

2017-01-01

Development of post-Pareto optimality methods for multiple objective optimization

Juan Valentin Fernandez

University of Texas at El Paso, jvfernandez77@gmail.com

Follow this and additional works at: https://digitalcommons.utep.edu/open_etd



Part of the [Operational Research Commons](#)

Recommended Citation

Fernandez, Juan Valentin, "Development of post-Pareto optimality methods for multiple objective optimization" (2017). *Open Access Theses & Dissertations*. 443.

https://digitalcommons.utep.edu/open_etd/443

This is brought to you for free and open access by DigitalCommons@UTEP. It has been accepted for inclusion in Open Access Theses & Dissertations by an authorized administrator of DigitalCommons@UTEP. For more information, please contact lweber@utep.edu.

DEVELOPMENT OF POST-PARETO OPTIMALITY METHODS FOR
MULTIPLE OBJECTIVE OPTIMIZATION

JUAN V. FERNANDEZ

Master's Program in Industrial Engineering

APPROVED:

Heidi A. Taboada, Ph.D., Chair

Jose F. Espiritu, Ph.D.

Vinod Kumar, Ph.D.

Natalia Villanueva Rosales, Ph.D.

Charles Ambler, Ph.D.
Dean of the Graduate School

Copyright ©

by

Juan V. Fernandez

2017

Dedication

There are no words to express how grateful I am to everyone that believed in me throughout this journey. This work is dedicated to everyone that believed in me, for their encouragement, love, support, and motivation I am truly blessed to have great individuals in my life. This achievement is especially dedicated to my parents, it is thanks to their hard work and all their lessons that I am at this stage in my life. To my younger brothers, always dream big and remember that you can accomplish anything that you set your mind to.

DEVELOPMENT OF POST-PARETO OPTIMALITY METHODS FOR
MULTIPLE OBJECTIVE OPTIMIZATION

by

JUAN V. FERNANDEZ, B.S.

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

Department of Industrial, Manufacturing and Systems Engineering

THE UNIVERSITY OF TEXAS AT EL PASO

December 2017

Acknowledgements

At this stage in my journey, I find myself reminiscing in the events and everyone and everything that has contributed to my personal and professional development. At this point, I realize that this achievement would have not been possible without the assistance and support of so many individuals. There are certain individuals that deserve my recognition and I would like to take this opportunity to express how grateful to the following individuals:

To begin with, this material is based upon work supported by the National Science Foundation under Grant No. #1060113. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. In addition, to The United States Department of Agriculture for providing me with financial support and all the great opportunities provided for my professional development. I also want acknowledge the Department of Industrial, Manufacturing and Systems Engineering at The University of Texas at El Paso for their support in this journey.

Furthermore, there are certain individuals that deserve my most sincere appreciation. My mentors, Dr. Heidi A. Taboada and Dr. Jose F. Espiritu who found me a little lost during my bachelor's degree and gave me the opportunity to join their research team. It was at that point that I started growing professionally and I cannot express how thankful I am to them for believing in me. It was thanks to their proper guidance and support that I now find myself at the completion of this thesis. I would like to thank the following individuals for their support, teachings, friendship and for all those long days and nights of hard work, Oswaldo Aguirre, Eduardo Castillo, Ileana Delgado, Ana Cram, Olivia Moreno, Claudia Valles, Luis Ramirez and

Jorge Montes. Thank you for your contributions and for helping me grow personally and professionally.

Lastly, I want to take this opportunity to thank my family. To my parents, thank you for everything that you do for me every day. For your guidance for your support, but more importantly for always having faith in me, it is thanks to the way you raised me that I am able to accomplish many great things. To my main motivation, my younger brothers Fernando, Leonardo, and Gabriel, you are the reason I keep setting higher goals every day, thank you for being who you are and for motivating me to never give up.

Abstract

In the past years, multiple objective optimization has been considered, as an important research area since in many real life problems there exists multiple criteria that need to be optimized simultaneously. The use of evolutionary algorithms or metaheuristic methods as solution methodologies lead to a large number of Pareto solutions rather than a single unique optimum. This Pareto-optimal set most of the time tends to be very large and the decision maker now faces the challenge of reducing its size to analyze a feasible number of solutions, thus deciding the best possible solution. In this work, two methods will be introduced for post-Pareto analysis in order to reduce the size of the Pareto-optimal set. The first method is a scalarization method using a Non-uniform weight generator with pseudo-ranking scheme. The second method, the Nash-Dominant Pareto set reduction algorithm, based on Game Theory and the Nash dominance concept. Furthermore, the two methods will be used to reduce the size of the Pareto-optimal set of very popular problems such as DTLZ1 Test Problem, Printed Wiring Board (PWB) Problem, and the Redundancy Allocation Problem (RAP).

Table of Contents

Acknowledgements.....	v
Abstract.....	vii
Table of Contents.....	viii
List of Tables	x
List of Figures.....	xi
Chapter 1: Introduction.....	1
1.1 Background and Problem Description.....	1
1.2 Thesis Objective.....	4
1.3 Thesis Outline	5
Chapter 2: Pareto Optimality	7
2.1 Literature Review.....	8
2.2 Multiple Objective Optimization	10
2.3 Apriori Methods.....	11
2.3.1 Weighted Sum Method	11
2.3.2 Goal Programming.....	12
2.3.3 Multi-Attribute Utility Theory.....	13
2.3.4 Compromise Programming.....	14
2.3.5 Lexicographic Method.....	15
2.4 A posteriori Methods	16
2.4.1 Adaptive Weight Sum Method	16
2.4.2 Normal Boundary Intersection (NBI)	17
2.4.3 Multiple Objective Evolutionary Algorithm (MOEA)	18
2.5 During the search methods.....	20
2.5.1 Light Beam Search (LBS) Method	20
2.5.2 NIMBUS Method.....	21
Chapter 3: A Posteriori Method.....	23
3.1 Non- uniform Weight Generator with pseudo-ranking scheme.....	23
3.2 Case Study 1: DTLZ1 Test Problem.....	27

Results:.....	28
3.3 Case Study 2: Printed Wiring Board (PWB) Problem.....	31
Results:.....	33
3.4 Case Study 3: Redundancy Allocation Problem (RAP)	35
Results:.....	37
Chapter 4: Post-Pareto Optimality using Game Theory	41
4.1 Literature Review.....	42
4.2 Game Theory Concepts.....	44
4.2.1 Steady State / Nash Equilibrium	44
4.2.2 Player Definition	44
4.2.3 Strategy Definition	44
4.2.4 Utility Definition.....	45
4.3 Strategic Game Models.....	45
4.3.1 Definition	46
4.4 Extensive Form Game Models.....	46
4.4.1 Definition	47
4.5 Evolutionary Game Theory.....	48
4.6 Nash-Dominant Pareto Set Reduction Algorithm.....	49
Chapter 5: Post-Pareto Optimality using Game Theory	54
5.1 Case Study 1: DTLZ1 Test Problem.....	54
Results:.....	55
5.3 Case Study 2: Printed Wiring Board (PWB) Problem.....	57
Results:.....	58
5.3 Case Study 3: Redundancy Allocation Problem (RAP)	61
Results:.....	62
Chapter 6: Conclusions	64
Chapter 7: Future Research.....	67
References	69
Vita.....	74

List of Tables

Table 3.2.1: DTLZ1 Pareto-optimal solutions.....	30
Table 3.3.1: PWB Pareto-optimal solutions	34
Table 3.3.1: RAP Pareto-optimal solutions	38
Table 5.3.1: PWB Pareto-optimal set	59
Table 5.3.2: PWB Pareto-optimal set	60

List of Figures

Figure 2.4.3: Multiple Objective Evolutionary Algorithm diagram.	20
Figure 3.2.1: DTLZ1 3-Dimensional graph of each objective.	28
Figure 3.2.2: DTLZ1 Bi-Dimensional graph of each objective.	29
Figure 3.2.3: DTLZ1 Weight distribution graph.	29
Figure 3.2.4: DTLZ1 reduced Pareto-optimal set.	30
Figure 3.3.1: PWB Bi-Dimensional view of the objectives.	33
Figure 3.3.2: PWB Weights Distribution Graph.	33
Figure 3.3.1: RAP 3-Dimensional view of the objectives.	37
Figure 3.3.2: RAP Bi-Dimensional view of the objectives.	37
Figure 5.1.1: DTLZ1 3-Dimensional graph of each objective.	55
Figure 5.1.2: DTLZ1 3-Dimensional graph of each objective.	56
Figure 5.3.1: PWB Bi-Dimensional view of the objectives.	58
Figure 5.3.2: PWB 3-Dimensional graph.	59
Figure 5.3.3: PWB 3-Dimensional graph.	60
Figure 5.3.1: RAP 3-Dimensional graph of each objective.	62
Figure 5.3.2: RAP 3-Dimensional graph of each objective.	63

Chapter 1: Introduction

1.1 Background and Problem Description

Computational optimization or mathematical programming has been recognized as a powerful tool for multi-criteria optimization problems that arise in many different areas. It consists of finding the best solution by maximizing or minimizing some objective function under given constraints in a certain situation. Optimization can be used to improve different systems by maximizing reliability, minimizing cost or other aspects that need to be improved. Since the beginning one single objective was taken in to account to be optimized separately from other objectives, this was not suitable when improving a system due to quality reduction of the objective functions within the system. Optimizing one objective in most cases, does not serve to describe many real life problems because most of the time more than one objective needs to be optimized simultaneously.

In the past decades, multiple objective optimization has emerged as a solution to face many real life applications. Multiple objective optimization problems can emerge in different areas across different disciplines such as biology, economics, and engineering to name a few. In many instances more than one objective needs to be optimized resulting in conflict among the criteria which leads to a large number of solutions rather than a single unique optimum. These situations are noticeable on our daily lives when an individual wants to acquire a product. The individual has many different alternatives to choose from of the same product and the selection of such product can be determined by the color, brand, cost, reliability, etc. In practice, choosing the best solution can result in an arduous task for any individual.

There are two main challenges in multiple objective optimization as seen in Figure 1.1. The first challenge is the generation of multiple trade-off optimal solutions. For this first

challenge a large Pareto-optimal set is generate so decision maker can have many alternatives for solving his/hers problem. Thus, this creates the second challenge and that is to choose one of the obtained optimal solutions for system implementation. The focus of my thesis will be in the second challenge of multiple objective optimization, to reduce the size of the Pareto-optimal set.

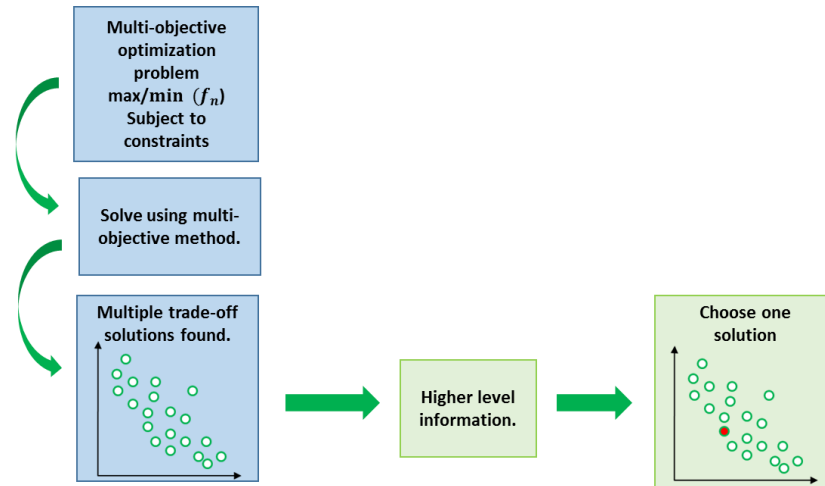


Figure 1.1: Multiple Objective Optimization

Generally, multiple objective optimization can be divided into two categories: Scalarization methods and Pareto Methods. Although different names are used throughout literature to describe these categories, the fundamentals are always the same. The first category is scalarization method; scalarization means replacing a multiple objective optimization problem by a fitting scalar optimization problem (Kasimbeyeli, R., 2013). Examples of scalarization methods include Weighted Sum Approach, Compromise Programming, Utility Theory, Goal Programming, Lexicographic approaches among others (De Weck, O. L., 2004). The second category deals with Pareto methods; ideally, there should be sufficient number of Pareto points to represent the Pareto frontier where each Pareto point represents a solution. A well-distributed

Pareto set is important to obtain the maximum information on the Pareto surface at a minimum computational cost (Erfani, T. & Utyuzhnikov, S. V., 2011). Using this second approach results in a set of non-dominated solutions that is also called the Pareto-optimal set. Examples of Pareto methods include Normal Boundary Intersection, Adaptive Weight Sum method, Evolutionary Algorithms to name a few.

Nevertheless, in the past decades there has been an interest in incorporating techniques from other areas to solve multiple objective optimization problems. Game Theory can better be associated as a branch of economics that studies the interactions between rational decision makers. This is a concept that was derived from studying games (i.e. chess, checkers, and poker) and it became clear that this technique could be applied to all interactions between self-interested agents (i.e. players, decision makers) (Parsons, S. & Wooldridge, M., 2002). In Game Theory, different players take into consideration the decision of the other player and select the best outcome in order to optimize the result.

The analysis of a game begins by specifying the model that describes the game. Depending on the model, a simple model structure may cause to overlook important aspects of the real game, likewise a complicated model structure can obstruct the analysis by overlooking the fundamental issues. For avoiding the above scenarios, there are different forms for representing the games, and the most commonly used games are the extensive game models and the strategic game models. In Extensive game models, the decision-making follows a sequential structure allowing studying different situations where the decision maker is free to change his course of action as events unfold (Osborne, M. J., & Rubinstein, A., 1994). Examples of some extensive games are Tic-Tac-Toe, chess, checkers and poker. On the other hand, strategic game models rather than having a sequential structure is a model where each decision maker selects his

strategy and then all the decision makers move simultaneously. Some examples of these types of games include the prisoner's dilemma, battle of the sexes and rock, paper scissors models.

Moreover, multiple objective optimization and Game Theory problems provide many possible solutions for the decision maker to choose. The solution in multiple objective optimization is represented by the Pareto-optimal set, a set of non-dominated solutions that usually contains a large number of solutions. In Game Theory, the most used concept for choosing the optimal solution is Nash equilibrium or the steady state where no decision maker has an incentive to deviate by changing his strategy. Selecting the best possible solution from different approaches can be cumbersome making the task of the decision maker complicated. For that reason, the focus of this research is the decision making stage more importantly the focus on the post-Pareto optimality stage to reduce the size of the Pareto-optimal set.

1.2 Thesis Objective

The objective of this thesis is to focus on the post-Pareto optimality stage by making use of multiple objective optimization methods to develop efficient decision-making methods in order to reduce the size of the Pareto-optimal set. This thesis will identify trade-offs between different objectives in different scenarios and the performance of the methods will be demonstrated. The main contribution of this work is on the post-Pareto analysis stage by introducing two methods. The Non-uniform weight generator with pseudo-ranking scheme algorithm to expand on the previous work by Carrillo, V. and Taboada, H. in 2012 and the introduction of a Nash-Dominant Pareto set reduction algorithm to reduce the Pareto-optimal set to a feasible size for the decision maker to evaluate.

The performance of these methods will be proven by three different problems. DTLZ1 is a multiple objective test problem with controlled difficulties (Deb, K., 2001) that serve to

identify how the method performs. The second problem that will be address is the scheduling of a Printed Wiring Board (PWB) manufacturing line (Taboada, H. A., & Coit, D. W., 2008). The last problem that will be address is the Redundancy Allocation Problem for series parallel systems (Cao, D. et al., 2013)

Additionally, there are other problems as the ones presented in this thesis that have been solved with different multiple objective approaches. All these approaches have the same complication, being that the post-Pareto stage can result in many possible solutions (sometimes in the thousands) making the decision-making process troublesome.

1.3 Thesis Outline

The present work is divided in to seven chapters. The background and problem description has been provided in chapter 1, providing an overview of multiple objective optimization methods and how Game Theory is being introduced. It also provides an overview of different forms for analyzing Pareto sets. The rest of this work is structured as follows.

Chapter 2 provides a historical review of different multiple objective optimization methods. More importantly, different methods for apriori, a posteriori, interactive methods (during the search), and will provide a review of different applications of such.

Chapter 3, presents a posteriori methods or the post-Pareto optimization stage of this work. It will introduce the non-uniform weight generator with a filtering technique and the performance is tested with three different problems: DTLZ1, scheduling of Printing Wiring Board (PWB), and the Redundancy Allocation Problem (RAP).

Chapter 4 is dedicated to provide a comprehensive review of Game Theory. It will go in depth in describing the types of games that exists. This chapter will define the steady state or Nash equilibrium, and different concepts in literature to identify the best solution for the decision

maker. Furthermore, Nash-Dominant Pareto set reduction algorithm (NDPRA) will be introduced.

Chapter 5 will provide three different case studies to analyze the performance of NDPRA. The test problems are as follows: DTLZ1, the scheduling of Printing Wiring Board (PWB) problem, and the Redundancy Allocation Problem (RAP).

Chapter 6 will serve to make conclusions on the performance of the methods.

Chapter 7 will provide future research on an idea of creating a hybrid algorithm using Game Theory and Fair Division.

Chapter 2: Pareto Optimality

In multiple objective optimization, the selection of the best solution depends on the decision maker preferences and experiences, and this can be achieved by three different ways. The first way is the a priori method; this method incorporates the decision maker preferences before generating the solution points. The second one is a posteriori method; that generates first the solution points or the Pareto-optimal set and then from this set the decision maker decides the best possible solution. The third way is the interactive method or during the search method where the decision maker preferences are incorporated during the search of optimization.

Multiple objective optimization, involves the simultaneous optimization of more than one objective function that leads to conflict among the criteria. Ideally, a set of alternative solutions is wanted, not a single optimal solution. The alternative solutions can be considered optimal solutions because no other solutions within the search space are superior to them taking into account all the objectives. Optimal solutions are solutions that are not dominated by any other solution and these optimal solutions are known as the Pareto-optimal set in Figure 2.1

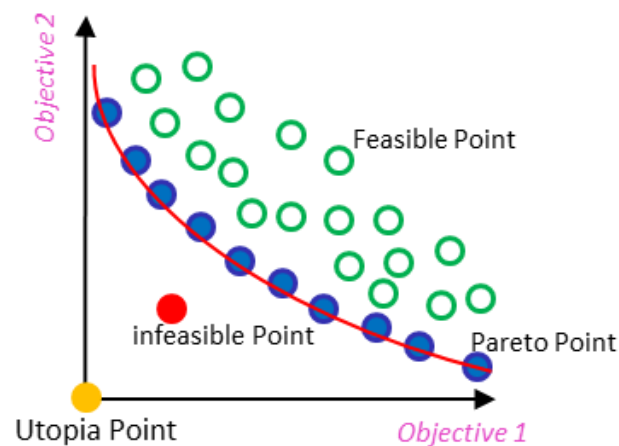


Figure 2.1: Pareto Optimization Visualization

Furthermore, analyzing and evaluating the Pareto-optimal set can be a cumbersome task for the decision maker since some problems can have as many as thousands of alternative solutions. It requires knowledge from the decision maker based on the priorities and assigning weights to each of the objectives. Reducing the size of the Pareto-optimal set can be beneficial to the decision maker in problems where there is a large Pareto-optimal set.

2.1 Literature Review

Throughout literature, many researches have been identifying different optimization methods to make the decision making stage less difficult. In (Pinchera, D. *et al*, 2017) introduces the function named Quantized Lexicographic Weighted Sum (QLWS) built on the definition of a Global Cost Function (GCF) for avoiding the evaluation of the Pareto front solutions being that it is a very time consuming task. To translate priorities correctly among the different targets the GCF was built by requesting the decision maker to define priorities among targets and constraints. In (Reynoso-Meza, G. *et al.*, 2010) a multiple objective optimization (MOO) can be used to solve constrained single objective optimization problems. It uses Differential evolution and spherical pruning is introduced. Spherical Pruning is less sensitive to losing non-dominated solutions and it works as if the designer was standing in the ideal solution, with a given direction in the objective space, he will be searching for the non-dominated solutions with the best constraint tradeoff. As discussed by (Gong D., *et al.*, 2014.) interval multiple objective optimization problems (IMOPS) have few theories since they are complicated in practical applications. The goal of an IMOP is to find the decision maker's most preferred solution, in this case the authors utilized an interactive method. The study employs NSGA-II in which the decision maker inputs the importance relations among the objectives before the evolution. Then based on the relations the corresponding mathematical model is reduced.

There have been other methods for reducing the size of the Pareto-optimal set such as, in (Carrillo, V. and Taboada, H., 2012) a non-uniform weight generator method is introduced to prune Pareto-optimal set obtained by multiple objective genetic algorithms. (Taboada, H. and Coit, D., 2008) introduces a multiple objective genetic algorithm using two approaches to filter the Pareto optimal set: pruning using non-numerical ranking preferences and data clustering. (Messac, A. and Mattson, C. A., 2004) introduces normal constraint method to generate evenly distributed Pareto points by reducing the feasible design space and chooses a sequence of reductions and optimization to obtain Pareto solutions. (Mattson, C. A., et al., 2004) introduces smart Pareto filtering to control degree of freedom and the Pareto set size. (Syu, Y., et al., 2012) uses genetic algorithm using prioritize objective functions. In (Venkat, V., et al., 2004) a greedy reduction algorithm is introduced to allow the decision maker to choose the size of a subset where a small subset is likely to contain Pareto optima.

In (Soltani, R. et. al. 2015), the study utilizes compromise programming to maximize reliability and minimize non-linear cost of the system simultaneously, and one consideration being done is the distribution of the weight components within the subsystems as another form of entropy. The model being use in this study for the first time maximizes reliability, maximizes entropy and minimizes the cost simultaneously. In (Cao, 2013 et al.), utilizes exact efficient Pareto set generation method to identify all the non-dominated solutions. Furthermore the proposed method was compare with NSGA-II were NSGA-II failed to identify all Pareto-Optimal points causing for the proposed method to be more efficient. (Kulturel-Konak S. et. al., 2008) solved a redundancy allocation problem which uses TS meta-heuristic to generate a Pareto optimal set and Monte-Carlo simulation where random weights are generated from uncertain weight function.

2.2 Multiple Objective Optimization

Many real life problems have several objectives to be optimized resulting in conflict. Ideally, the most efficient way to optimize these scenarios is by simultaneously optimizing all the objectives in regards to the problem. To better understand how multiple objective optimization works, it is important to understand how single-objective optimization functions. In single-objective optimization, the aim is to find an optimum solution. Within the search space, there may exist many local optimal solutions, single-objective optimization aims to always finding the global optimum solution. In single-objective optimization, an acceptable solution is one with the best objective function value (Deb, K., 2001).

Basic single-objective optimization problems can be mathematically written as follows:

$$\text{Minimize } f(x)$$

$$x \in S$$

Where, f is a scalar function and $S = \{x \in R^m: h(x) = 0, g(x) \geq 0\}$ defines the set of constraints.

However, multiple objective optimization aims to progress towards the Pareto-optimal front and it is essential to maintain diversity between the solutions. In multiple objective optimization, all the objectives are important for that reason it is essential to have a diverse set of solutions that are close to the Pareto-optimal front, this will result in a variety of optimal solutions. Maintaining diversity between a diverse set of solution and emphasizing convergence near the Pareto-optimal front is a dual task that makes multiple objective optimization more difficult than single-objective optimization. Multiple objective optimization, involves simultaneously optimizing several objectives that are often competing objectives and can be mathematically expressed as follows:

Minimize/Maximize $f_i(x)$ for $i = 1, 2, \dots, n$

Subject to

$$g_j(X) \leq 0, \quad j = 1, 2, \dots, J$$

$$h_k(X) \leq 0, \quad k = 1, 2, \dots, K$$

Where, there are n objective functions to be minimized or maximized, the parameter x is a p dimensional vector that has p decision variables. The solutions to a multiple objective optimization problem are expressed in terms of non-dominated solutions. For instance consider vectors a and b . In a minimization problem $f_i(a)$ dominates $f_i(b)$ when, $f_i(a) \leq f_i(b)$ for all i and $f_i(a) < f_i(b)$ for at least one i . In maximization problem $f_i(a)$ dominates $f_i(b)$ when, $f_i(a) \geq f_i(b)$ for all i and $f_i(a) > f_i(b)$ for at least one i .

2.3 Apriori Methods

Scalarization methods are based on different assumptions, one assumption is that the decision maker knows his preferences before finding the design solutions; second assumption is that the objectives can be combined to indicate a dimensionless scalar amount that expresses how good a particular solution is. The following approaches combine all the objective functions into a single-objective.

2.3.1 Weighted Sum Method

One of the most common approaches to multiple objective optimization is the weighted sum method, it is mathematically expressed as follows (Augusto, O. B. *et al*, 2012):

$$\text{minimize:} \quad \sum_{i=1}^k w_i f_i^s(X)$$

Subject to:

$$X \in S$$

$$w_i \geq 0, \quad \sum_{i=1}^k w_i = 1$$

Where k represents the total number of objectives i , and the decision maker preferences are expressed by the weights w_i .

The combination of multiple objectives into a single objective by means of aggregation is one of the most intuitively ways for optimization. The only main drawback of this method, is deciding the best weighting coefficients for each of the objectives and it can become a cumbersome task for the decision maker when he has to prioritize one objective over the other. This can become especially difficult when there are many objectives to optimize. In addition, research in the generation of weights has been proposed by different researchers in order to alleviate the decision making when prioritizing objectives.

2.3.2 Goal Programming

The following methods was originally developed by (Charnes, A., & Cooper, W. W., 1977). This method is a preference base approach that requires the decision maker to set goals for all the objectives. The best design solution is one that minimizes the weighted sum of deviations from the goals. This model separates the values into positive and negative parts that represent achievement and under achievement respectively, where achievement means that the goal has been reached. Goal programming can be formulated as follows:

$$\text{minimize: } \sum_{i=1}^k w_i (d_i^+ + d_i^-)$$

Subject to:

$$f_i(X) + d_i^+ + d_i^- = b_i, \quad i = 1, 2, \dots, k$$

$$d_i^+ \geq 0, d_i^- \geq 0 \quad i = 1, 2, \dots, k$$

$$d_i^+ d_i^- = 0 \quad i = 1, 2, \dots, k$$

$$w_i \geq 0$$

$$X \in S$$

Where b_i represents the goals for all the objectives f_i , underachievement is denoted by deviation d_i^- and achievement is denoted by deviation d_i^+ allowing the decision maker to assign weights w_i and to define achievement goal levels.

Goal programming main advantage is its capability of handling large scale problems, this method can produce an infinite number of alternatives, however it lacks the ability to weight coefficients and this is one major disadvantage since typically this method is used in combination with other multi-criteria decision making methods to weight coefficients (Velasquez, M., & Hester, P. T., 2013). Another disadvantage for this method is that it does not guarantee obtaining a Pareto-optimal solution.

2.3.3 Multi-Attribute Utility Theory

This method provides the basis for expressing decision between different alternatives in which the consequences are characterized by multiple attributes. This approach allows for the comparison of many measures by rescaling numeric values on the scale 0 to 1 where 0 is the worst preference and 1 is the best. Utility expresses the satisfaction each attribute provides to the decision maker and the result is a rank evaluation order of the possible alternatives of the decision maker preferences.

The utility function can be express as the sum of individual utilities, expressed as follows:

$$U(X) = \sum_{i=1}^k U_i(X_i), \quad i = 1, 2, \dots, k$$

The most common form of multi-attribute utility function is additive and can be expressed as follows (Regier, D. A., & Peacock, S., 2017).

:

$$U(X) = \sum_{i=1}^k w_i U_i(X_i), \quad i = 1, 2, \dots, k$$

Where w_i is the scaling constant such that $\sum_{i=1}^k w_i = 1$.

The advantages of multi-attribute utility theory are that it allows to use deterministic and stochastic decision environments, takes uncertainty into account and it allows for the incorporation of preferences. Major disadvantage of this method is that it requires a lot of input at every step of the procedure to record the preferences of the decision maker. This method is extremely data intensive since the decision of the decision maker need to be precise and can be subjective.

2.3.4 Compromise Programming

Compromise programming, is a multi-criteria decision-making approach considered as a complement to multiple objective optimization problems. This method allows for the reduction of the Pareto to a reasonable size. The basic idea of this method is to identify an ideal point (Zeleny, M., 1973), this ideal point serves as a reference for the decision maker, and this ideal point is the only assumption made in this method. In order to identify the set of solutions that are closest to the ideal point, the distance concept is introduced General formulation for the metrics can be expressed as follows:

$$L_j = \left[\sum_{i=1}^n (w_i d_i)^p \right]^{\frac{1}{p}}$$

Where, L_j is the distance metric for each alternative, the preferences of the decision maker are denoted by the weights w_i of the i^{th} objective and d_i denotes the normalized differences between i^{th} objectives to the ideal point and p serves as a weight for the deviation according to their magnitude. The simplicity of the method makes it an advantage, whereas a disadvantage is that it can result in corner solutions or extreme solutions.

2.3.5 Lexicographic Method

Lexicographic can be defined as the total order of the sequence of objectives. The main concept of this method is that it requires the decision maker to rank the objectives in order of importance. In this approach if the objective ranked as the best by decision maker has a unique solution no further optimization is need, if that is not the case optimize the second preferred objective but at the same time the objective rank as best maintains its optimal value. For instance a mathematical representation is as follows (Hwang, C. L., & Masud, A. S. M., 2012):

$$\max f_1(x)$$

Subject to:

$$g_j(x) \leq 0, \quad j = 1, 2, \dots, n$$

If objective f_1 has a unique solution f'_1 this solution is favored for the entire problem. If that is not the case than the second ranked objective is solved as follows:

$$\max f_2(x)$$

subject to:

$$g_j(x) \leq 0, \quad j = 1, 2, \dots, n$$

$$f_1(x) - f'_1$$

Where f'_2 is the solution, if f'_2 has a unique solution the solution is favored for the entire problem. Otherwise, the procedure is repeated until all the objectives have been considered. If

the unique solution is found at the i^{th} objective, the solution will be chosen and there is no need to compute the less ranked objectives.

The main drawback of this method is that the decision maker has to rank the objectives, and this approach is very sensitive to the ranking of the objectives, that is why the decision maker should exercise caution when two objectives are close in terms of importance.

2.4 A posteriori Methods

In a posteriori methods, a representation of the Pareto-optimal set is generated and the decision maker has to choose the best possible solution from that set. The following are examples of methods that are used in the decision making stage.

2.4.1 Adaptive Weight Sum Method

Adaptive Weight-Sum (AWS) Method is a method that was developed to solve the issues of using the traditional weight-sum method, which, methodically, changes weights across the objective functions with the intention to find one by one a Pareto-optimal solution. The Adaptive Weight-Sum method was introduced by Kim, I. Y. & de Weck, O. L. I. in 2004, and is intended to focus on exploring uncharted regions by changing the weights adaptively rather than by using a priori weight selection and by specifying additional inequality constraints. The Adaptive Weight-Sum Method can be formulated as follows:

$$\text{Minimize: } \lambda \frac{J_1(x)}{sf_{1,0}(x)} + (1 - \lambda) \frac{J_2(x)}{sf_{2,0}(x)}$$

Subject to:

$$J_1(x) \leq P_1^* - \delta_1$$

$$I_2(x) \leq P_2^y - \delta_2$$

$$h(x) = 0$$

$$g(x) \leq 0$$

$$\lambda \in [0,1]$$

Where, λ represents the uniform step size of the weighting factor, δ_1 and δ_2 are the offset distances selected by the user, P_i^x and P_i^y are the x and y positions of the i th endpoint and $I_{1,0}$ and $I_{2,0}$ are scaling factors.

The Adaptive Weight-Sum Method produces evenly distributed solutions, finds Pareto optimal solutions in non-convex regions, and neglects non-Pareto optimal solutions in non-convex regions. The Adaptive Weight-Sum Method is an extension of the ordinary weight-method, the adaptive weighted-sum method is known to perform adequately when it is applied in multidimensional multiple objective optimization problems, meaning to problems where there are more than two objectives functions. The Adaptive Weight-Sum Method still needs to be tested with problems with a higher dimensionality to fully see its optimization potential, (Kim, I. Y., & De Weck, O. L., 2006) and (Kim, I. Y., & de Weck, O. L., 2005).

2.4.2 Normal Boundary Intersection (NBI)

The Normal Boundary Intersection (NBI) Method is a method developed by Das and Dennis (1998) and it is used to generate the Pareto-optimal front in non-linear multiple objective optimization problems. One of its biggest advantages over other common multiple objective approaches, is that it successfully creates an evenly distributed set of solutions across the Pareto front. Normal Boundary Intersection's mathematical formulation is written as follows:

$$\begin{aligned}
& \text{Min } \bar{f}_1(x) \\
& \text{Subject to:} \\
& \bar{f}_1(x) - \bar{f}_2(x) + 2w - 1 = 0 \\
& g_j(x) \geq 0 \\
& 0 \leq w \leq 1
\end{aligned}$$

Where $\bar{f}_1(x)$ and $\bar{f}_2(x)$ are the normalized objective functions, $g_j(x) \geq 0$ and $0 \leq w \leq 1$ are the set of constraints for experimental region and the cuboidal region, respectively. This mathematical formulation allows you to find the Pareto-optimal set, and it is up to the user to define which solutions are more feasible and appropriate than others.

When Normal Boundary Intersection is compared to the Weighted Sum Method, it can be confirmed that only the Normal Boundary Intersection method was able to generate convex and equally spaced Pareto frontiers. Normal Boundary Intersection has been used in different areas for instance, environmental and economic hydrothermal self-scheduling (Ahmadi, A., et al., 2015), resource scheduling of renewable energy based on micro grids (Izadbakhsh, M., et al., 2015), and machining process with control and noise variables (Brito, T. G., et al., 2014), just to mention a few. Other areas where Normal Boundary intersection has been tested yield superior results when compared to other multiple objective optimization methods (Das, I., & Dennis, J. E., 1998), and (Costa, D. M., et. al., 2016).

2.4.3 Multiple Objective Evolutionary Algorithm (MOEA)

Since the year 1985, there has been an increasing interest of research in evolutionary algorithms fitted for multiple objective practices. Multiple objective evolutionary algorithms, developed with the purpose to solve multiple objectives of a problem in a single run. There are

an extensive number of diverse methods that are MOEA, and in general, they mainly only differ in the fitness evaluation phase. Some of the available multiple objective evolutionary algorithms are: Multiple Objective Genetic Algorithms, Niched Pareto Genetic Algorithm (NPGA) introduced by Horn, Nafpliotis, and Goldberg (Horn, J., et. al., 1994). Strength Pareto Evolutionary Algorithm (SPEA) introduced by Zitzler and Thiele (Zitzler, E., et. al. 2001), Cultural Algorithm with Evolutionary Programming (CAEP) (Coello, C. A. C, and Becerra, L. R., 2003). Other evolutionary algorithms include, Vector Evaluated Genetic Algorithm (VEGA), Non-dominated Sorting Genetic Algorithm (NSGA) to name a few.

Each of them has their own fitness methodology to determine the best set of solutions to a given problem. General multiple objective evolutionary algorithms have the following formulation:

$$\min / \max \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_n(\mathbf{x}))$$

Subject to:

$$\mathbf{x} = (x_1, x_2, \dots, x_m) \in X$$

$$\mathbf{y} = (y_1, y_2, \dots, y_m) \in Y$$

Where, \mathbf{x} is called the decision vector, X is the parameter space, \mathbf{y} is the objective vector, and Y is the objective space. In addition, the set of solutions obtained from applying a Multiple Objective Evolutionary Algorithm, consists of all decision vectors whose corresponding objective vectors cannot be improve in any dimension without downgrading in another. These sets of solutions are called non-dominated solutions and together they compose the Pareto-optimal front.

Figure 2.4.3 shows the flow diagram of a Multiple Objective Evolutionary Algorithm. Most of all MOEA's follow the same steps, perhaps, the only variation would be that each has their own method to perform the fitness evaluation step.

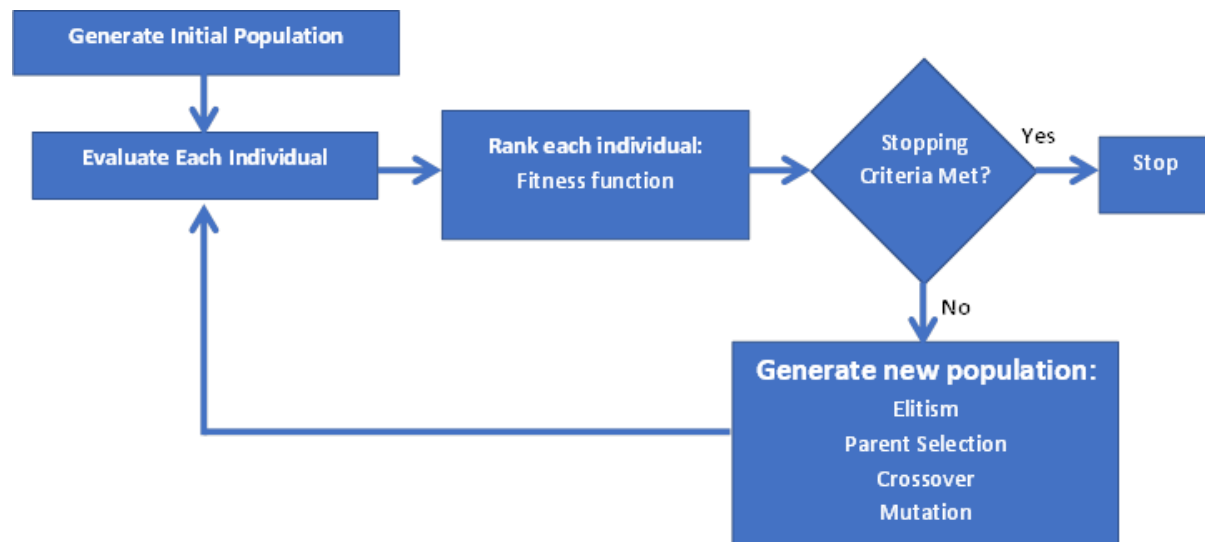


Figure 2.4.3: Multiple Objective Evolutionary Algorithm diagram.

2.5 During the search methods

There are some points in an optimization model where the decision maker is able to interact in the analysis in order to find the solutions, there are different types of interaction but the main goal of these methods is to allow the decision maker to be part of the analysis of finding the solution. These types of methods can also be call during the search methods. A few examples of the methods are provided below.

2.5.1 Light Beam Search (LBS) Method

The Light Beam Search method was first introduced by (Jaszkiewicz, A., & Slowiński, R., 1995). Different optimization investigations have been performed using the LBS method, in

detail, there is the Linear Antenna Array Optimization (An, S. et. al. 2017), the Inverse Problem Optimization (An, S. et. al., 2016), among other methods and the main concept of this method is that it enables an interactive analysis in the decision of a multiple objective problem due to the amount of non-dominated solutions in the Pareto-optimal front, to the decision maker in each iteration. The decision maker supplies two points: aspiration point and a reservation point. The points determine the direction of a search during an iteration. The decision maker has the ability to control the search by performing some modifications while selecting the aspiration and reservation points. The decision maker also has the possibility to shift the current point to a selected better point from its neighborhood. It is essential these two points are provided by the decision maker, in case they are not suggested, any point in the plain, including the ideal point can be advocated as the aspiration or the reservation points. Wierzbicki's scalarizing achievement function is utilized to determine the aspiration point on to the Pareto-optimal front, a non-dominated middle point is selected (Wierzbick, A. P., 1997). An outranking relation is used as a local preference model in a neighborhood of the current point. To define the outranking selection, the decision maker needs to specify three preference thresholds: indifference threshold, preference threshold and veto threshold. All solutions found must outrank and be indifferent or incomparable to the middle point.

2.5.2 NIMBUS Method

The Non-differentiable Interactive Multiple objective Bundle-based Optimization System (NIMBUS) method is an interactive method used for non-differentiable multiple objective optimization problems. NIMBUS was first introduced by (Miettinen, K. and Makela, M. M., 1995) (Miettinen, K. and Makela, M. M., 1997), and the uniqueness of the method is that it

allows the decision maker to analyze the objective functions during each iteration and indicate what kinds of improvements are coveted. During each iteration, the decision maker will classify the objective functions into up to five different categories:

- Should be improved
- Should be improved down until some aspiration level
- Are satisfactory at the moment
- Are allowed to increase up until some upper bound
- Are allowed to change freely

Where, the decision maker will determine the aspiration levels. In addition, the decision maker can also introduce a weighting coefficient to the objective function. Once the decision maker has classified the objective functions, the he or she must define how many solutions wants to compare. From x number of solutions, the decision maker can select any of the solutions as the final solution or as the starting point for a new classification. It is also possible to generate a search asking for intermediate solutions found from two promising solutions. It is important to note that this process is not irrevocable; if the solutions obtained by the decision maker are not what expected, the decision maker has the liberty to explore intermediate points and find a better solution for the problem.

Chapter 3: A Posteriori Method

In a posteriori method a representation of the Pareto-optimal set is first generated, afterwards the decision maker needs to decide a solution from that set. This type of methods are intended to alleviate that part of the decision making stage by reducing the Pareto-optimal set to a more feasible amount of solutions, there can be cases when there are more than a thousand solutions from which the decision maker needs to choose one solution for system implementation. It can be difficult to visualize the Pareto-optimal set in cases where there are more than two objectives, and presenting the Pareto-optimal set to the decision maker is another challenge to consider.

3.1 Non- uniform Weight Generator with pseudo-ranking scheme

The following approach was developed in (Carrillo, V. & Taboada, H., 2012). This method uses a non-uniform weight generator to reduce the size of the Pareto-optimal set. The distributed weights are used as the basis for non-numerical ranking weights scalar function $f' = w_1f_1 + w_2f_2 + \dots + w_nf_n$. The method is expressed as follows:

$$f' = w_1f_1 + w_2f_2 + \dots + w_nf_n$$

To generate the weights w_k , a sequence of $0 < x_1 < x_2 < \dots < x_n < n$ positive values is produce from X_1, \dots, X_n random uniformly distributed variables $X_i \sim U(i-1, i)$ is produce where the general iterative formula to solve for x_k is expressed by:

$$x_k = u_k + (k-1) \quad \text{for } k = 1, 2, \dots, n$$

Where u_k is a randomly generated uniform distribution. Then a non-uniformly distributed increasing sequence of weights is generated by calculating:

$$M = \sum_{k=1}^n x_k \quad \text{and} \quad w_k = \frac{x_k}{M}$$

Where, the sequence of weights $w_1 < w_2 < w_3 < \dots < w_n$ is acquired from $w_k = \frac{x_k}{M}$.

The mathematical formulation is as follows:

For X_1, \dots, X_n random uniformly distributed variables, $X_i \sim U(i-1, i)$ the corresponding density functions are shown by equation (1):

$$f(x_i) = \begin{cases} 1, & x_i \in (i-1, i) \\ 0, & \text{otherwise} \end{cases} \quad \text{for } i = 1, \dots, n \quad (1)$$

Where, X_1, \dots, X_n are independent and equation (2) shows its joint probability density function:

$$f(x_1, x_2, \dots, x_n) = \prod_{i=1}^n f(x_i) = \begin{cases} 1, & x_i \in (i-1, i) \\ 0, & \text{otherwise} \end{cases} \quad \text{for } i = 1, \dots, n \quad (2)$$

Furthermore, multiple integration proves that the joint function in equation (2) is a probability density function since its summation equals to one,

$$\int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, x_2, \dots, x_n) dx_1 dx_2 \dots dx_n = \int_{n-1}^n \dots \int_1^2 \int_0^1 dx_1 dx_2 \dots dx_n = 1$$

For X_i values such that $X_1 < X_2 < \dots < X_n$ the marginal probability density function (p.d.f) and cumulative distribution (c.d.f) must be calculated for each X_i . According to equation (1) the probability density function for X_1 is,

$$f(x_1) = \begin{cases} 1, & x_i \in (0, 1) \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Its cumulative distribution function is $F(x_1) = \int_0^x ds = x_1$. The probability integral transformation theorem $F(x_1) = u_1$ has a uniform distribution $u \sim U(0, 1)$. Therefore the first is $x_1 = u_1$. In order to calculate for x_2 given that x_1 is already known, the conditional probability density function must be calculated as shown in equation (4):

$$f(x_1|x_2) = \frac{f(x_1, x_2)}{f(x_1)} = \begin{cases} 1, & \text{if } x_i \in (i-1, i) \text{ for } i=1,2 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The same way x_1 was obtained; the cumulative distribution function of x_2 given that x_1 is known must be calculated as follows:

$$F(x_1|x_2) = \int_1^{x_2} ds = x_2 - 1 = u_2 \quad \text{where } u_2 \in U(0,1).$$

Solving for x_2 , the second value $x_2 = u_2 + 1$ is obtained to be included in the increasing sequence of x_2 values needed to construct the collection of weights for the composite function:

$$f' = w_1 f_1 + w_2 f_2 + w_3 f_3 + \dots + w_n f_n$$

Since this is an iterative procedure for the k -th case the cumulative distribution function is:

$$F(x_k|x_1, x_2, \dots, x_{k-1}) = \int_1^{x_k} ds = x_k - (k-1) = u_k \quad \text{where } u_k \in U(0,1).$$

Solving for x_k we get the general iterative formula equation (5) for each one of the x_k values:

$$x_k = u_k + (k-1) \quad (5)$$

For $k = 1, 2, 3, \dots, n$ the x_k sequence obtained is as follows:

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

For x_i values, the weighting search generates a sequence of weights between (0,1).

Let $M = \sum_{k=1}^n x^k$, and then we can obtain a non-uniformly increasing sequence of weights $\{w_k\}_{k=1}^n$ such that $0 < w_1 < w_2 < \dots < w_n < 1$ and $\sum_{k=1}^n w_k = 1$ as shown in equation (6).

$$w_k = \frac{x_k}{M} \text{ for } k = 1, \dots, n \quad (6)$$

Furthermore, once the generation of the weights is complete, f' will be calculated and introduced into an algorithm for filtering the solution. Where, the solution that yields the minimum value for f' is assigned a 1 the rest of the solutions are assigned a 0. The process

repeats for other weight sets for as many as k times, and the counter keeps increasing every time the solution is assigned a 1. At the end, all solutions that have non-zero values will be the ones that become the pruned Pareto-Optimal set. Based on the above expressions the algorithm for the non-uniform weight generator method can be express as follows:

Pseudo code for Non-uniform weight generator with pseudo-ranking scheme algorithm

START

Determine n

1. Randomly generate $u_1 \in (0,1)$
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 $k = 1, \dots, n$
6. Finally calculate $M = \sum_{k=1}^n x_k$ and $w_k = \frac{x_k}{M}$ for $k = 1, \dots, n$

After w_k is obtained

7. Convert all objectives to maximization
8. Normalize objectives
9. Sum weighted objectives to form $f' = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4$
10. Find the solution with the maximum value
11. Increase the counter that corresponds to that solution
12. Repeat steps (1-11) several thousand times
13. Identify the pruned Pareto optimal set (counter > 0)

END

3.2 Case Study 1: DTLZ1 Test Problem

Throughout literature, researches have created different test functions to identify the suitability of different algorithms to different scenarios. Arguably, these test functions may not be appropriate for inclusion into multiple objective evolutionary algorithms, since explanations are rarely offered for the specific problem. Yet, they are highly used by many researches to understand the performance of the algorithms due to its simplicity in the sense of finding the optimal solution. The following test problem is from a popular set DTLZ, in most cases these problems can be adjusted to make them more difficult by using different variables. The general methodology for formulating these examples can be found in (Abraham, A., & Jain, L., 2005). The following formulation, and Pareto-optimal sets were obtain from (Coello, CAC., n.d.).The problem is formulated as follows:

$$\text{Minimize } F = (f_1(X), f_2(X), f_3(X))$$

Where,

$$f_1(X) = \frac{1}{2}x_1x_2(1 + g(X))$$

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

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

and

$$g(X) = 100 \left[10 + \sum_{i=3}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right]$$

Subject to:

$$n = 12 \quad 0 \leq x_i \leq 1 \quad i = 1, \dots, 12$$

Moreover, several assumptions were made in this case study and those are that the decision maker has no knowledge about the priorities of the problem: he does not know how to generate the weights, nor the specific weight values. A set of $k=5000$ weights will be generated and the solution where the counter is greater than zero will be chosen as the Pareto-optimal solution.

Results:

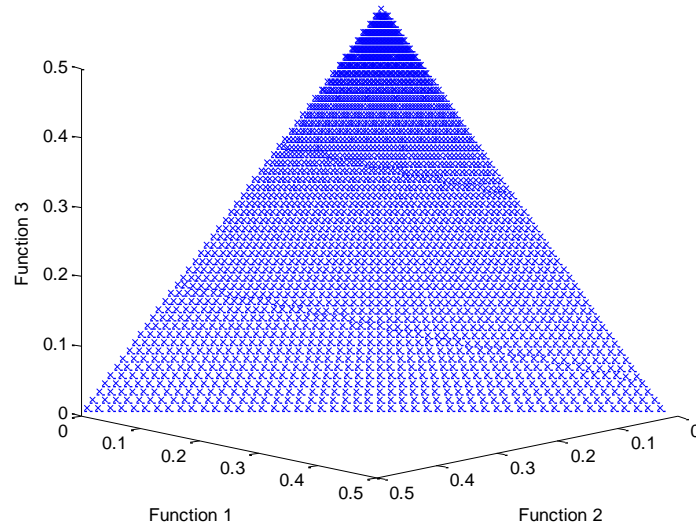


Figure 3.2.1: DTLZ1 3-Dimensional graph of each objective.

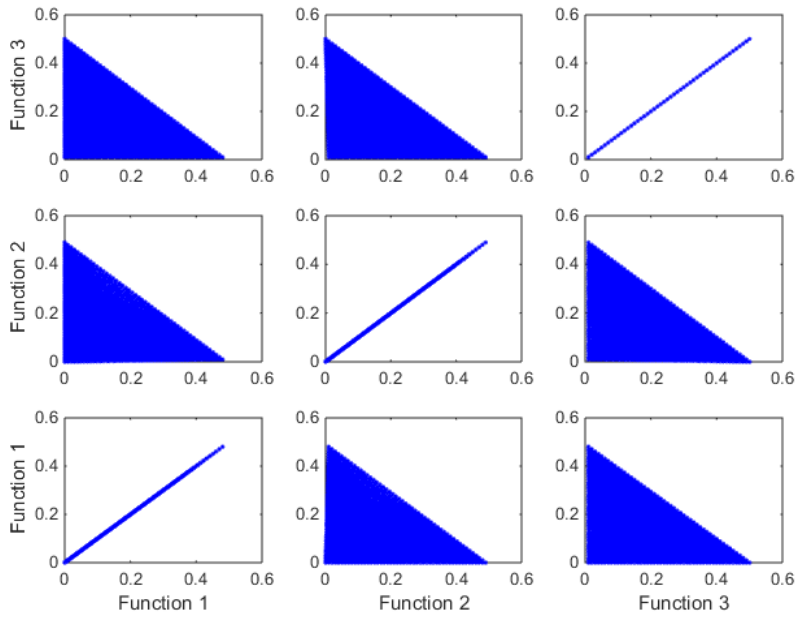


Figure 3.2.2: DTLZ1 Bi-Dimensional graph of each objective.

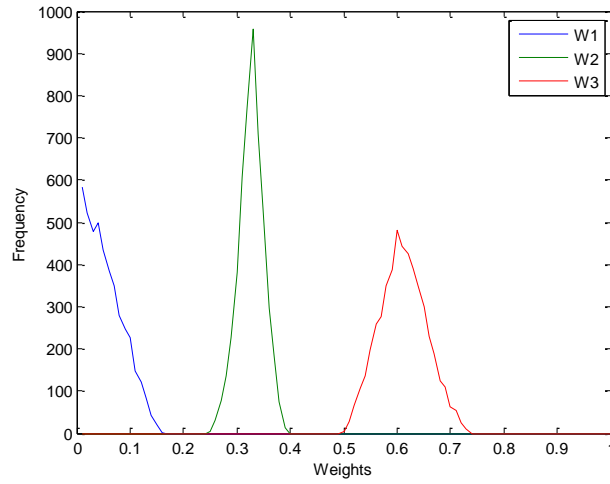


Figure 3.2.3: DTLZ1 Weight distribution graph.

Figure 3.2.1 is the Pareto-optimal set that the mathematical model generates, that accounts for a total of 2,500 non-dominated solutions, and it serves to visualize the three dimensional space. Figure 3.2.2 is a two dimensional graph of all the objective functions

compare to one another. Figure 3.2.3 shows the frequency of the 5,000 weight sets that were generated for evaluating our scalar function f' . Based on our pseudo-ranking scheme the following results observed in Table 3.2.1 were obtained:

Table 3.2.1: DTLZ1 Pareto-optimal solutions

Importance order	Solution #	f1	f2	f3	Counter
f1 < f2 < f3	2500	0.4802	0.0098	0.01	5000
f1 < f3 < f2	2500	0.4802	0.0098	0.01	5000
f2 < f1 < f3	2451	0	0.49	0.01	5000
f2 < f3 < f1	2451	0	0.49	0.01	5000
f3 < f2 < f1	50	0	0	0.5	5000
f3 < f1 < f2	50	0	0	0.5	5000

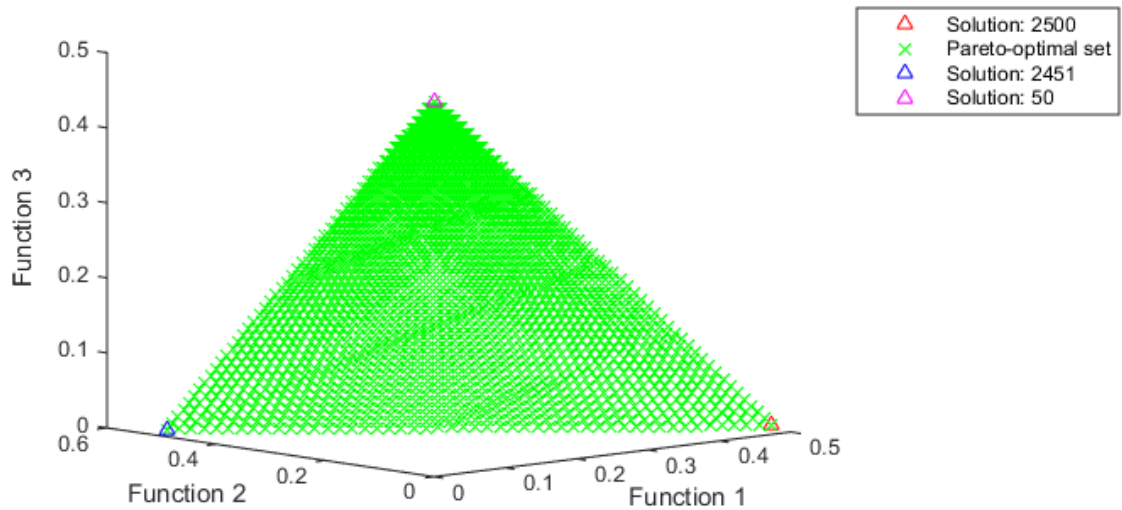


Figure 3.2.4: DTLZ1 reduced Pareto-optimal set

In Table 3.2.1 the first columns shows the importance order of the objective functions, where in the first rows function 1 is less important than function 2 and less important than function 3. This table provides all the possible combinations in the event that the decision maker does not know the priorities of the problem. In this scenario three solutions were obtained, and can be observed in Figure 3.2.4, based on the counter the three solutions were the best in their respective importance order, this is due on how the weights are generated where weight one has a very low value compare to weight two and three. Then, from the original Pareto-optimal set of 2,500 solutions the Non-uniform weight generator with pseudo-ranking scheme algorithm reduces the size to only one solution.

3.3 Case Study 2: Printed Wiring Board (PWB) Problem

The next problem is a popular real world scenario, the scheduling of a Printed Wiring Board (PWB) manufacturing line in (Taboada, H. A., & Coit, D. W., 2007). This problem seeks to minimize overtime f_1 , minimize average finish time f_2 , minimize the variance of the finish time f_3 and minimize the total cost f_4 . The model formulation is expressed as follows:

$$f_1 = \min \sum_{i=1}^m O_i, \quad f_2 = \min \mu_c, \quad f_3 = \min \sum_{i=1}^m \frac{(C_i - \mu_c)^2}{m}, \quad \text{and} \quad f_4 = \min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

Subject to:

$$\sum_{i=1}^m x_{ij} = 1$$

$$O_i = \max \left(\sum_{j=1}^n p_{ij} x_{ij} - T, 0 \right)$$

$$C_i = \sum_{j=1}^n p_{ij} x_{ij}$$

$$\mu_c = \frac{\sum_{i=1}^m C_i}{m}$$

Where,

$$x_{ij} = \begin{cases} 1, & \text{if lot } j \text{ is assigned to machine } i \\ 0, & \text{otherwise} \end{cases}$$

Where, O_i is the overtime of machine i , n is the number of lots to schedule, m is the number of parallel machines, p_{ij} is the processing time of lot j on machine i , c_{ij} is the cost of processing a lot j on machine i , and T is the lot release interval times.

Similarly, as in case study 1 several assumptions are made in this case study; that the decision maker has no knowledge about the priorities of the problem: he does not know how to generate the weights, nor the specific weight values. A set of $k=5000$ weights will be generated and the solution where the counter is greater than zero will be chosen as the Pareto-optimal solution.

Results:

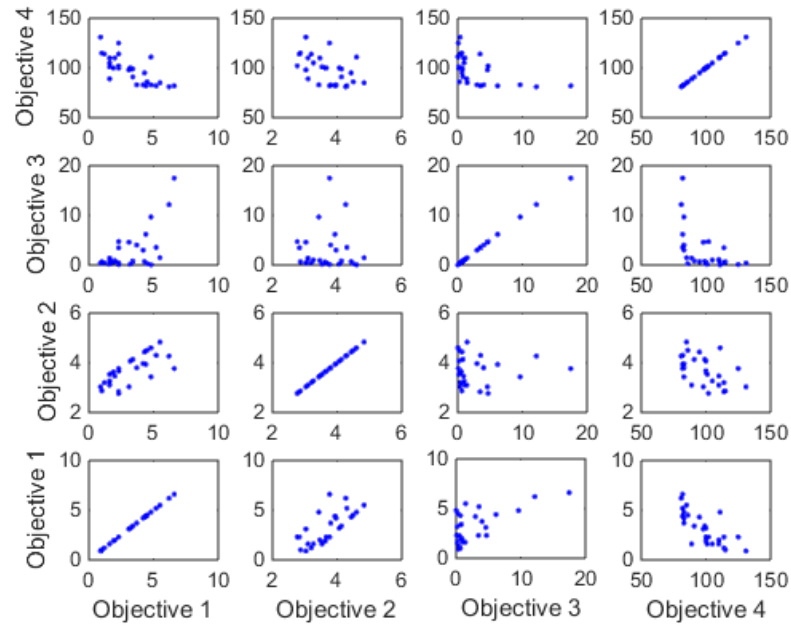


Figure 3.3.1: PWB Bi-Dimensional view of the objectives.

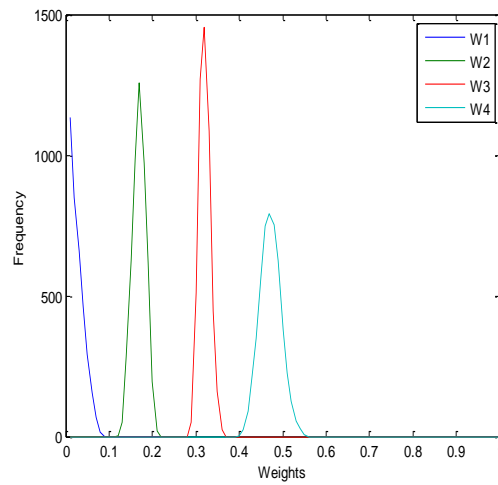


Figure 3.3.2: PWB Weights Distribution Graph.

Figure 3.3.1 is the Pareto-optimal set that the mathematical model generates, that accounts for a total of 28 non-dominated solutions, and it serves to visualize the interaction of

the objectives to each other, the visualization is in the two dimensional space. Figure 3.3.2 shows the frequency of the 5000 weight sets that were generated for evaluating our scalar function f' . Based on our pseudo-ranking scheme the following results observed in Table 3.3.1 were obtained:

Table 3.3.1: PWB Pareto-optimal solutions

Importance order	Solution #	f1	f2	f3	f4	Counter
f1 < f2 < f3 < f4	28	6.6	3.76667	17.4956	82	5000
f1 < f2 < f4 < f3	13	2.3	3.76667	0.1089	125	3843
	24	4.8	4.6	0.0467	111	1157
f1 < f3 < f2 < f4	27	6.2	4.26667	12.1756	81	4692
	26	5.5	4.83333	1.4689	85	308
f1 < f3 < f4 < f2	1	0.9	3.03333	0.3889	131	5000
f1 < f4 < f3 < f2	28	6.6	3.76667	17.4956	82	5000
f1 < f4 < f2 < f3	24	4.8	4.6	0.0467	111	5000
f2 < f1 < f3 < f4	28	6.6	3.76667	17.4956	82	5000
f2 < f1 < f4 < f3	13	2.3	3.76667	0.1089	125	2829
	24	4.8	4.6	0.0467	111	1770
	1	0.9	3.03333	0.3889	131	401
f2 < f3 < f1 < f4	28	6.6	3.76667	17.4956	82	5000
f2 < f3 < f4 < f1	1	0.9	3.03333	0.3889	131	5000
f2 < f4 < f3 < f1	28	6.6	3.76667	17.4956	82	5000
f2 < f4 < f1 < f3	24	4.8	4.6	0.0467	111	3419
	28	6.6	3.76667	17.4956	82	1581
f3 < f2 < f1 < f4	27	6.2	4.26667	12.1756	81	3696
	26	5.5	4.83333	1.4689	85	1293
	28	6.6	3.76667	17.4956	82	11
f3 < f2 < f4 < f1	13	2.3	3.76667	0.1089	125	5000
f3 < f1 < f2 < f4	26	5.5	4.83333	1.4689	85	5000
f3 < f1 < f4 < f2	13	2.3	3.76667	0.1089	125	3855
	1	0.9	3.03333	0.3889	131	1078
	24	4.8	4.6	0.0467	111	67
f3 < f4 < f1 < f2	28	6.6	3.76667	17.4956	82	4744
	24	4.8	4.6	0.0467	111	256
f3 < f4 < f2 < f1	24	4.8	4.6	0.0467	111	5000
f4 < f2 < f3 < f1	28	6.6	3.76667	17.4956	82	5000
f4 < f2 < f1 < f3	26	5.5	4.83333	1.4689	85	4706
	27	6.2	4.26667	12.1756	81	294

f4 < f3 < f2 < f1	27	6.2	4.26667	12.1756	81	4561
	26	5.5	4.83333	1.4689	85	439
f4 < f3 < f1 < f2	28	6.6	3.76667	17.4956	82	5000
f4 < f1 < f3 < f2	28	6.6	3.76667	17.4956	82	5000
f4 < f1 < f2 < f3	26	5.5	4.83333	1.4689	85	5000

In Table 3.2.1 the first columns shows the importance order of the objective functions, where in the first rows function 1 is less important than function 2 and less important than function 3. Second column shows the solution that was selected as best, column 3 to 5 show the objectives and column six serves as the counter, how many times that solution was the best from all the other solutions when solved for f' . This table provides all the possible combinations in the event that the decision maker does not know the priorities of the problem. In this scenario depending on the importance of our objectives at least one solution is obtain and the largest number of solutions obtain is three. From the original Pareto-optimal set of 28 solutions the Non-uniform weight generator with pseudo-ranking scheme algorithm reduces the size to at least one solution and at most three therefore the method is reducing the Pareto-optimal set from 28 solution to a smaller size to be evaluated for the decision maker.

3.4 Case Study 3: Redundancy Allocation Problem (RAP)

The redundancy allocation problem for series parallel is another popular problem that was addressed by (Cao, D., *et al*, 2013) the aim of this problem is to maximize reliability f_r , maximize cost f_c , and minimize weight f_w . The problem is mathematically formulated as follows:

$$\begin{aligned} & \max f_r(X) \\ & \min (f_c(X), f_w(X)) \end{aligned}$$

Subject to

$$1 \leq \sum_{j=1}^{m_i} x_{ij} \leq n_{max,i} \quad \forall i \in S$$

$$X = \{x_{ij} | \forall i \in S, j = 1, \dots, m_i\}$$

$$x_{ij} \in \{0, 1, 2, \dots, n_{max,i}\}$$

Where,

$$f_r(X) = \left[\prod_{i=1}^s \left(1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right]$$

$$f_c(X) = \left[\sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} \right]$$

and

$$f_w(X) = \left[\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \right]$$

Where, r_{ij} denotes the reliability, c_{ij} denotes the cost and w_{ij} denotes the weights. $s = |S|$ is the number of subsystem for the set of subsystems S , x_{ij} is the decision variable with j th type components in subsystem i , m_i stands for the availability of components for subsystem i and $n_{max,i}$ stands for the maximum number of components in parallel used in subsystem i .

Similarly, as in case study 1 and 2, for the post-Pareto analysis stage assumptions where made. The decision maker has no knowledge about the priorities of the problem: he does not know how to generate the weights, nor the specific weight values. A set of $k=5000$ weights will be generated (this set of weights is determined by the decision maker he can decide of k amount of weights to test the Pareto-optimal set with) the solution where the counter is greater than zero will be chosen as the Pareto-optimal solution.

Results:

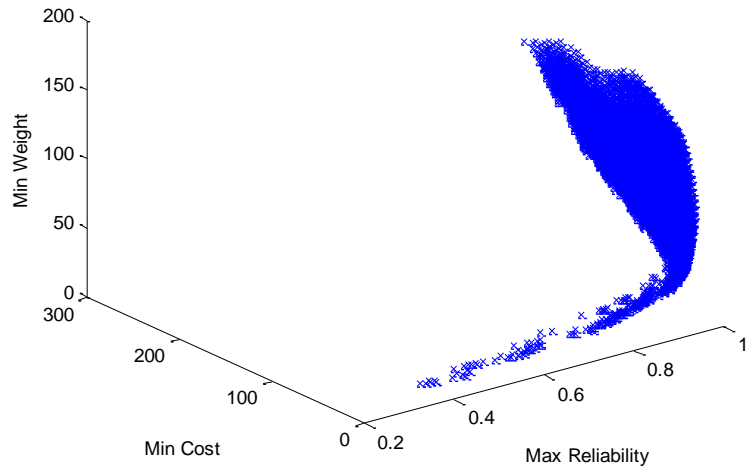


Figure 3.3.1: RAP 3-Dimensional view of the objectives.

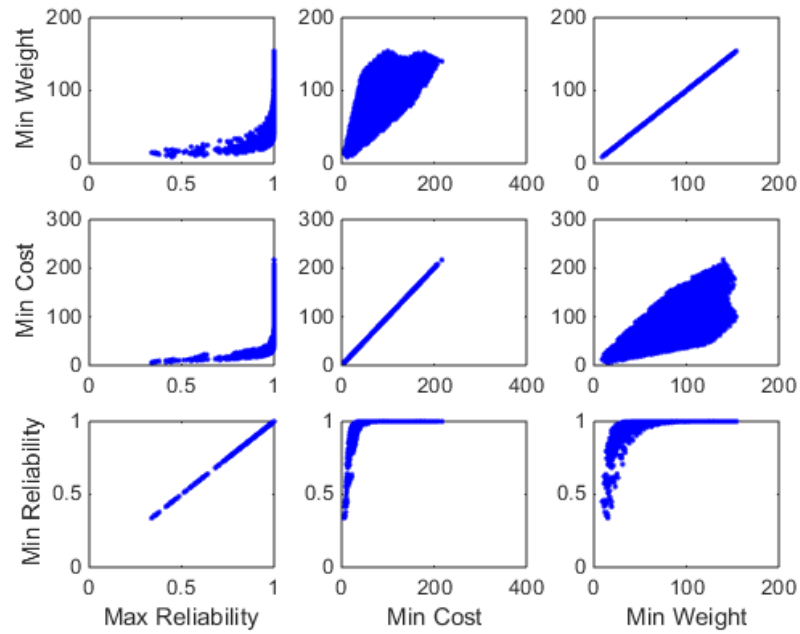


Figure 3.3.2: RAP Bi-Dimensional view of the objectives.

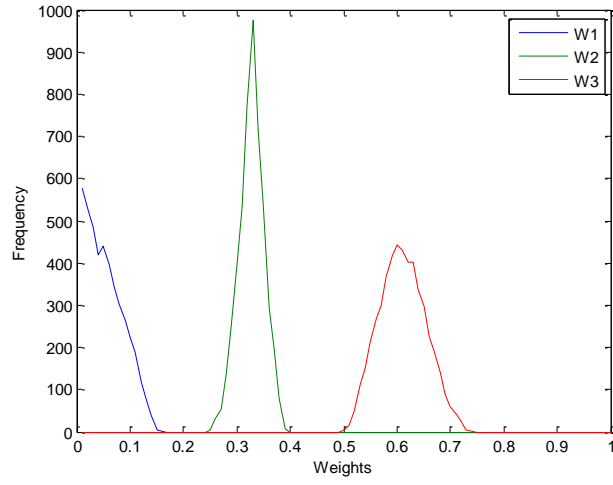


Figure 3.3.3: Weights Distribution Graph.

Table 3.3.1: RAP Pareto-optimal solutions

Importance order	Solution #	f1	f2	f3	Counter
f1 < f2 < f3	30	0.44856	12	9	3323
	92	0.746404	18	14	1383
	252	0.795617	26	13	294
f1 < f3 < f2	92	0.746404	18	14	1550
	30	0.44856	12	9	1444
	1	0.33768	6	15	1387
	16	0.41741	10	11	278
	47	0.721702	14	18	152
	66	0.735065	16	16	114
	48	0.583128	14	12	68
	123	0.798078	20	17	7
f2 < f1 < f3	560	0.937227	36	19	4898
	730	0.952373	40	21	102
f2 < f3 < f1	820	0.972991	42	24	2093
	1684	0.991637	58	30	1348
	1349	0.980944	52	26	1160
	1896	0.996053	62	32	399
f3 < f1 < f2	135	0.954272	21	53	1910
	182	0.969693	23	61	857
	203	0.976293	24	65	639
	309	0.968277	28	36	565
	154	0.958621	22	52	525
	118	0.947507	20	54	323

	218	0.955437	24	42	78
	202	0.974112	24	60	54
	156	0.962819	22	62	44
	196	0.951103	23	43	5
$f3 < f2 < f1$	328	0.991644	29	81	1053
	440	0.985617	32	47	808
	275	0.986222	27	68	779
	376	0.980015	30	44	598
	297	0.987558	28	67	502
	276	0.987247	27	73	468
	372	0.978237	30	42	177
	408	0.984283	31	48	94
	321	0.970433	28	38	89
	363	0.970038	30	34	86
	492	0.990299	34	51	68
	353	0.991956	30	75	60
	381	0.994254	31	85	51
	368	0.986655	30	57	41
	420	0.971532	32	32	41
	373	0.972198	30	36	34
	342	0.985319	29	58	25
	309	0.968277	28	36	19
	486	0.972799	34	30	4
	433	0.973695	32	34	3

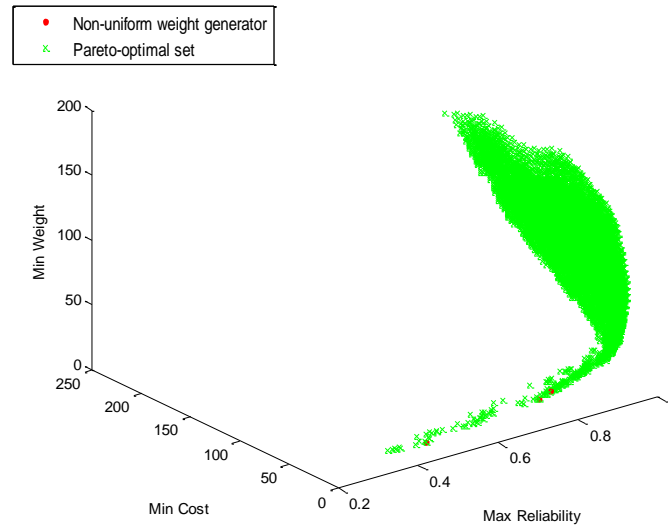


Figure 3.3.4: RAP 3-Dimensional view of reduced Pareto-optimal set

In Figure 3.3.1, the Pareto-optimal of the Redundancy Allocation Problem consisting of 6,112 solutions can be observed. In figure 3.3.2 a two dimensional representation of the objectives comparison can be observed. Figure 3.3.3 represents the set of 5,000 weights that were generated by the algorithm. Table 3.3.1 represents the Pareto-optimal solutions of all the possible combinations in regards to the importance order. Where, in the first row function 1 is less important than function 2 and less important than function 3. Second column shows the solution that was selected as best, column 3 to 5 show the objectives and column six serves as the counter, how many times that solution was the best from all the other solutions when solved for f' . The original Pareto-optimal set had a total of 6,112 solutions and from the table we can see that depending on the importance order the algorithm reduces the Pareto-optimal set size to at least two solution and at most twenty solutions. Furthermore, the table provided can aid to identify the decision maker what objectives he can consider more important compare to the rest, in Figure 3.3.4 the reduced Pareto-optimal set obtain in row one of our table can be observed by the red marks from the total Pareto-optimal set represented by the green marks.

Chapter 4: Post-Pareto Optimality using Game Theory

Game Theory can be defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers (Myerson, R. B., 2013). Game Theory provides a mathematical foundation for analyzing the decisions of two or more players; these decisions can result in a positive or negative outcome for each player. The term “Game” can be misleading; in Game Theory, it simply means an interactive situation where the players share rules and consequences.

In Game Theory, a game can be defined as a model and interaction between two or more players subject to constraints on actions that players can take based on their interests. Game Theory allows the players in a game to be animals, humans, government entities, university programs to name a few making it a cross discipline concept. Generally, there are two types of games, cooperative and non-cooperative games. Cooperative games are those where the players are able to make enforceable contracts, and non-cooperative the players are unable to make enforceable contracts.

The formal application of Game Theory for different problems requires some key elements: the players or the participants who have a set of significant strategies, their preferences, the strategic moves or action they are allowed to make, and the outcome or utility of each decision usually expressed as a measure, profit, quantity or desirability rank. The game varies from model to model and depending on the game, some other requirements and assumptions may be necessary. Game Theory models include Strategic games and Extensive form games explained in the following sections.

4.1 Literature Review

In order to understand how a game works, we need to identify the type of model we are trying to solve, then we apply an appropriate mathematical model and define the players, the possible actions each player can take, and the outcomes. The model is then computed to find the best decisions the players can take. The following works provide interpretations of how Game Theory can be applied to different scenarios, also modern changes in the concept of Game Theory more specific, different ways of obtaining the steady state of a game.

Since its original conception in the early 20th century, it has been a popular method of modeling both competitive and cooperative games. (Moorthy, K., 1985) explained the applications of Game Theory in market competition as a way of predicting competitor's actions and move to benefit from them through the understanding of equilibriums. Their work, however, proves more useful in understanding how different real-world concepts can be modeled through Game Theory rather than in providing concrete examples. (Srivastava, V. et al, 2005) proved the versatility of Game Theory as a modeling method by using it to analyze Ad Hoc networks, showing an improvement over previous techniques by having each node in the network as a player capable of making decisions to improve itself. However, they found difficulties in choosing the right utility functions and reaching the required model complexity. (Rabin, M., 1993) attempted applying fairness and Game Theory into economics. His premise was that people not always tend to work towards their own individual benefit, but that they can, at some time also work to help or harm others, as a reflection of how others behave towards them. Their idea of a kindness function can be taken as a tentative model for some measures of fairness.

There has been focus on computing the steady state of a game, reaching the Nash equilibrium of such. For instance, (Sefrioui, M., & Perlaux, J., 2000) identified Nash genetic

algorithms as a fast alternative to multiple objective optimization and suggested that Nash equilibrium is on the Pareto frontier. The Nash Domination evolutionary algorithm was introduced in (Koh, A., 2012), and proposed that the Nash Dominance concept measures the proximity of a strategy to Nash equilibrium by counting the number of players that can profitably deviate. Others have developed a hybrid algorithm to optimize integrated process and scheduling (IPPS) problem based on Nash equilibrium approach (Li, X. et al., 2012). (Dumitrescu, D., et al., 2011) introduces Relational Evolutionary Equilibria Detection (REED) incorporating Berge Pareto equilibrium. (Pavlidis, N. G., 2005) investigated the effectiveness of computational intelligence techniques to compute Nash equilibria. Sequential game theoretic approach for multiple objective clustering, utilizing backward induction to calculate Nash equilibrium for each game was introduced by (Heloulou, I. et al., 2017)

Furthermore, the application of Game Theory to many different scenarios can best be observed in the following works where, (Madani, K., 2010) reviews the applicability of Game Theory to water resource management and conflict resolution, and discussed how the structure of the games might be changed by third parties to promote Pareto-optimal resolution. (Gao, J., & You, F., 2017) proposed Stackelberg based game model to optimally design a non-cooperative gas supply chain considering economic and life cycle GHG emissions. (Ahmad, I., et al, 2008) showed yet another application of Game Theory, using it to schedule tasks in multi-core processors to perform a multiple objective optimization of performance and energy use. Game Theory has also seen applications in politics, such as did (Snidal, D., 1985), economics (McMillan, J., 2013) and communications (Saad, W., et al., 2009).

4.2 Game Theory Concepts

In order to model a problem as a game, there are many key elements that need to be defined, and depending on the model other requirements or assumptions may be necessary. This section provides general terminology of the key elements a game requires to have in order to work properly.

4.2.1 Steady State / Nash Equilibrium

In traditional Game Theory, Nash equilibrium is obtained by definition and states that if there is a strategy with the property that no player can increase their payoff by changing of strategy while the other players maintain their strategies unchanged then it is said that the strategy has reached Nash equilibrium or a steady state.

4.2.2 Player Definition

There are two types of players in Game Theory. The first definition for a player is, a participant who has a nontrivial set of strategies, usually more than one and the participant chooses a strategy based on the payoff.

4.2.3 Strategy Definition

A strategy can be defined as, set of moves or actions a player may follow in a given game. An action may be decided by chance, and in some cases, no probability is involved.

Pure Strategy

A pure strategy is defined by a specific move or course of action where there is no probability involved. For example, a right-footed soccer player always shoots to the left.

Mixed Strategy

A strategy consisting of different possible moves and preferences (weights) on how frequently each move is to be played mixed strategy allows the player to choose action probabilistically. For example, a goalie can flip a coin to decide if he will dive left or right.

4.2.4 Utility Definition

The utilities are used to define the payoff of each player depending on the outcome of the game. Payoffs in most cases represent a quantity, profit, a measure or may simply rank the desirability of the outcome

4.3 Strategic Game Models

A strategic game is an interactive decision making model where each decision maker chooses his strategy, each strategy is made simultaneously. Some examples to understand the general idea behind simultaneous games include:

Rock, Paper, Scissor, a game played by two players, where the players select simultaneously either, rock, paper or scissors. For this game paper>rock, rock>scissors, and scissors>paper. The outcome of the game is that the player wins, loses, or draws (Fisher, L., 2008).

Battle of sexes, in these situation two players: girl and boy tried to decide a place to spend the evening together. The boy chooses place a and the girl place b , but both would like to be together. The place they attend will favor one or the other (Nawaz, A., & Toor, A. H., 2004).

Prisoner's dilemma, players (prisoners) can choose from two strategies: to cooperate or stay quiet. If both players a and b confess they both get 4 years in prison. If player a confesses

and player b remains silent, a will be set free and b will be imprisoned for 7 years and vice versa. If both players remain silent, both will serve one year in prison.

4.3.1 Definition

The mathematical representation of general strategic games consists of the following (Osborne, M. J., & Rubinstein, A):

- a finite set N (set of players)
- for each player $i \in N$ a nonempty set A_i (the set of actions available to player i).
- For each player $i \in N$ a preference relation \succsim_i on $A = \prod_{j \in N} A_j$ (the preference relation of player i).

If the set of actions A_i of every player i is finite, then the game is *finite*. In some situations, preferences may be represented by a payoff function:

$$p_i: A \rightarrow R$$

The function associates payoff with action profiles, the payoff represents the motivation (i.e. profit, quantity, utility).

4.4 Extensive Form Game Models

An extensive game depicts the order, in which the players make a move, and the available information each player has at each decision point. These models are usually described with a tree on how the game is played. In extensive form games, the players do not act simultaneously but rather sequentially. Examples of extensive games include:

Chess Game, a game played by two players. In this game, the player with the white pieces moves first followed by the player with the black pieces. Following each strategy or move each player decision is based on what the other decides to move. The outcome of the game is that either player wins, loses, or draws.

Tic-Tact-Toe, a two-player game where the players move sequentially and each move is decided based on the other player's actions. The outcome of the game is that either player wins, loses or draws.

4.4.1 Definition

Extensive form of a game consists of the following components (Osborne, M. J., & Rubinstein, A., 1994):

- A finite N (set of players $i = 1, 2, \dots, n$)
- A set H of consequences (finite or infinite)
- Set of actions available after nonterminal history h denoted by $A(h) = \{a: (h, a) \in H\}$ and set of terminal histories Z .
- Function P that assigns to each nonterminal history a member of $N \cup \{c\}$. Where P is the player function, $P(h)$ being the player who takes an action after the history h . If $P(h) = c$ then the action taken after the history h is determined.
- Function f_c that associates with every history h for which $P(h) = c$ a probability measure on $A(h)$, where each such probability is independent of every other measure.
- For each player $i \in N$ a partition I_i of $\{h \in H: P(h) = i\}$ with the property that $A(h) = A(h')$ whenever h and h' are in the same member of partition. For $I_i \in I_i$, $A(I_i)$ denotes the set $A(h)$ and $P(I_i)$ the player $P(h)$. I_i is the information partition of player i ; a set $I_i \in I_i$ is an information set of player i

- Utility function $u_i: Z \rightarrow R$ for each player $i \in N$

4.5 Evolutionary Game Theory

Evolutionary Game Theory originated from the idea that frequency introduces a strategic aspect to evolution. Evolution in this context is often described as cultural evolution, where this refers to the changes in beliefs over time. In this theory, the agent adapts the chosen strategy based on its payoff, or its fitness, by doing such, equilibrium in terms of static and dynamic behavior can be analyzed (Han, Z., 2012). In the past years, Evolutionary Game Theory has become very popular by, sociologists, economists, and social scientists in general because it provides a missing element to the traditional game theory as an explicit dynamic theory.

Evolutionary Game Theory studies strategic behavior with respect to evolutionary forces in terms of a game played many times in large populations by agents with bounded rationality. In Evolutionary Game Theory, the rational is more appropriate for modeling social systems. These agents are randomly chosen from a large number population and have no information about the game.

One key concept in evolutionary game theory is the Evolutionary Stable Strategy (ESS). This concept provides a static conceptual analysis for finding the evolutionary stability. The second approach of evolutionary game theory constructs an explicit model of the process and studies the evolutionary dynamic of the frequency of strategies changed in the population by the process. This second approach does not intend to find stability it rather defines a model of the population dynamics.

Furthermore, Evolutionary Game Theory has identified many aspects of human behaviors. In (Gintis, H., 2007) demonstrated that an endowment effect can be modeled for private property where, private property is seeing in nature as species territory. The prisoner's

dilemma interactions demonstrated that spatial structure benefited cooperation (Hauert, C., 2006). Other models have been created for understanding the culture increase over time, for instance in (Enquist, M., et al., 2008) developed a model on how human creativity and cultural transmission can exponentially increase culture over time. The implementation of environmental strategies by core enterprises and governments was modeled in (ZHU, Q. H., & DOU, Y. J., 2007) where the model served to study strategy from which governments and core enterprises can benefit from implementing a green supply chain management. There are many more examples from many different disciplines where Evolutionary Game Theory has been implemented.

4.6 Nash-Dominant Pareto Set Reduction Algorithm

Nash dominance is a relatively new concept, developed by (Lung, R. I., & Dumitrescu, D., 2008). It is based on the idea of Nash equilibrium, a concept in game theory in which none of the players would like to switch their strategy. In Nash equilibrium, a specific strategy can be denoted by $s \in S$, where S represents the set of all possible strategies that can be taken. It can be said that a strategy s^* Nash-dominates a strategy s if the number of players who would benefit from switching from strategy s to strategy s^* is greater than the number of players who would benefit from switching from a s^* to s :

Lung introduces the operator k to denote the number of players n benefitting from s to s^* :

$$0 \leq k(s^*, s) \leq n$$

This operator will yield a number of players benefitting from the switch between 0 (no players will benefit) and n (all players will benefit). From this concept, we can say that a strategy x dominates a strategy y in the Nash sense, and $x \succ y$ if the inequality:

$$k(x, y) > k(y, x)$$

holds. Thus, a set of Non-Nash-Dominated solutions can be formed.

In order to apply this methodology into a Post-Pareto analysis, we have defined that each of the objectives in a non-dominated solution set is a player in a game, and each of all the solutions in the set are to be considered strategies. The players will be designated by $\rho \in P$, while the strategies are to be designated by $s \in S$. The values of the objective functions for each of the solutions will be considered as the utility values for each player i under each strategy j . As such, the players aim to choose a strategy (or number of strategies) which will benefit the most of them individually, but which will decide the actions of them all as a team.

In order to find the strategy that will benefit the most players, our first step is to do pairwise comparisons between all strategies, determining k for all of them. As before, a strategy x dominates a strategy y if $k(x, y) > k(y, x)$. After obtaining all k , a dominance check is performed, marking the strategies that are dominating and those that are dominated. For the first option of Pareto set pruning, all dominated strategies are ignored, and only the non-dominated solutions are considered for the Nash non-dominant set. However, this method is not very effective for a small number of ρ , since one strategy will easily dominate all others, yielding a strategy that may be too undesirable for the other players. This method works better with a larger number of players, where it is harder for a single strategy to have better results. The basic algorithm of this method is given below:

1. Determine k for each pair of strategies s_i and s_j :

For $i=1:S$

For $j=1:S$

$k(i, j)=0$;

For each player $\rho=1:P$

If $u(i, \rho) > u(j, \rho)$;

$k(i, j)=k(i, j)+1$;

```

        end if
    end for
end for
2. For each pair of strategies, determine which strategies are dominated:
For  $i=1: S$ 
     $\Delta_0(i)=0$ 
    For  $j=1: S$ 
        If  $k(i,j) < k(j,i)$ 
             $\Delta_0(i)=1$ 
        End if
    End for
End for
3. Save the solutions that were not marked as dominated (Those for which  $\Delta_0=0$ )
END

```

The second alternative involves a requirement factor δ applied over the dominant set. That is, the set of solutions that dominates another solution. This set is determined also through the k operator, and the number of times each strategy dominates another is recorded as Δ . The strategies that do not meet the required number of dominances are deleted from the set, producing another pruned set of solutions. The criterion for determining which strategies are kept and which are eliminated through this method is as follows where D represents the decision on whether a solution is kept (1) or eliminated (0):

$$D = \begin{cases} 1, & \text{if } \Delta \geq \delta \\ 0 & \text{if otherwise} \end{cases}$$

The capability of arbitrarily choosing δ can be very useful in varying the size of the pruned set. δ can generally be chosen as $\max_{\Delta} \cdot (1-\epsilon)$, where ϵ is determined by the decision maker, the data to be kept according to the highest dominance count. It guarantees the presence of strategies or solutions that, according to their Nash-dominance, can be considered as better,

while allowing the inclusion of less dominant solutions in order to give the decision maker more options, which can just as easily be removed. The basic algorithm for this method is given below:

1. Determine δ .
2. Obtain k for all pairs of strategies as described previously.
3. . For each pair of strategies, determine how many times each strategy dominates others:

For $i=1: S$

$\Delta(i)=0$

For $j=1: S$

If $k(i,j) < k(j,i)$

$\Delta(i)= \Delta(i)+1$

End if

End for

End for

4. Make the decision D for every strategy $s \in S$

5. Save the solutions for which $D=1$

END

The second method is based partly in a method proposed by (Li, X., et al., 2012). They, however, propose a normalization for the utility values of each player based on the best result that player can obtain, which they denote as DOB_{ji} :

$$DOB_{ji} = \frac{CurrentObjective_{ji} - BestObjective_i}{BestObjective_i}$$

Where, i represents the players and j represents the strategies. They then obtain for each solution the Nash Equilibrium Criterion $NashE_j$:

$$NashE_j = \sum_{i=1}^p DOBJ_{ji}$$

They compare all the Nash Equilibrium criteria to the best criterion, and then against ε , the “Nash equilibrium solution factor”, which simply put is an allowance factor for the difference between each criterion to the best. The solutions that fall within their solution factor are recorded within the pruned Pareto set.

Chapter 5: Post-Pareto Optimality using Game Theory

This chapter will provide three different case studies to identify the performance of the method developed in section 4.6.

5.1 Case Study 1: DTLZ1 Test Problem

This is the simplest test problem with multiple objectives from the DTLZ set of test problems. The experiment was run on an acer desktop, with Windows 7 Home Premium 64-bit operating at 3.2 GHz and 8GB of RAM. The program was coded in MATLAB® R2014b. This scenario has three objectives to be minimized, represented by functions f_1 , f_2 , and f_3 . In order to introduce Game Theory to this problem, the functions will be treated as the players. The strategies will be represented by the solutions, in this case there are a total of 2,500 solutions. The utility will be represented by how many times the strategy had the best outcome which is most likely the preference of the decision maker, where a good strategy is denoted by 1 and a bad strategy by 0. The data set was obtained from Coello CAC., (n.d.), and mathematically it can be expressed as follows:

$$\text{Minimize } F = (f_1(X), f_2(X), f_3(X))$$

Subject to

$$n = 12$$

$$0 \leq x_i \leq 1$$

$$i = 1, \dots, 12$$

Where,

$$f_1(X) = \frac{1}{2}x_1x_2(1 + g(X))$$

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

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

and

$$g(X) = 100 \left[10 + \sum_{i=3}^n (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right]$$

Results:

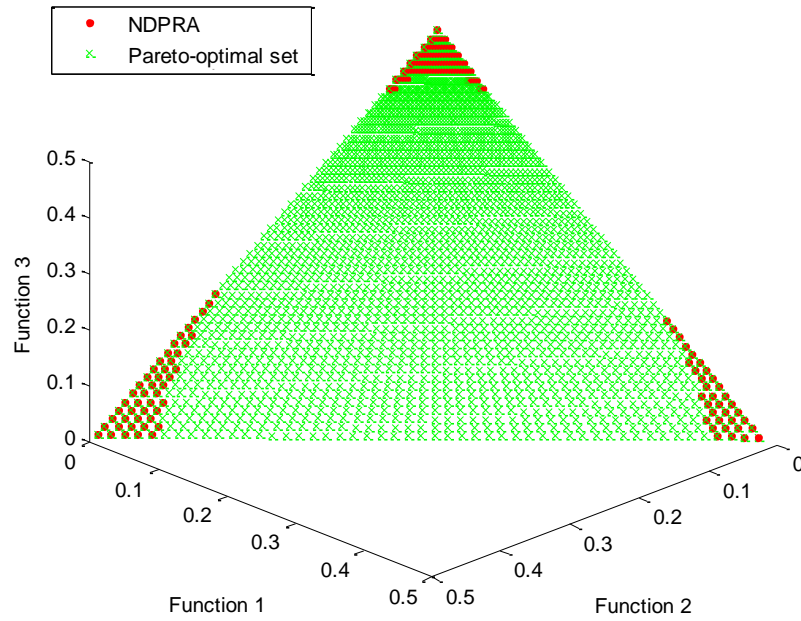


Figure 5.1.1: DTLZ1 3-Dimensional graph of each objective.

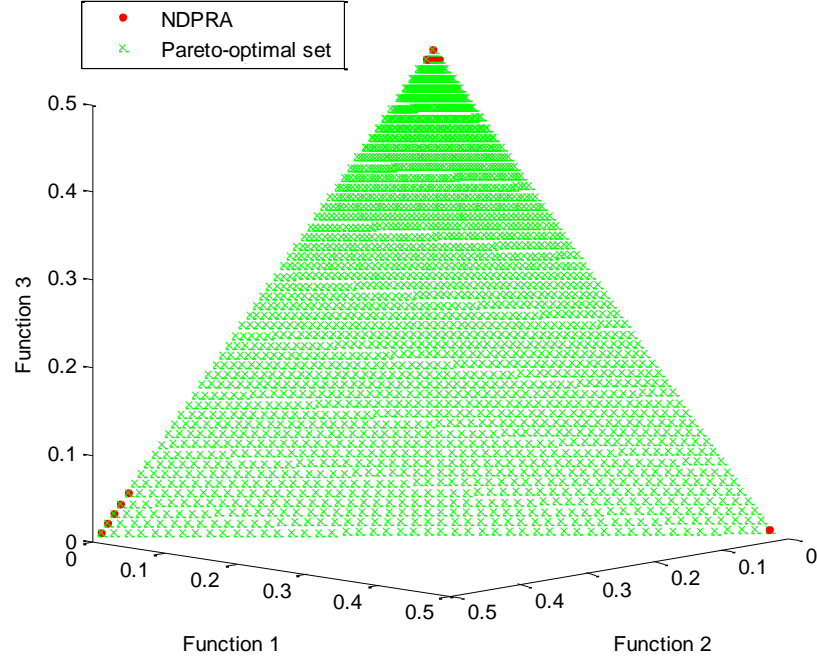


Figure 5.1.2: DTLZ1 3-Dimensional graph of each objective.

In Figure 5.1.1, a visual representation of the DTLZ1 test problem can be observed. The Pareto-optimal set represented by the blue green points consists of 2,500 solutions, for this scenario $\epsilon=0.3$. The introduction of the Nash-Dominant Pareto set reduction algorithm produced a pruned Nash-Dominant Pareto set of 403 solutions depicted by the red marks. In Figure 5.1.2 the scenario where $\epsilon=0.1$ can be observed and the pruned Pareto-optimal represented by the red mark consist of 106 solutions. An immediate conclusion that can be observed is that based on what the decision maker selects as ϵ , he / she can control the size of the Pareto-optimal set to evaluate. Selecting the size of the Pareto-optimal set to be evaluated is up to the decision maker. After reducing the size of the Pareto-optimal set in terms of Nash-Dominance the algorithm is reaching the solution where most of the players chose the best strategy. Alternatively, it can be observed that this method is selecting solutions at the extreme points for each objective.

5.3 Case Study 2: Printed Wiring Board (PWB) Problem

The next example is the scheduling of a Printed Wiring Board (PWB) manufacturing line introduced in section 3.3. The experiment was run on an acer desktop, with Windows 7 Home Premium 64-bit operating at 3.2 GHz and 8GB of RAM. The program was coded in MATLAB® R2014b the objective of this problem is to minimize overtime f_1 , minimize average finish time f_2 , minimize the variance of the finish time f_3 and minimize the total cost f_4 . To represent this model in to a game the players will be f_1, f_2, f_3 , and f_4 respectively. The strategies will be represented by the 28 solutions in the Pareto-optimal set. The utility of this game will be represented by the strategies where a good strategy is denoted by 1 and a bad strategy is denoted by 0. Based on the NDPRA the strategy that will prefer is based on how many times the strategy had the best outcome.

The Printed Wiring Board problem is mathematically expressed as follows:

$$f_1 = \min \sum_{i=1}^m O_i, \quad f_2 = \min \mu_c, \quad f_3 = \min \sum_{i=1}^m \frac{(C_i - \mu_c)^2}{m}, \quad \text{and} \quad f_4 = \min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

Subject to

$$\sum_{i=1}^m x_{ij} = 1$$

$$O_i = \max \left(\sum_{j=1}^n p_{ij} x_{ij} - T, O \right)$$

$$C_i = \sum_{j=1}^n p_{ij} x_{ij}$$

$$\mu_c = \frac{\sum_{i=1}^m C_i}{m}$$

Where,

$$x_{ij} = \begin{cases} 1, & \text{if lot } j \text{ is assigned to machine } i \\ 0, & \text{otherwise} \end{cases}$$

Where, O_i is the overtime of machine i , n is the number of lots to schedule, m is the number of parallel machines, p_{ij} is the processing time of lot j on machine i , c_{ij} is the cost of processing a lot j on machine i , and T is the lot release interval times.

Results:

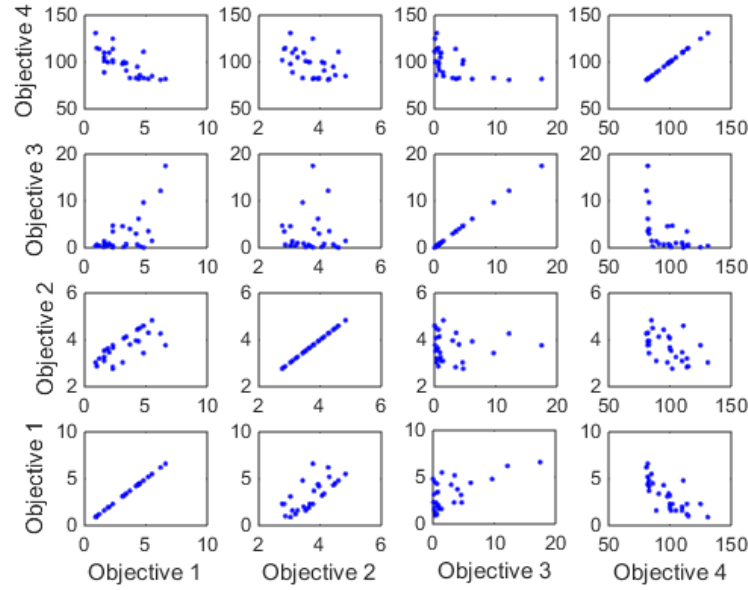


Figure 5.3.1: PWB Bi-Dimensional view of the objectives.

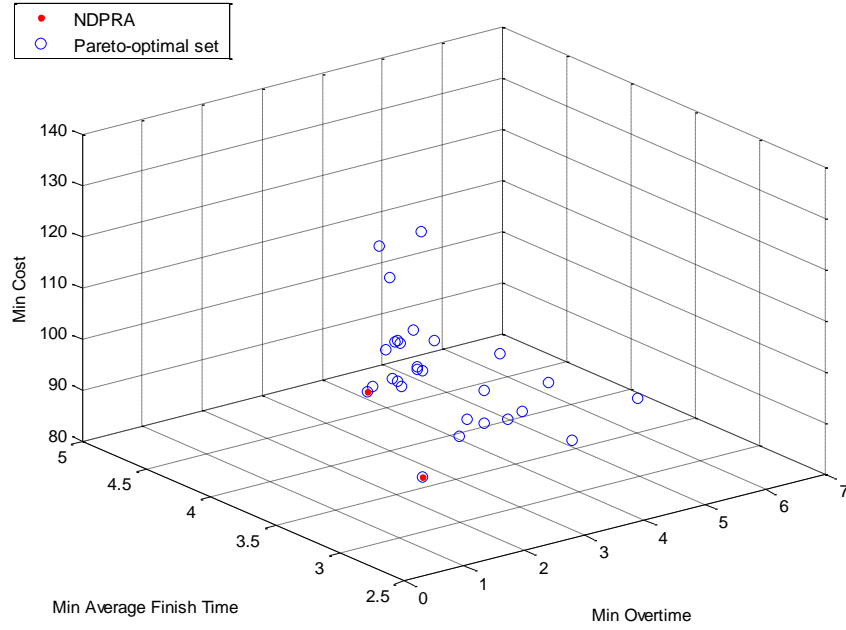


Figure 5.3.2: PWB 3-Dimensional graph.

Table 5.3.1: PWB Pareto-optimal set

Solution #	Min Overtime	Min Avg. Finish Time	Min Variance Avg. Finish Time	Min Cost
5	1.6	3.1	1.4067	89
7	1.6	3.53333	0.1622	101

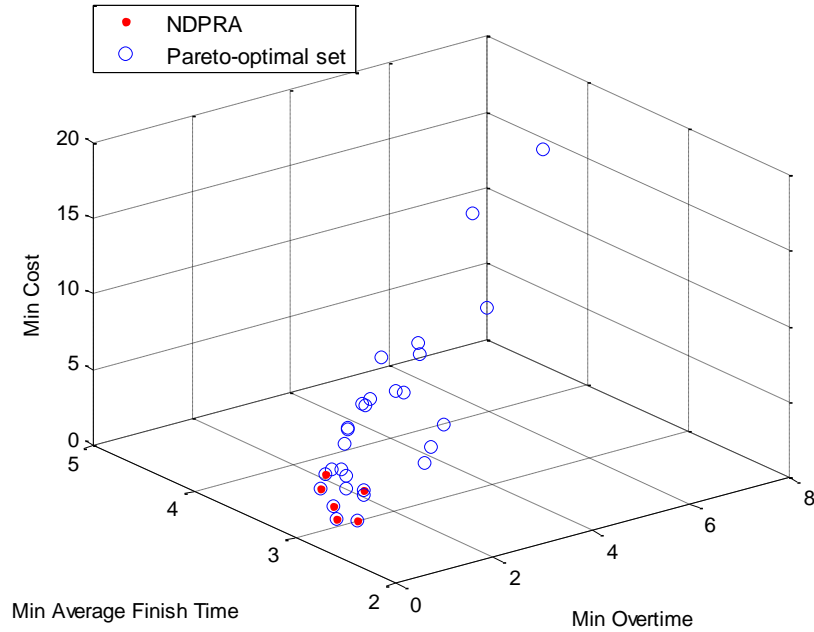


Figure 5.3.3: PWB 3-Dimensional graph.

Table 5.3.2: PWB Pareto-optimal set

Solution #	Min Overtime	Min Avg. Finish Time	Min Variance Avg. Finish Time	Min Cost
1	0.9	3.03333	0.3889	131
2	1	2.86667	0.6489	115
3	1.2	3.2	0.4267	114
5	1.6	3.1	1.4067	89
7	1.6	3.53333	0.1622	101
8	1.9	3.63333	0.5756	100

In Figure 5.3.1, a representation of the Processing Wiring Board can be visualized by the bi-dimensional graph taking into account the four objectives. Figure 5.3.2 and Figure 5.3.3 shows the three-dimensional space taking into account Min Overtime, Min Average Finish Time, and Min Cost. Min average variance time is taken in to account for the model but not for the three dimensional visual representation. For Figure 5.3.2, $\epsilon=0.1$ and it can be observed that the

NDPRA approach produced two solution from the 28 solution. In Figure 5.3.3, $\epsilon=0.3$ and the NDPRA approach obtained a Pareto-optimal set of 6 solutions out of the 28. Table 5.3.1 and Table 5.3.2 show the pruned solutions obtained by the NDPRA respectively. The Pareto-optimal set represented by the blue color circles consists of 28 solutions. The introduction of the Nash-Dominant Pareto set reduction algorithm reduced the size of the Pareto-optimal set represented by the red points in the three dimensional graphs. The main goal is to minimize all the objectives, and the results show that the NDPRA effectively reduces the Pareto-optimal set, it is important to note that deciding the reduced size of the Pareto-optimal set can be determine by ϵ .

5.3 Case Study 3: Redundancy Allocation Problem (RAP)

The next example is the Redundancy Allocation problem used in section 3.3. The experiment was run on an acer desktop, with Windows 7 Home Premium 64-bit operating at 3.2 GHz and 8GB of RAM. The program was coded in MATLAB® R2014b. The objectives of this problem are to maximize reliability f_r , maximize cost f_c , and minimize weight f_w . In order to represent this problem as a game, player one will be represented by f_r , player two will be represented by f_c , and player three will be represented by f_w respectively. The strategies will be represented by the 6112 solutions in the Pareto-optimal set. The utility of this game will be represented by the strategies; good strategy is represented by 1 and a bad strategy by 0, where in each strategy the players will simultaneously decide if strategy s_i is better than strategy s_j based on the NDPRA introduced in section 4.6. The strategies will be chosen based on the utility, in this case the preference of how many times each strategy was the best.

The RAP is formulated as follows (Cao, D., *et al*, 2013):

$$\max f_r(X), \min (f_c(X), f_w(X))$$

Subject to:

$$1 \leq \sum_{j=1}^{m_i} x_{ij} \leq n_{max,i} \quad \forall i \in S$$

$$X = \{x_{ij} | \forall i \in S, j = 1, \dots, m_i\}$$

$$x_{ij} \in \{0, 1, 2, \dots, n_{max,i}\}$$

Where,

$$f_r(X) = \left[\prod_{i=1}^s \left(1 - \prod_{j=1}^{m_i} (1 - r_{ij})^{x_{ij}} \right) \right], \quad f_c(X) = \left[\sum_{i=1}^s \sum_{j=1}^{m_i} c_{ij} x_{ij} \right] \text{ and } f_w(X) = \left[\sum_{i=1}^s \sum_{j=1}^{m_i} w_{ij} x_{ij} \right]$$

Where, r_{ij} denotes the reliability, c_{ij} denotes the cost and w_{ij} denotes the weights.

$s = |S|$ is the number of subsystem for the set of subsystems S , x_{ij} is the decision variable with j th type components in subsystem i , m_i stands for the availability of components for subsystem i and $n_{max,i}$ stands for the maximum number of components in parallel used in subsystem i .

Results:

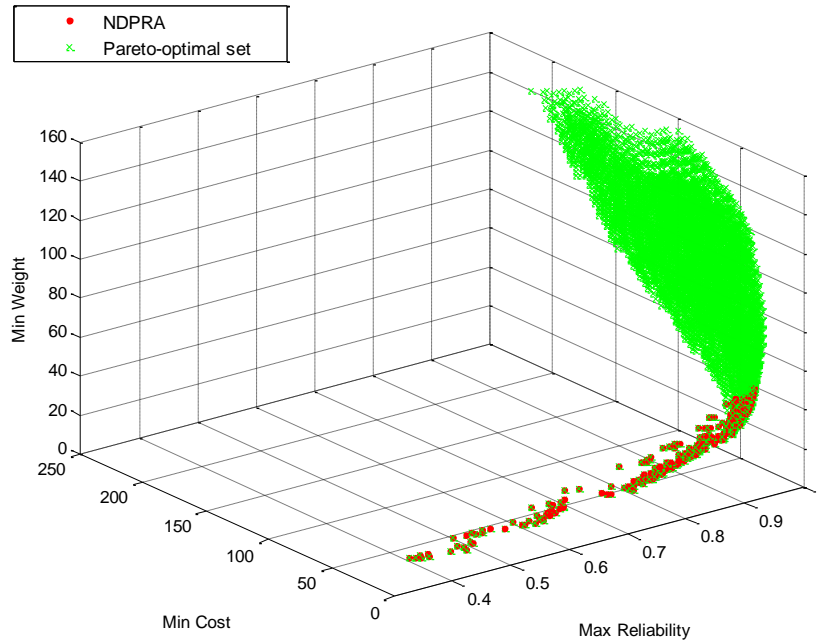


Figure 5.3.1: RAP 3-Dimensional graph of each objective.

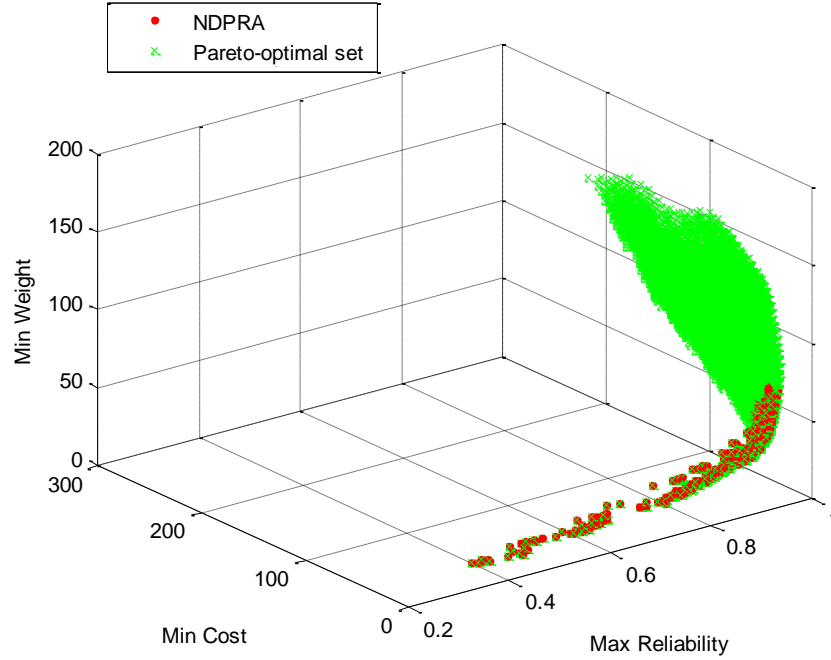


Figure 5.3.2: RAP 3-Dimensional graph of each objective.

In Figure 5.3.1, a representation of the Redundancy Allocation Problem can be observed. The Pareto-optimal set represented by the green color points and consists of 6112 solutions. The introduction of the Nash-Dominant Pareto set reduction algorithm produced a pruned Pareto-optimal set to 410 solutions using $\epsilon=0.1$. In Figure 5.3.2, $\epsilon=0.2$ and the Nash-Dominant Pareto set reduction algorithm pruned the Pareto-optimal set to 832 solutions. The redundancy allocation problem main goal is to maximize reliability and to minimize cost, and weight. The decision maker can choose ϵ to identify the set of solutions to evaluate and it can be observed that the algorithm discarded solutions with relatively high costs and high weights and now the decision maker has a reduce set of the Pareto-optimal set to evaluate.

Chapter 6: Conclusions

The post-Pareto analysis stage can cause the decision makers a hard time selecting one solution from the Pareto-optimal set. In many problems, the Pareto-optimal set can be in the thousands and for that reason, two approaches to reduce the size of the Pareto-optimal set were introduced.

In chapter three, the Non-uniform weight generator with pseudo-ranking scheme algorithm was introduced to further expand on the previous work of (Carrillo, V. and Taboada, H., 2012). In Carrillo, V. and Taboada, H. the Redundancy Allocation Problem was the only problem that the method's performance was tested with, and that specific problem consisted of a Pareto-optimal set of 75 solutions. The method introduces a boundary technique to reduce the size of the Pareto-optimal set by using thresholds values to reduce the size; by using this value the Pareto-optimal set was reduced. As the value of the threshold value increases, the Pareto-optimal solutions subset increases providing a pruned Pareto-optimal set of at least one solution.

The Non-uniform weight generator with pseudo ranking scheme algorithm introduced in chapter 3, was tested with three different problems, the method's selection of the pruned Pareto-optimal set is what determines the Pareto-optimal set (counter > 0). For the redundancy, Allocation Problem a set of 6112 Pareto-optimal solutions was used. The Pareto-optimal set was found for all the possible combinations in terms of importance of the objective functions to provide the decision maker with no knowledge about the priorities of the problem different alternatives. To test the performance of this method the DTLZ1 test problem consisting of a Pareto-optimal set of 2,500 solutions and Printed Wiring Board (PWB) Problem consisting of a Pareto-optimal set of 28 solutions were used as well, in all of the cases the algorithm reduced the size of the Pareto-optimal set.

Therefore, in Carrillo, V. and Taboada, H. the proposed method was tested with a Pareto-optimal set of 75 solutions, on the other hand the method introduced in this thesis uses a Pareto-optimal set of 6112 solutions it is hard to assess the improvement of the method due to differences in data. Although, some advantages and disadvantages can be noted in terms of improvement. In Carrillo, V. and Taboada, H. by using thresholds values, the advantage is that it can reduce the Pareto-optimal set to any size depending on what the decision maker wants. This can also be a disadvantage as it requires additional involvement and knowledge from the decision maker in selecting a threshold value, and not knowing what threshold value to select it can lead to no solutions. Moreover, the method proposed in this thesis does not have a boundary technique, but by the pseudo-ranking scheme provided in order to select the Pareto-optimal set it provides at least one solution and no extra involvement from the decision maker is required.

The second proposed method is a Game Theory based method. This method uses the concept of Nash-dominance that was derived from Nash-equilibrium to reach a steady state to a game. In the proposed method, ϵ was introduced to determine δ , the decision maker decides the size of the Pareto-optimal set subset to evaluate. The method's performance was tested by three different problems, the first problem is the DTLZ1 test problem, the second was Printed Wiring board Problem (PWB), and lastly the Redundancy Allocation Problem (RAP). Based on the results the method does reduce the Pareto-optimal set to a small subset, and it can be observed that by the decision maker selecting high or low ϵ the size of the Pareto-optimal set reduces accordingly, thus providing a less complicated task for selecting one solution.

In order to introduce the Nash-Dominant Pareto Set reduction algorithm in to other problem it is important for the problem to be model as a game. For any game there needs to be players and strategies and for this algorithm to work the strategies of the players need to be

known, that is all the possible solutions a player can decide. In this thesis, the case studies were model as a game where the objective functions represented the players, the strategies were represented by the Pareto-optimal set, and knowing these key elements the Nash-Dominant Pareto Set reduction algorithm reduced the Pareto-optimal set.

Furthermore, my main contribution to the industrial engineering community is to expand on the method proposed by Carrillo, V. and Taboada, H. and testing of the method with three different case studies with large Pareto-optimal sets. Also, by proposing a Game Theory based algorithm that allows the decision maker to reduce the size of the Pareto-optimal set to a more suitable size for evaluation.

Chapter 7: Future Research

In addition to the previously described method in section 4.6, an attempted to create an evolutionary algorithm incorporating fair division into the Nash equilibrium was performed. Fair division is a mathematical concept concerned with maximizing the satisfaction of each individual with the portion of a whole it has received. It has been largely studied with resource allocation, such as did (Thomson, W., 1983) and (Moulin, H., 2004). Its applications have also been explored in politics (Brams, S. J., 2008). It has also been used in conjunction with game theory in order to create better models or obtain more fair solutions, which can result in better equilibria (Tadenuma, K. and Thomson, W., 1995; Crawford V. P., 1977; Haake, C. J., et al, 2002).

The idea is that if, within all the possible non-dominated strategies, each player stands to gain, on average, the same then there is a fair division when obtaining a sub-set of the strategies. A genetic algorithm is applied into the dominant set of solutions described previously, where all objectives have been transformed into the same type (min or max) and the utilities have been normalized between 0 and 1 for each player.

The chromosome for this evolutionary algorithm is a binary vector with length S_D , the number of strategies in the dominant solution set, where 1 would represent the inclusion of the corresponding strategy into the pruned solution set and 0 its exclusion. This decision to include or exclude will also be represented by D . The algorithm is given as follows:

1. Generate random initial population
2. Evaluate Objective Function for each individual:
 - a. sum all normalized utility values for each player:

for $i=1:P$

for $j=1:S$

$$U_i = \sum_{j=1}^S D_j * u_j$$

End for

End for

b. obtain objective function

$$\min \text{OF} = \text{range}(U_j)$$

3. Rank solutions

4. Until the desired number of iterations has been run, perform steps 5 and 6, then go to 7:

5. Perform Crossover to generate a new population, accounting for elitism and mutation

6. Perform steps 2 and 3

7. Obtain optimal solution

This method attempts to generate a pruned Pareto set that is most fair to all players in terms of possible utility gain. However, as with all metaheuristic algorithms, it is possible to not obtain an ideal solution. Thus, future research is needed to further develop this hybrid algorithm.

References

- [Abraham, A., & Jain, L. (2005). Evolutionary multiobjective optimization. *Evolutionary Multiobjective Optimization*, 1-6.]
- [Ahmad, I., Ranka, S., & Khan, S. U. (2008, April). Using game theory for scheduling tasks on multi-core processors for simultaneous optimization of performance and energy. In *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on* (pp. 1-6). IEEE.]
- [Ahmadi, A., Kaymanesh, A., Siano, P., Janghorbani, M., Nezhad, A. E., & Sarno, D. (2015). Evaluating the effectiveness of normal boundary intersection method for short-term environmental/economic hydrothermal self-scheduling. *Electric Power Systems Research*, 123, 192-204.]
- [An, S., Yang, S., & Ren, Z. (2017). Incorporating Light Beam Search in a Vector Normal Boundary Intersection Method for Linear Antenna Array Optimization. *IEEE Transactions on Magnetics*, 53(6), 1-4.]
- [An, S., Li, Q., & Yang, S. (2016). An Improved Light Beam Search Method in Multiobjective Inverse Problem Optimizations. *IEEE Transactions on Magnetics*, 52(3), 1-4.]
- [Augusto, O. B., Bennis, F., & Caro, S. (2012). A new method for decision making in multi-objective optimization problems. *Pesquisa Operacional*, 32(2), 331-369.]
- [Brams, S. J. (2008). Mathematics and democracy: Designing better voting and fair-division procedures. *Mathematical and Computer Modelling*, 48(9), 1666-1670.]
- [Brito, T. G., Paiva, A. P., Ferreira, J. R., Gomes, J. H. F., & Balestrassi, P. P. (2014). A normal boundary intersection approach to multiresponse robust optimization of the surface roughness in end milling process with combined arrays. *Precision Engineering*, 38(3), 628-638.]
- [Cao, D., Murat, A., & Chinnam, R. B. (2013). Efficient exact optimization of multi-objective redundancy allocation problems in series-parallel systems. *Reliability Engineering & System Safety*, 111, 154-163.]
- [Cao, D., Murat, A., Chinnam, R. B. (2013). Efficient exact optimization of multi-objective redundancy allocation problems in series-parallel systems. *Reliability Engineering and Systems Safety*, 111, 154-163.]
- [Carrillo, V. M., & Taboada, H. (2012). A post-pareto approach for multi-objective decision making using a non-uniform weight generator method. *Procedia Computer Science*, 12, 116-121.]
- [Charnes, A., & Cooper, W. W. (1977). Goal programming and multiple objective optimizations: Part 1. *European Journal of Operational Research*, 1(1), 39-54.]
- [Coello, C. A. C., & Becerra, R. L. (2003, April). Evolutionary multiobjective optimization using a cultural algorithm. In *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE* (pp. 6-13). IEEE.]

- [Coello CAC., (n.d.). *Sample Pareto fronts*. Retrieved from <http://delta.cs.cinvestav.mx/~ccoello/EMOO/testfuncs/>]
- [Costa, D. M., Paula, T. I., Silva, P. A., & Paiva, A. P. (2016). Normal boundary intersection method based on principal components and Taguchi's signal-to-noise ratio applied to the multiobjective optimization of 12L14 free machining steel turning process. *The International Journal of Advanced Manufacturing Technology*, 87(1-4), 825-834.]
- [Crawford, V. P. (1977). A game of fair division. *The Review of Economic Studies*, 44(2), 235-247.]
- [Das, I., & Dennis, J. E. (1998). Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3), 631-657.]
- [Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms* (Vol. 16). John Wiley & Sons.]
- [De Weck, O. L. (2004). Multiobjective optimization: History and promise. In *Invited Keynote Paper, GL2-2, The Third China-Japan-Korea Joint Symposium on Optimization of Structural and Mechanical Systems, Kanazawa, Japan* (Vol. 2, p. 34).]
- [Dumitrescu, D., Lung, R. I., & Gaskó, N. (2011, May). Detecting strong Berge Pareto equilibrium in a non-cooperative game using an evolutionary approach. In *Applied Computational Intelligence and Informatics (SACI), 2011 6th IEEE International Symposium on* (pp. 101-104). IEEE.]
- [Erfani, T., & Utyuzhnikov, S. V. (2011). Directed search domain: a method for even generation of the Pareto frontier in multiobjective optimization. *Engineering Optimization*, 43(5), 467-484.]
- [Enquist, M., Ghirlanda, S., Jarrick, A., & Wachtmeister, C. A. (2008). Why does human culture increase exponentially?. *Theoretical population biology*, 74(1), 46-55.]
- [Fisher, L. (2008). *Rock, paper, scissors: game theory in everyday life*. Basic Books (AZ).]
- [Gao, J., & You, F. (2017). Life Cycle Optimisation from A Noncooperative Perspective: Game Theory-Based Models and Applications. In *Computer Aided Chemical Engineering* (Vol. 40, pp. 1915-1920). Elsevier.]
- [Gintis, H. (2007). The evolution of private property. *Journal of Economic Behavior & Organization*, 64(1), 1-16.]
- [Gong, D., Ji, X., Sun, J., & Sun, X. (2014). Interactive evolutionary algorithms with decision-maker's preferences for solving interval multi-objective optimization problems. *Neurocomputing*, 137, 241-251.]
- [Gorantiwar, S. D., & Smout, I. K. (2010). Multicriteria decision making (compromise programming) for integrated water resources management in an irrigation scheme.]
- [Haake, C. J., Raith, M. G., & Su, F. E. (2002). Bidding for envy-freeness: A procedural approach to n-player fair-division problems. *Social Choice and Welfare*, 19(4), 723-749.]
- [Han, Z. (2012). *Game theory in wireless and communication networks: theory, models, and applications*. Cambridge University Press.]

- [Hauert, C. (2006). Spatial effects in social dilemmas. *Journal of Theoretical Biology*, 240(4), 627-636.]
- [Heloulou, I., Radjef, M. S., & Kechadi, M. T. (2017). Automatic multi-objective clustering based on game theory. *Expert Systems with Applications*, 67, 32-48.]
- [Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994, June). A niched Pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on* (pp. 82-87). Ieee.]
- [Hwang, C. L., & Masud, A. S. M. (2012). *Multiple objective decision making—methods and applications: a state-of-the-art survey* (Vol. 164). Springer Science & Business Media.]
- [Izadbakhsh, M., Gandomkar, M., Rezvani, A., & Ahmadi, A. (2015). Short-term resource scheduling of a renewable energy based micro grid. *Renewable Energy*, 75, 598-606.]
- [Jaszkiewicz, A., & Słowiński, R. (1995). The Light Beam Search — Outranking Based Interactive Procedure for Multiple-Objective Mathematical Programming. *Nonconvex Optimization and Its Applications Advances in Multicriteria Analysis*, 129-146.]
- [Kasimbeyli, R. (2013). A conic scalarization method in multi-objective optimization. *Journal of Global Optimization*, 1-19.]
- [Kim, I. Y., & De Weck, O. L. (2006). Adaptive weighted sum method for multiobjective optimization: a new method for Pareto front generation. *Structural and multidisciplinary optimization*, 31(2), 105-116.]
- [Kim, I. Y., & de Weck, O. L. (2005). Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and multidisciplinary optimization*, 29(2), 149-158.]
- [Koh, A. (2012). An evolutionary algorithm based on Nash dominance for equilibrium problems with equilibrium constraints. *Applied Soft Computing*, 12(1), 161-173.]
- [Kulturel-Konak, S., Coit, D. W., & Baheranwala, F. (2008). Pruned pareto-optimal sets for the system redundancy allocation problem based on multiple prioritized objectives. *Journal of Heuristics*, 14(4), 335-357.]
- [Lung, R. I., & Dumitrescu, D. (2008). Computing nash equilibria by means of evolutionary computation. *Int. J. of Computers, Communications & Control*, 3(suppl. issue), 364-368.]
- [Li, X., Gao, L., & Li, W. (2012). Application of game theory based hybrid algorithm for multi-objective integrated process planning and scheduling. *Expert Systems with Applications*, 39(1), 288-297.]
- [Madani, K. (2010). Game theory and water resources. *Journal of Hydrology*, 381(3), 225-238.]
- [Mattson, C. A., Mullur, A. A., & Messac, A. (2004). Smart Pareto filter: Obtaining a minimal representation of multiobjective design space. *Engineering Optimization*, 36(6), 721-740.]
- [Messac, A., & Mattson, C. A. (2004). Normal constraint method with guarantee of even representation of complete Pareto frontier. *AIAA journal*, 42(10), 2101-2111.]
- [McMillan, J. (2013). *Game theory in international economics*. Taylor & francis.]

- [Miettinen, K., & Mäkelä, M. M. (1995). Interactive bundle-based method for nondifferentiable multiobjective optimization: nimbus §. *Optimization*, 34(3), 231-246.]
- [Miettinen, K., & Mäkelä, M. M. (1997). Interactive method NIMBUS for nondifferentiable multiobjective optimization problems. In *Multicriteria Analysis* (pp. 310-319). Springer, Berlin, Heidelberg.]
- [Moorthy, K. (1985). Using Game Theory to Model Competition. *Journal of Marketing Research*, 262-282]
- [Moulin, H. (2004). *Fair division and collective welfare*. MIT press.]
- [Myerson, R. B. (2013). *Game theory*. Harvard university press.]
- [Nawaz, A., & Toor, A. H. (2004). Dilemma and quantum battle of sexes. *Journal of Physics A: Mathematical and General*, 37(15), 4437.]
- [Osborne, M. J., & Rubinstein, A. (1994). *A course in game theory*. MIT press.]
- [Parsons, S., & Wooldridge, M. (2002). Game theory and decision theory in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 5(3), 243-254.]
- [Pavlidis, N. G., Parsopoulos, K. E., & Vrahatis, M. N. (2005). Computing Nash equilibria through computational intelligence methods. *Journal of Computational and Applied Mathematics*, 175(1), 113-136.]
- [Pinchera, D., Perna, S., & Migliore, M. D. (2017). A Lexicographic Approach for Multi-Objective Optimization in Antenna Array Design. *Progress In Electromagnetics Research M*, 59, 85-102.]
- [Rabin, M. (1993). Incorporating Fairness into Game Theory and Economics. *The American Economic Review*, 1281-1302.]
- [Regier, D. A., & Peacock, S. (2017). Theoretical Foundations of MCDA. In *Multi-Criteria Decision Analysis to Support Healthcare Decisions* (pp. 9-28). Springer International Publishing.]
- [Reynoso-Meza, G., Blasco, X., Sanchis, J., & Martínez, M. (2010, July). Multiobjective optimization algorithm for solving constrained single objective problems. In *Evolutionary Computation (CEC), 2010 IEEE Congress on* (pp. 1-7). IEEE.]
- [Saad, W., Han, Z., Debbah, M., Hjørungnes, A., & Basar, T. (2009). Coalitional game theory for communication networks. *IEEE Signal Processing Magazine*, 26(5), 77-97.]
- [Sefrioui, M., & Perlaux, J. (2000). Nash genetic algorithms: examples and applications. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* (Vol. 1, pp. 509-516). IEEE.]
- [Soltani, R., sadjadi, S. J., Tavakkoli-Moghaddam, R. 2015. Entropy based Redundancy Allocation in Series-Parallel Systems with choices of a Redundancy Strategy and Component Type: A Multi-Objective Model. *Applied Mathematics and Information Sciences* 9, No. 2, 1049-1059.]
- [Snidal, D. (1985). The game theory of international politics. *World Politics*, 38(1), 25-57.]

- [Srivastava, V., Neel, J. O., MacKenzie, A. B., Menon, R., DaSilva, L. A., Hicks, J. E., ... & Gilles, R. P. (2005). Using game theory to analyze wireless ad hoc networks. *IEEE Communications Surveys and Tutorials*, 7(1-4), 46-56.]
- [Syu, Y., FanJiang, Y. Y., Kuo, J. Y., & Ma, S. P. (2012, March). A Genetic Algorithm with Prioritized Objective Functions for Service Composition. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference on* (pp. 932-937). IEEE.]
- [Taboada, H. A., & Coit, D. W. (2008). Multi-objective scheduling problems: Determination of pruned Pareto sets. *IIE Transactions*, 40(5), 552-564.]
- [Tadenuma, K., & Thomson, W. (1995). Games of fair division. *Games and Economic Behavior*, 9(2), 191-204.]
- [Thomson, W. (1983). The fair division of a fixed supply among a growing population. *Mathematics of Operations Research*, 8(3), 319-326.]
- [Velasquez, M., & Hester, P. T. (2013). An analysis of multi-criteria decision making methods. *International Journal of Operations Research*, 10(2), 56-66.]
- [Venkat, V., Jacobson, S. H., & Stori, J. A. (2004). A post-optimality analysis algorithm for multi-objective optimization. *Computational Optimization and Applications*, 28(3), 357-372.]
- [Wierzbick, A. P. (1977). Basic properties of scalarizing functionals for multiobjective optimization. *Optimization*, 8(1), 55-60.]
- [Zeleny, M. (1973). Compromise programming. *Multiple criteria decision making*.]
- [ZHU, Q. H., & DOU, Y. J. (2007). Evolutionary game model between governments and core enterprises in greening supply chains. *Systems engineering-theory & practice*, 27(12), 85-89.]
- [Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm.]

Vita

Juan V. Fernandez was born in El Paso, Texas. He received his B.S. in Industrial Engineering in December 2015. After graduating, he started his Masters of Science in Industrial Engineering and received a National Science Foundation scholarship. In August 2014, he became a research assistant at the Sustainability Engineering and Optimization Lab (SESOL), where he continues to work. He has been part of two study abroad experiences. In summer 2014 in Peru, “Global and regional Sustainability experience”. In summer 2016 Ensenada, Mexico, “Engineering Together Sustainable Communities Program”. He has been an intern with the United States Department of Agriculture (USDA) with the Agriculture Research Service (ARS), and with the National Institute of Food and Agriculture (NIFA) respectively. His research interests include optimization, sustainability, supply chain management and manufacturing. He has two peer-reviewed publications. “Multiple Objective Optimization of a Biomass to Bio Refinery Logistics System” in Industrial and Systems Engineering Research Conference (ISERC), and “Global Warming Potential and Cost Minimization for the Centralized Carrier Collaboration and Multi Hub Location Problem” in International Conference on Industrial Engineering and Operations Management (IEOM) in this last one his work obtained best track paper award. Additionally, he had the opportunity to present his research work at other conferences where he obtained first place poster presentation award at Workforce Diversity and Career Opportunities within the USDA for Current and Recent Graduates (USDA NIFA HSI Program Conference). He is a member of IISE, and is Green Belt Six Sigma certified.

Contact Information: jvfernandez77@gmail.com

This thesis was typed by Juan V. Fernandez