. He started pursing his PhD degree in Computational Science in August 2009. He presented the talk "Applications and performance of the non-numerical ranking preferences method for post-Pareto optimality" at the Complex adaptive systems conference in Chicago, IL 2011. He attended to INFORMS 2012 Annual Meeting at Phoenix, AZ and had the opportunity to present a talk about his research in Post-Pareto Analysis. In November 2012 he presented the talks "General Iterative procedure for the Non-numerical ranking preferences method in Multi-objective Optimization" and " A Post-Pareto Approach for Multi-Objective Decision Making Using a Non-Uniform Weight Generator Method", in the Complex adaptive systems conference in Washington D.C.

Permanent address: Solar de Gardenias 2440
Juarez, Chihuahua Mexico

This thesis was typed by the author .