Universidad de Alcalá

Doctorate Programme in
Information and Knowledge Engineering

On the Application of Multiobjective Optimization to Software Development Process and Antenna Designing

PhD. Thesis Presented by

Javier Moreno Martín

2020

Universidad de Alcalá

Doctorate Programme in

Information and Knowledge Engineering

On the Application of Multiobjective Optimization to Software Development Process and Antenna Designing

PhD. Thesis Presented by

Javier Moreno Martín

Advisors

Daniel Rodríguez García y José Javier Martínez Herraiz

Alcalá de Henares, 6th of June, 2020

A mis padres, maestros de maestros

Jean Paul Sartre dijo una vez, que somos lo que hacemos de lo que hicieron de nosotros.

Este concepto y todo lo que implica, me resulta, ahora más que nunca, el compromiso más delicado, complejo y gratificante de todos.

Con esta miscelánea de posibilidades, decidisteis hacer de mí uno de los proyectos más importantes de vuestra vida, poniéndola a mi entera disposición, emprendiendo conmigo mi propio viaje.

Caminásteis, siempre a mi lado, allanando el camino, evitándome obstáculos y piedras que con gusto hacíais vuestras, dejándome ir, pero acompañándome siempre, porque siempre estáis conmigo.

Por la dulzura, el cariño, la protección, la sabiduría, el coraje, el valor, la comprensión, la paciencia, por no rendiros nunca, por vuestro amor, el más absoluto e incondicional; gracias.

Por esto, por mucho más que esto, por todo, esta tesis, este tramo del camino lleva vuestro nombre.

Con el anhelo de haber hecho un mínimo de justicia a aquello que hicisteis de mí un día, y todos,

¡Gracias!

To my parents: wise teachers of mine

Jean Paul Sartre once said "freedom is what we do with what is done to us"

This conception and everything it implies, appeals to me, now more than ever, the most delicate, complex, and grateful compromise so far.

With regard to this blend of possibilities, you decided to do of me one of the most important projects of your life, placing your existence at my entire disposal, taking me along my own journey.

You have always walked next to me, paving the way for me beforehand, making it smoother, preventing me from any obstacles and stones, making them entirely yours, then you had to let me go, and I left, but always with you by my side, because you are always with me.

For all your sweetness, your protection, your wisdom, your courage, your bravery, your understanding, your patience, for never giving up, and above all, for that absolute and unconditional love of yours towards me; thank you.

For this, for a lot more than this, my thesis, this stretch of the way, has your name written on it.

My greatest desire, is to be able to give you back just a small part of what you did for me one day, and every single day,

Thank you!

Acknowledgements

And the coin went through the marble.

Ricardo Martín

I would like to start by thanking my wife Alícia for her endless patience, her help in countless and difficult moments, for being the source of my inspiration when I have most needed it; I also want to thank her for the unconditional support she has always given me. Also to my son Èric, who was born while developing this thesis, and has become the most important project in my whole life. Both of them are my light and my greatest joy, thank you for giving me so much peace and quietness, thank you for providing me with the combination of time and support that I needed to carry on with this thesis.

I would also like to express my most sincere gratitude to my mentors Daniel Rodriguez and José Javier Martinez, for giving me the necessary guidance throughout this tough way that has involved a great effort and dedication. I must admit, that if it wasn't for Daniel this thesis would have never existed; Daniel, you encouraged me to start and helped me to continue when climbing uphill was getting so hard. Thank you.

I also want to thank Iván Gonzalez, José Antonio Lozano, Antonio J. Nebro, Mercedes Ruiz and Bernabé Dorronsoro for your advice, your teachings and your valuable assistance across this grateful experience that I have been lucky to share with you. Its been a great pleasure for me to work with such brilliant people.

I also want to express my gratitude to my whole family, all of them wonderful people, for being there every single day, in the good times, as well as the bad. Especially my parents and sisters, the ones I grew up with, and I still do each day; I carry you both in my heart, I am so proud you. Thank you for your laughter, for your songs, your stories, for all our moments, for being there, and sharing your world with me. To my little niece Ana, who fills everything with joy and is always giving us so funny moments.

And last but not least, to my friends, I have to say that you are always in my mind, those who have in someway struggled with me, always by my side. Greetings and affections for Jorge Abad for all his technical and scientific advice while the elaboration of this thesis.

Resumen

Esta tesis doctoral, presentada como compendio de artículos, explora los beneficios prácticos del uso combinado de la optimización multi-objetivo con aplicaciones de simulación.

En esta tesis, con un caracter de aplicación, se aportan ideas prácticas sobre cómo las meta-heurísticas combinadas con la simulación, es decir, la optimización de la simulación, pueden proporcionar soluciones a problemas del mundo real. Esta tesis comienza presentando un caso de técnicas de optimización multiobjetivo a través de la simulación para ayudar a los directores de proyectos de software a encontrar las mejores configuraciones para los proyectos basados ITIL (*Information Technology Infrastructure Library*), de manera que se optimicen las estimaciones de calendario para un proyecto determinado, el tiempo y la productividad. Los datos de gestión de proyectos pueden obtenerse mediante simulación, por ejemplo, para optimizar el número de recursos utilizados en cada fase de la vida del proyecto.

También se presenta otro caso de estudio sobre la forma en que la optimización de la simulación puede ayudar en el diseño de cualquier tipo de antena. En este caso de estudio, el objetivo es lograr una antena helicoidal, de doble banda, lo más compacta posible, para la telemetría, el seguimiento y el control (TTC) de los satélites. En los satélites es esencial reducir el volumen y el peso de los dispositivos instalados, manteniendo al mismo tiempo los requisitos de funcionamiento.

Adicionalmente, esta tesis realiza un aporte teórico proponiendo un nuevo algoritmo que mejora el rendimiento de los algoritmos de optimización multi-objectivo basados en el cálculo del Pareto front.

Palabras clave: Optimización Multiobjetivo, Algoritmos Evolutivos, Simulación, Diseño de Antenas, ITIL, Pareto Front.

Abstract

This thesis, presented as a collection of articles, explores the practical benefits of using multi-objective optimization combined with simulation applications.

In this thesis, with practical application, we provide practical insights about how meta-heuristics combined with simulation, i.e., simulation optimisation, can provide solutions in real world applications. This thesis begins by presenting a case of multi-objective optimisation techniques via simulation to help software project managers find the best configuration for ITIL (Information Technology Infrastructure Library) based projects, such that schedule estimates for a given project, time and productivity are optimised. Project management data can be obtained via simulation, for example, to optimize the number of resources used in each phase of the project life.

Another case study is also presented about how simulation optimisation can help with the design of any type of antenna. In this case study, the objective is to achieve a dual-band helical antenna, as compact as possible, for Telemetry, Tracking, and Control (TTC) of satellites. Reducing the volume and weight of any devices installed on satellites while maintaining their operational requirements is of paramount importance.

Finally, an algorithm will be presented that improves the performance of a part of some meta-heuristic multi-objective optimization algorithms, known as the Pareto front calculation.

Keywords: Multi and Many-Objective Optimization, Evolutionary Algorithms, Simulation, Antenna Design, ITIL, Pareto Front.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Predicting what will happen in certain situations and knowing which is the best decision to make can be very difficult.

The problem of decision making is as old as humanity itself. Everyday, we face situations or problems where we have to choose from a set of possible solutions. Once the decision criteria associated with the problem have been identified, it is necessary to study the possible alternatives in order to choose the most advantageous one. In many real cases we also deal with problems with many variables that need to be optimized at the same time, and usually these variables to be optimized are compromised with each other.

On the other hand, having the chance to experiment with certain situations before facing them can be very advantageous, and simulation tools are an excellent tool for doing so.

Design optimization in any engineering field, process or economic/financial parameters are clear examples where multi- and many-objective optimization techniques can help to find optimal solutions to the problems at hand.

These are multi-objective optimization problems (MOOP), and each element presents a set of characteristics that, depending on the case, we will want to maximize or minimize. Choosing, therefore, means arranging the elements, so that, after arranging, the most advantageous are the first ones. Unfortunately, ordinary sorting algorithms do not allow to sort a set of variables that may be contradictory to each other. As we will see later, there are many algorithms and methods to solve this type of problems. Some algorithms offer a set of optimal solutions, while others offer only one solution. Depending on their own needs, the decision maker will have to consider which algorithm is the most suitable to solve the problem.

From the decision maker's point of view, multi-objective optimization algorithms can be classified into [1]:

- A priori: Algorithms that use a priori methods require the decision maker to establish a previous order of each objective, before launching the optimization process.
- A posteriori: In this case, the decision maker launches the optimization process and then chooses among the solutions obtained by the process.
- Interactive: The algorithms within this classification request additional information from the decision maker during the execution of the optimization process.
- No-preference methods: In these methods, the opinion of the decision maker is ignored and the problem is resolved by whatever method is deemed appropriate. The decision maker will only decide whether to accept or reject the solution obtained.

In this thesis, we provide practical insights about how a posteriori meta-heuristics combined with simulation, i.e., simulation optimization, can provide solutions in real world applications. This thesis begins by presenting a case of multi-objective optimization techniques via simulation to help software project managers find the best configuration for ITIL-based projects such that schedule estimates for a given project, time and productivity are optimized. Project management data can be obtained via simulation, for example, to optimize the number of resources used in each phase of the project life.

Another case study is also presented about how simulation optimization can help with the design of any type of antenna. In this case study, the goal is to achieve a dual-band helical antenna as compact as possible, for Telemetry, Tracking, and Control (TTC) of satellites. Reducing the volume and weight of any devices installed on satellites while maintaining their operational requirements is of paramount importance.

Finally, an algorithm that improves the performance of a part of some meta-heuristic algorithms for multi-objective optimization, known as the Pareto front calculation, will be presented.

1.2 Aim and Objectives

The main aim of this thesis is to show how a posteriori meta-heuristic algorithms can be combined with simulation to provide solutions to real problems. Also, a major issue was how to speed up the execution of such algorithms and we also show an important new algorithm. To do so, we intend to achieve this through several objectives:

Objective 1: Study the different multi-objective optimization algorithms.

Objective 2: Study of simulation optimization in software development projects.

Objective 3: Study of multi-objective approaches in antenna design, combined with simulation of antenna radiation patterns.

1.3 Outline of the Thesis

This thesis is divided into two parts: (i) main contributions on simulation based optimization with multi-objective approach, and (ii) selected refereed publications.

In Part I, an introduction to multi-objective optimization is described in Chapter 2. This introduction serves as the basis for understanding the contributions made in this thesis. After providing a brief introduction, in Section 2.3, multi-objective concepts are introduced. Sections 2.4 and 2.5 show the most known algorithms, and Section 2.6 presents the metrics for evaluating this type of algorithm.

Chapter 3 introduces the need and applications of multi-objective optimization. Section 3.2 presents the application of multi-objective optimization in the field of Simulation applied to Information Technology Service Management, while Section 3.3 shows the use of multi-objective optimization to solve the problem of TTC antenna design. Finally, Section 3.4 presents the theoretical contribution by means of a new algorithm, called Merge Non-Dominated Sorting (MNDS), which improves the calculation times of the ranking of the solutions and their Pareto Front. This algorithm presents, in the best case, a computational cost of $O(NlogN)$ and a worst case of $O(MN^2)$. MNDS was proven to be the most efficient algorithm of all the existing ones at the time of writing this thesis. Finally, Chapter 4 presents the conclusions, in Section 4.1, raises potential future challenges for further progress, in the field of multi-objective optimization, in Section 4.2, and in Section 4.3, presents the works published during the elaboration of this thesis.

In Part II, the refereed journal publications derived from this work are included. Chapters 5 and 6 describe the practical and real applications of simulation optimization with meta-heuristics. The former is in the domain of software engineering, which is also known as Search based Software Engineering (SBSE) [2]. This term, SBSE, refers to the application of optimization techniques to the field of Software Engineering. The Software Engineering presents many optimization problems where constraints and conflicts between the variables are given to optimize, from testing to project management as it is the case in this work. The latter is in the domain of antenna design. Next, Chapter 7 presents a more theoretical publication improving the running time of all algorithms that calculate the ranking of Pareto front.

Part I

Main Contributions on Simulation based Optimization with Multi-objective approaches

Chapter 2

Multi-Objective Problems in a Nutshell

2.1 Introduction

Many of the problems we face, whether in personal or professional life require to consider several objectives to be optimized in a simultaneous way. In these situations, the objectives have conflicts between them and cannot be optimized separately. Multi-Objective Problems (MOOPs) have an additional challenge. There are several functions to optimize (each function is an objective), and optimizing one often implies not-optimizing the other one. Therefore is necessary to find solutions to the problem that individually optimize each objective and also, be feasible with the rest of solutions found for the remainder objectives.

In this chapter, we will show different algorithms that deal with these type of problems.

2.2 Multi-objective Algorithms

In this section, we provide a brief introduction to a type of metaheuristics specializing in optimization, known as multi-objective algorithms. As stated in Section 1.1, traditionally, algorithms dedicated to solving MOOPs are classified from the decision maker's point of view into: (i) a priori, (ii) a posteriori, (iii) interactive and (iv) non-preferential.

The a priori algorithms define a total order in the objective space[1]. Additionally, the decision maker must establish his preferences or criteria to discriminate and apply the order of the solutions. In this way, the decision maker is guiding the search for solutions in advance. The lexicographic sorting is the most used a priori algorithm. When a problem has several objectives, it consists of prioritizing these objectives. The decision maker prioritizes and sets an arbitrary order of all the objectives, so that the first objective will always be the most important. In lexicographic order, when two solutions are compared,

[1]This concept is introduced in Section 2.3

the one whose first objective is better, is always the best solution. In case of a tie on the first objective, the second one is compared and so on, until one solution is considered better than the other.

On the other hand, the a posteriori algorithms perform an unguided search for solutions. These methods generate a set of solutions from which the decision maker must choose the best one. The solutions are distinguished by applying a partial order on the objective space. Normally, Pareto order is applied and the algorithm searches for solutions that minimize it, obtaining what is known as Pareto front. The Pareto front whose solutions can no longer be minimized is known as Pareto optimal. All a posteriori methods try to obtain as many solutions as possible within the Pareto optimal and at the same time, they try to cover the whole Pareto optimal. Two big groups gather the multi-objective optimization algorithms, those based on mathematical programming and those based on metaheuristics. Multi-objective metaheuristics are described in Section 2.5. Two of the best known algorithms have been applied in this thesis: NSGA-II [3] and SPEA2 [4].

There are also interactive algorithms, which allow the decision maker to guide the search while the algorithm is running.

Finally, no preference methods solve the problem without the need for interaction with the decision maker. Once a solution is obtained, the decision maker can accept or reject it. One of the best known non preference methods is the Global Criterion. This method is based on minimizing the distance to the ideal objective vector[1].

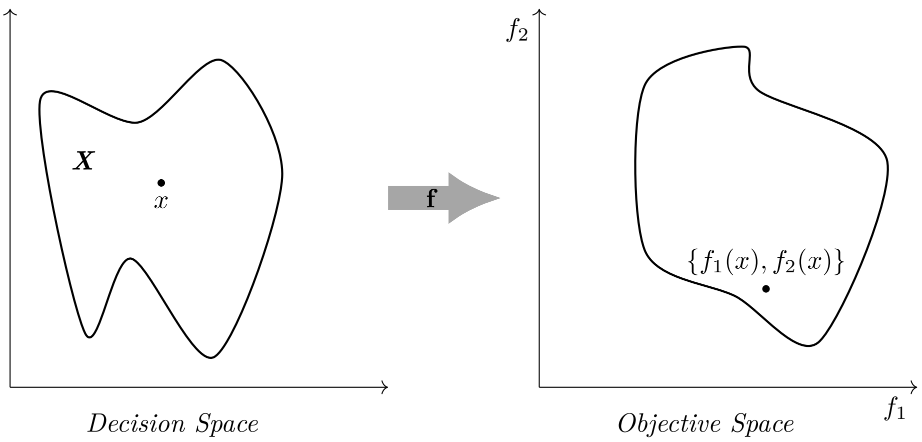
2.3 Multi-objective optimization concepts

Given a set of functions $f = \{f_1(x), f_2(x), ..., f_M(x)\} : M \geq 2, x \in X$,where M is the number of objective functions to optimize, and x a decision vector, within a feasible region X, defined by constraint functions, the goal is to minimize all objective functions simultaneously. It is important to understand well the meaning of the previous expression. The variable x is a vector with the value of some concrete parameters (decision space), which characterize a possible solution to the problem.Applying a set of functions to the values of these parameters, we will obtain a set of results, which are the objectives to optimize (objective space). These objectives are the ones that should be minimized simultaneously. The Figure 2.1 shows the relationship between the decision space and the objective space.

In the case of an antenna, for example, these parameters could be the length and width of the antenna.Applying different electromagnetic functions we would obtain the radiation values of the antenna for a frequency, as could be the gain and the cross-polar polarization level, which would be the objectives to optimize.

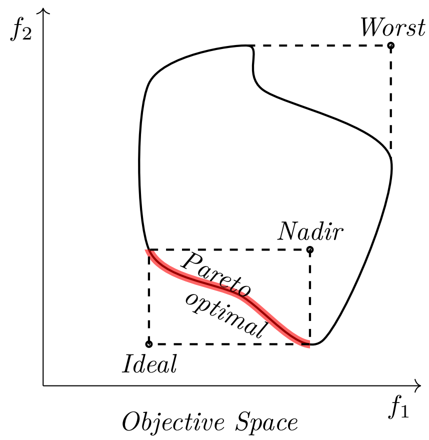
Each solution, in the decision space, has a representation in the objective space, by means of an m-dimensional vector, known as the objective vector. In the case of minimizing, the smallest possible objective vector would be the one that minimizes each of

Figure 2.1: Parameter and objective spaces



the objective functions separately, something that in reality does not happen, since the objectives are usually contradictory. Similarly, the worst vector would be the one that maximizes each of the objective functions separately. In the figure, both objective vectors are represented as points. The objective vectors (Pareto front) closest to the ideal vector are those within what is known as the Pareto optimal. All these points are equally valid solutions. All the a posteriori algorithms try to find as many solutions within this front as possible. The decision maker will choose one of these solutions. The nadir objective vector is the starting point for the interactive algorithms: It is the worst Pareto optimal solution, and from there, through the interaction with the decision maker, the search is guided until a solution is reached in the Pareto optimal. Note that in real life problems, it is very difficult to obtain the Pareto optimal.

Figure 2.2: Ideal, nadir and worst objective vectors



2.3.1 The Pareto front

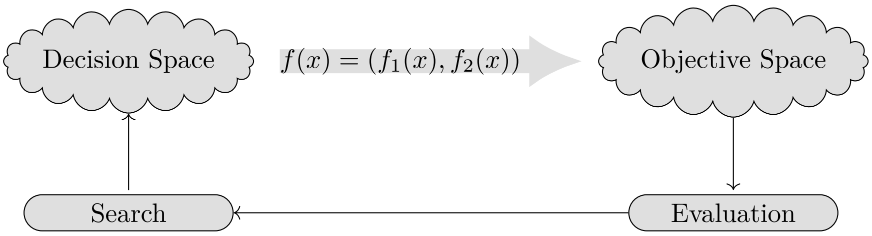
The Pareto front concept allows to select those solutions which are best together. It is considered that a solution dominates another when the former offers better value (more optimal) for at least one objective and is the same for all other objectives of the problem.

Iteration after another, the multi-objective algorithms find solutions that dominate those of the previous iteration, until at last the solutions not dominated by any other remain. The non-dominated solutions form the Pareto front.

Let P be a population of N solutions, $\{s_1, \ldots, s_N\} \in P$, where each solution contains a vector of M objectives to minimize, $(f_1(s_i), \ldots, f_M(s_i)), \forall i \in \{1, \ldots, N\}$. A solution s_i dominates a solution s_j, denoted by $s_i \preceq s_j$, if the objective vector of s_i is partially less than the objective vector of s_j, i.e., $\forall m \in \{1, \ldots, M\}, f_m(s_i) \leq f_m(s_j) \wedge \exists m' \in \{1, \ldots, M\}$ *s.t.* $f_{m'}(s_i) < f_{m'}(s_j)$ (assuming minimization without loss of generality). Given a set of solutions, it is possible to establish a partial order between them, known as the Pareto ranking. Those solutions which are non-dominated by any other are assigned rank 1. If these solutions are removed, then those solutions which are non-dominated by any other are assigned rank 2, and so on, the Pareto ranking is built.

To find the Pareto front, the algorithms must look for those variables in the decision space that minimize the objective functions in the objective space. Figure 2.3 shows the search process.

Figure 2.3: Pareto front search.



The calculation of the objective vector nadir is necessary in the interactive methods. Normally it cannot be obtained and the aim is to obtain good estimates using different metaheuristics. Deb et al, for example, proposed in 2010 to use a hybrid algorithm of evolutionary - local search [5].

2.4 Metaheuristics

Metaheuristic (term coined by Glover[6]) algorithms are the last option when solving problems, even behind brute-force algorithms. Metaheuristic algorithms are able to deal with very large search spaces to find a good (may or not be the best) solution applying a certain heuristic, in an iterative way, for a number of iterations or until the solution found fulfils the requirements and is good enough as to stop the search.

The resolution of Multi-objective problems present two major difficulties:

- The computational complexity involved in working with more than two objectives, today is $O(MN^2)$, where M is the number of objectives and N is the number of solutions to the problem.

- Objectives that are contradictory or conflictive between each other, which prevent the application of a priority or order of some objectives over others.

Due to these drawbacks, the use of metaheuristic has become widespread, in an attempt to obtain good solutions in acceptable times.

As the algorithms are generic and independent of the problem that solve, often they incorporate mechanisms to avoid searches in areas where there are no known optimal solutions.

There are many possible classifications depending on different aspects such as (i) Local Search (LS) vs. Global Search (GS), (ii) Single-Solution (SS) vs. Population Based (PB), (iii) Hybridization (HY), (iv) Memetic (MM) algorithms, (v) Parallel metaheuristics (PM), and (vi) nature-inspired (NI) and metaphor-based (MB) metaheuristics. We next present a set of well-known metaheuristics classified according to type of search or whether they are based on single or population based solutions.

Local search algorithms are possibly the oldest metaheuristics. Many of them belong to the folklore of computer science. For example, the first known version of the Gradient Descent algorithm was proposed by Louis Augustin Cauchy in 1847. Many of these algorithms have versions to extend the search when they fall into local minimum or maximums. Hybridization metaheuristics are combinations of two or more metaheuristics. In particular, Memetic metaheuristics combine population-based algorithms (usually genetic algorithms) with individual/local search techniques. Finally, parallel metaheuristics apply concepts and techniques of computational parallelization to existing metaheuristics. Currently, the most popular algorithms are those based on population and/or inspired by nature.

The following are some of the best known metaheuristic algorithms.

Hill Climbing (LS)(SS). This algorithm is based on random search. It starts with a random position and at each iteration the algorithm explores the neighbourhood by selecting the best solution found. Many times this algorithm stops searching without finding an acceptable solution because it falls in a local maximum (or minimum when minimizing).

Gradient Descent (LS)(SS). This algorithm is based on going through a function in an iterative way, always moving in the direction that minimizes the value of that function. In case of reaching a point where the value of the function increases, the algorithm goes back to the previous point and ends. This algorithm, like Hill Climbing, is sensitive to local maximum / minimum.

Tabu Search (LS)(SS). Like hill-climbing or Gradient Descent algorithms, tabu search [7] follows a path. It uses a memory search to which applies a strategy for analyzing and exploring different areas of the search space. The name is due to the recent memory is implemented by a tabu list. In each iteration is chosen the best solution among the permitted (solutions already on the list are not allowed) and added to the tabu list. The tabu list maintains the latest solutions, which over the iterations may end being replaced by better ones.

Simulated Annealing (GS)(SS)(NI)(MB). It is one of the oldest metaheuristics, developed by Kirkpatrick et al. in 1983 [8]. Simulates the anneal of steel, ceramics or glass by applying heat and slow cooling, which causes the atoms to become repositioned achieving the hardening of the material. In each iteration a solution from the current population is chosen. If the best solution is chosen it replaces the current and if not, is still accepted with a certain probability of being chosen which avoids falling into local optimums.

Evolutionary Algorithms (GS)(PB)(NI). These algorithms try to imitate the evolutionary capabilities of living beings in nature, where only those best adapted to their environment survive. Given a population, each individual can be a solution. At each iteration the population is altered by three operators: selection, recombination and mutation, leading to a new population. In evolutionary algorithms, genetic algorithms are best known [9].

Particle Swarm Optimization (PSO) (GS)(PB)(NI)(MB). This algorithm, published in 1995 by Kennedy et al. [10] is inspired by the collective behavior of some birds that while flying behave as a super-organism, or the movement of schools of fish. Given a population of individuals (called particles in the algorithm) moving along the search space, each particle can change its speed or position based on the best position found by the particle, to the best position found by the entire population, or the position of neighboring particles.

Ant Colony Optimization (GS)(PB)(NI)(MB). This algorithm [11] is inspired by the behavior of ants when they are searching for food. Initially, ants randomly explore an area near the anthill, while they do it the ants leave by rubbing their belly to the ground a substance called pheromone, which is recognized by the other ants. At the same time that the ants go through the same road, pheromone accumulation in soil is greater and more ants decide to use this route, until at last the shortest route between food and anthill is established.

Bee Colony Algorithm (ABC) (GS)(PB)(NI)(MB). This algorithm [12] simulates the behavior of honeybees in the search of pollen. The algorithm assigns three different roles to bees: workers, observers and explorers. A bee can take more than one role depending on the work they have to do. The explorer bees randomly overfly long distances within the search space, and when they found an area with more food, return to the hive for reporting the location to the observers bees that are waiting. Part of bees assume the worker role and move to the area to collect the food, the rest remain as observers. The worker bees, once they reach the defined area, collect and search more food locally. When the worker bees return to the harvest, report whether or not the food source is abundant. Over time, the algorithm stores the solutions with greater abundance of food. This algorithm balances the local search, done by the worker bees, with the global search, done by the explorer bees.

Firefly Algorithm (GS)(PB)(NI)(MB). The algorithm [13] imitates behavior intermittently bioluminescence of the fireflies. Each firefly has a pattern (a possible solution) which is related to the attractiveness of the insect. The attraction is proportional to the brightness and distance to other fireflies. If all fireflies have similar brightness, the fireflies move randomly until there are different brightness intensities.

Gravitational Search Algorithm (GSA) (GS)(PB)(NI)(MB). It is based on Newton's theory of gravitation [14]. The population is formed by individuals with a given mass which attract each other. Each individual is a solution and at each iteration those heavier individuals attract towards them a lighter individuals, so that, in the end, the individual with a larger mass is the best solution.

2.5 Multi-objective Metaheuristics

This section presents some of the most popular and relevant multi-objective algorithms. As we will observe, many of these algorithms are based on the on the concepts previously described in section 2.4, that use as fitness the criteria of dominance between solutions.

Multi-Objective Particle Swarm Optimization (MOPSO). Algorithm PSO version adapted to multi-objective [15] that uses a mutation operator applied to individuals, and the range of each objective function, in order to cover the whole objective space.

Multi-Objective Firefly (MO-FA). This behavior-based firefly algorithm imitates the patterns of light intensity and behavior of fireflies attracting of those with a higher light intensity. In the multi-objective version, MO-FA [16], brightness is obtained from objective functions.

Multi-Objective Gravitational Search Algorithm (MO-GSA). MO-GSA [17] is the multi-objective version of the algorithm based on the physics of Newton. The biggest problem in this algorithm is to update the mass of each individual value from multiple objectives. To do this, the value of the mass of each particle is initially established and then the mass is updated according to the distance of that particle from its closest neighbours. Once the mass is updated, each particle moves to its new location by applying the gravitational force.

Multi-Objective Artificial Bee Colony (MOABC). MOABC [18] is similar to ABC algorithm except that now runs within an evolutionary algorithm. In each iteration, MOABC generates a new population of bees by applying mutation operators and selecting the best individuals between the new and the current population. The new population search food sources which lead to new solutions, which will be selected or rejected depending on they are or not dominated.

Multi-Objective Cellular Genetic Algorithm (MOCell). It is a cellular genetic algorithm (CGA) [19] that stores in a file the non-dominated solutions found. In each iteration, the algorithm randomly selects an individual from the file and one from the current population, with which makes the crossover and the mutation operations. The new population is compared with the current one and the best population is selected and stored in the archive.

Pareto Archived Evolution Strategy (PAES). This is one of the simplest multi-objective algorithms. PAES [20] performs a local search maintaining the current solution rather than a population. There are several versions of this algorithm, some of which maintains a population of solutions.

Strength Pareto Evolutionary Algorithm SPEA2. SPEA2 [4] is based on maintaining the non-dominated solutions, found in each iteration, stored in a file. Using the fitness function, each individual in the population is associated with the number of solutions that dominates and is dominated. When the population is too large, the kth nearest-neighbor solutions are removed.

Non-dominated Sorting Genetic Algorithm (NSGA II). NSGA-II [3] is possibly the most famous algorithm. It is a genetic algorithm that keeps the population ordered according to their Pareto ranking, so that the best individuals, those who are non-dominated are the first individuals in the population. To find the uniform dispersion among the solutions, apply the algorithm known as crowding distance.

2.6 Evaluation Metrics

To assess the quality of the solutions returned by the multi-objective algorithms, several metrics are usually considered, among which spread, hypervolume or coverage ratio stand out.

The greater difficulty to assess the results obtained by multi-objective algorithms is to find metrics that demonstrate the quality of the algorithms and the solutions with a reasonable computational cost. Each algorithm generates different solutions in the Pareto front, and when compared among them, some solutions tend to dominate others, but usually is not clear that an algorithm be better than another. The metrics are based on: cardinality, convergence, spread, uniformity and combinations of them.

In the following sections, each group of metrics is briefly explained, indicating the best known or most representative. There are currently many metrics. In [21] a collection of 100 quality indicators is presented.

It is important to highlight that some quality indicators present characteristics of two or more of the four groups mentioned. Among these indicators, it is worth highlighting the *hypervolume* [22], which is currently undoubtedly the most widely used indicator.

The *hypervolume* indicates the volume covered by the individuals of the population of the Pareto front (non-dominated solutions) in relation to the volume of the objective space. The volume is calculated from the Pareto front set and the worst objective vector (see figure 2.2). *Hypervolume* has quality characteristics of cardinality, convergence and spread. Computationally is really hard to calculate.

Note that to obtain some indicators, it is necessary to know the set of solutions that form the ideal or optimal Pareto front. In many problems in real life, it is difficult or impossible to obtain such Pareto-Optimal so these indicators can only be applied in theoretical problems, where the Pareto-Optimal is known in advance.

2.6.1 Cardinality

The simplest metric consists of counting the number of the solutions that are find in the Pareto front. This value can be calculated at each iteration for assess also the evolution in the calculation of the Pareto front. The quality index *error rate* [23] calculates the percentage of solutions obtained by the algorithm, the Pareto front, with respect to the Pareto-Optimal. The indicator *coverage* [24] is another indicator of cardinality based on comparing solutions of Pareto front obtained by applying the same problem to two algorithms, and determining which of the two algorithms provides more solutions throughout the entire objective space. To do this, it is calculated the percentage of solutions of an algorithm that are dominated by the solutions of the other algorithm and vice versa.

2.6.2 Convergence

Convergence is the most important quality feature of a solution set as it indicates the closeness of a solution set to Pareto-Optimal (also known as true Pareto front). One of the best known metrics is the *generational distance (GD)* [25]. This metric is based on the distance between each solution in the Pareto front obtained by an algorithm and the closest Pareto-optimal solution. Another well known metric, the *inverted generational distance* [26] is based on the distance of each solution in the Pareto Optimal to the solutions obtained by an algorithm.

2.6.3 Uniformity

Uniformity represents the way solutions are distributed in the sense of equidistance. Given two sets of solutions, both in the Pareto front, the one that presents the most evenly distributed solutions will have the highest quality. There are different metrics. There are different quality indicators of this type. For example, the *u-measure* [27], is based on the relationship between the distance of two neighbouring solutions with respect to the ideal distance (obtained as the average of the distances between neighbouring solutions).

It is important to note that uniformity alone does not guarantee diversity of solutions, so it must be used in combination with Spread metrics.

2.6.4 Spread

Spread measures the distribution of the solutions along the Pareto front. A set where the solutions are distributed along the Pareto front, will have a good spread. Note that solutions can be spread along the Pareto front, but not evenly. So this metric should be used together with uniformity metrics to ensure the quality of the solutions obtained. One of the best known metrics of this type is the *maximum spread* [28] metric.

Chapter 3

Main Contributions of this Thesis

3.1 Introduction

Simulators have been demonstrated to be very useful tools for decision support systems to help decision makers in their activity. We can find many examples in the literature, with different fields of application as logistics [29], planning [30], economics [31, 32], or supply chains [33], among many others.

There are three major paradigms applied the field of simulation:

- Discrete-event simulation (DES). Based on sequential event processing. The state of the system changes in specific moments of time, depending on the events processed between that moment of time and the previous one. Manufacturing systems, order processing, or vehicle movement can be modeled with DES.
- System Dynamics (SD). Used to model very complex systems, abstracting from details or properties of individual elements. Examples of application can be marketing campaigns, the study of natural ecosystems or social behaviours.
- Agent-based Model (ABM). It is based on the concept of the agent as an independent and autonomous element within the system. Each agent has its own states and can interact with the rest of the system.

Many simulation optimization use the discrete-event paradigm as the simulation approach. In addition, there are also some applications of simulation optimization using the System Dynamics simulation approach. For instance, Ng [34] reported an approach for integrating simulation and optimization of System Dynamics models using Matlab™ and Simulink™ and demonstrated how to combine genetic algorithms, fuzzy logic expert input and System Dynamics modelling for improving decision-making. They applied their approach in the classical market growth model. Kremmel *et al.* [35] developed a System Dynamics simulation model to analyse the dynamics of city problems and city development under three types of policy interventions. They used genetic algorithms for maximising the benefits of policy decision making. A review of the state of the art of the simulation optimization can be found in [36].

3.2 Optimization applied to Information Technology Service Management

Information systems are getting bigger. They involve more people, more resources and their management is getting more complicated. To assist in the Information Technology Service Management, different frameworks have appeared, such as COBIT[1] (1996), the ISO/IEC 20000 (2005) standard, or ITIL[2] (1989) which is the best known and possibly most applied today. Most of the processes described in these frameworks can be simulated, which allows to evaluate different scenarios, before starting the IT project.

Multi-objective optimization algorithms can effectively handle the optimization of different conflicting objectives simultaneously, offering a wide and diverse range of trade-off solutions to the problem, helping the IT manager to make the most appropriate decisions. In order to guide the search towards high-quality solutions for such a complex problem involving a large number of processes, interactions among them, and uncertainties, these algorithms must rely on accurate simulations. This approach is explicitly proposed in frameworks to improve decision-making in the ITSM scope, such as the Sim4ITSM framework, which includes activities of simulation optimization in the experimentation phase of the method [37]. For a more comprehensive literature review, please, read the work by Orta *et al* [37]

In this work, the simulation is carried out on the ITIL *change management process*. This process addresses the problem of controlling the life cycle of a project. The problem of *IT change process efficiency* was modeled using Anylogic™. One of the critical decisions that change process managers need to make relates to the configuration of the process staff, as different staff configurations may lead to different outcomes. This problem configures with 10 parameters the staff involved in the fulfillment of the change management process. Staff is grouped into three different roles: (i) change management, (ii) change developer, and (iii) change deployer. Also, for each rol, there are four possible basic shifts:

- Central, from 9 am to 6 pm, with one-hour break at 1 pm.
- Early, from 6 am to 3 pm, with one-hour break at 10 am.
- Late, from 2 pm to 11 pm, with one-hour break at 6 pm.
- Night, from 10 pm to 7 am, with a break at 2 am.

[1]Control Objectives for Information and related Technology
[2]Information Technology Infrastructure Library

Due to the complexity of software project deployments, change deployers only applies the night and a special shift:

- Weekend, Saturdays and Sundays, 7 am to 10 pm, with one-hour breaks at noon and 5 pm.

The objectives to be optimized are defined by the following Key Performance Indicators (KPIs):

- KPI 1: Percentage of changes completed successfully within the time estimates, as a measure of the time factor. The higher the value of this KPI, the better for the process efficiency.
- KPI2: Actual change duration/estimated change duration, as a measure of internal quality. The lower the value of this KPI, the better for the process efficiency.
- KPI 3: Overall number of resources utilized, as a measure of cost. The lower the value of this KPI, the better for the process efficiency.

The multi-objective algorithms NSGA-II [38] and SPEA2 [39] were used in the experiment defined in jMetal [40] to perform the optimization process.

In each simulation iteration, the experiment defined in jMetal provides a configuration for the IT Change Process Efficiency problem to Anylogic™, which runs the simulation and returns the KPIs to jMetal. The process continues until jMetal reaches the stopping criteria. Figure 3.1 shows the process graphically.

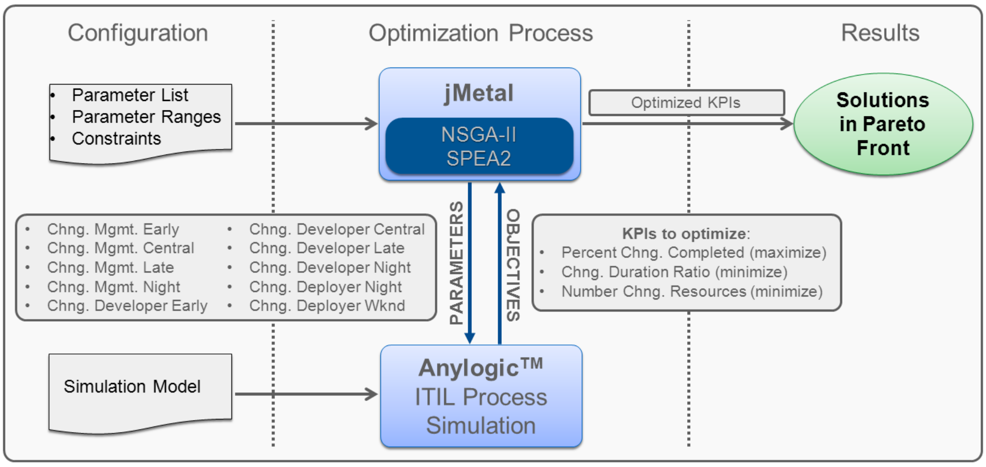


Figure 3.1: Interaction between jMetal and Anylogic™

3.3 Optimization applied to antenna design

Antenna design is another field of engineering where simulation plays a key role in reducing manufacturing time and costs. Computer simulation allows to model an antenna and evaluate its behaviour before manufacturing. Despite the advantage that simulation brings, in some cases, such as the design of satellite antennas, it may not be enough.

In a satellite, besides the specific electromagnetic requirements, the size of the antenna is critical, due to 1) the limited space available and 2) the weight limitation of each component installed on the satellite.

Gonzalez *et al.* [41] designed a compact dual-band equatorial helical antenna for TTC. In order to find the specific parameters that would meet all the restrictions, they used the MONURBS simulator together with the metaheuristic algorithm *Gradient Descent*. It was necessary over two calendar months of running thousands of simulations to find a valid solution. In the paper presented in Chapter 6, it is shown a solution to the problem using the MONURBS simulator together with the multi-objective algorithms NSGA-II [38] and SPEA2 [39], implemented in jMetal [40]. the best solution obtained reduced the size of the antenna by almost a third and it took three weeks to obtain.

The geometry of the helical antenna is defined by the following parameters (see Figure 3.2):

- Bottom radius (r).
- Top radius (R).
- Height (h).
- Number of turns of the helix (t).

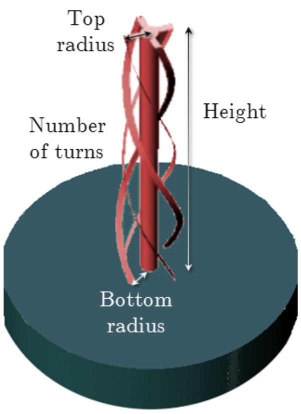


Figure 3.2: Geometrical parameters of a helical antenna

The electromagnetic objectives to be met were the following:

- Maximize the RHCP gain for 1.81-GHz frequency.
- Minimize the cross-polar polarization level for 1.81 GHz frequency in the range between 70º and 110º. In this range, gain must be above 0 dBi.
- Maximize the RHCP gain for 2.55-GHz frequency.

- Minimize the cross-polar polarization level for 2.55-GHz frequency in the range between 70º and 110º. In this range, gain must be above 0 dBi.

Before starting the optimization process, the geometric model of the antenna was defined in the MONURBS simulator and the optimization experiment in the jMetal tool. During the optimization process, from the jMetal experiment new values for the antenna geometry are provided and from the simulator the electromagnetic characteristics of the antenna are returned for the provided values. The process continues until a stop criterion is reached. Figure 3.3 illustrates this process.

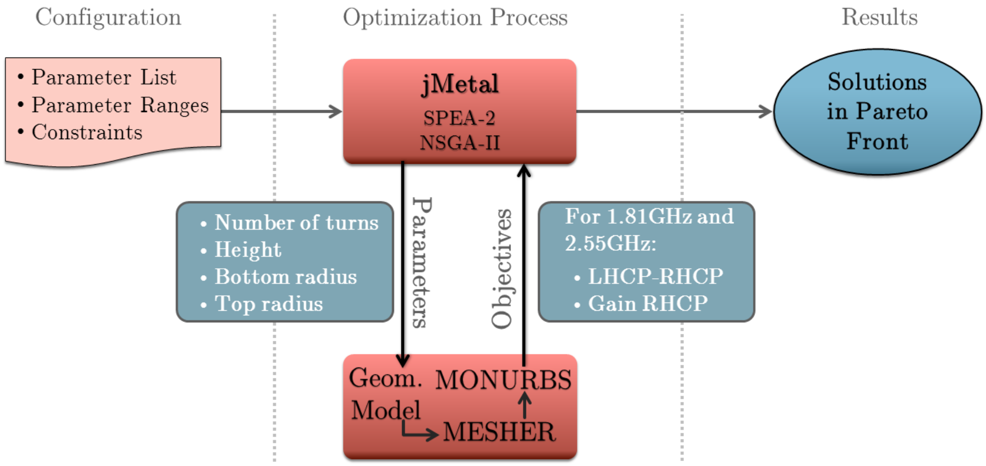


Figure 3.3: Interaction between jMetal and MONURBS

As it can be seen in Figures 3.1 and 3.3, despite being different problems, the simulation optimization process is similar.

3.4 Merge Non-Dominated Sorting

The Merge Non Dominated Sorting (MNDS) algorithm is briefly described here. This algorithm is a proposal to speed up the family of algorithms that use ranking to sort the population solutions. There are a multitude of algorithms dedicated to obtaining the Pareto Front. Some of the best known are FNDS [38], ENS [42], ENS-NDT [43] or BOS [44]. All these algorithms have a computational complexity, in the worst case, of $O(MN^2)$ so the differences between them are evaluated at run time. Table 3.1 shows the computational complexity of some representative algorithms.

Given a population of N solutions, and M objectives, MNDS ranks the solutions, in an iterative way, by each of the objectives. In each iteration, MNDS calculates the set of solutions that dominate each solution (dominance set), and finally, once the dominance set of all the solutions has been calculated, it obtains the ranking of each solution. Note that the dominance set represents the weakness of the solution, as opposed to the concept of strength, used by SPEA2.

Table 3.1: Complexity of non-dominated sorting algorithms representative of the state-of-the-art.

Algorithm	Complexity		
	Best Case	Worst Case	Space
FNDS [38]	MN^2	MN^2	N^2
Dominance Tree [45]	$MNlogN$	MN^2	M
Deductive Sort [46]	$MN\sqrt{N}$	MN^2	N
Corner Sort [47]	$MN\sqrt{N}$	MN^2	N
ENS-SS [42]	$MN\sqrt{N}$	MN^2	1
ENS-BS [42]	$MNlogN$	MN^2	1
ENS-NDT [43]	$MNlogN$ if $M > logN$ $Nlog^2N$	MN^2	$NlogN$
M-Front [48]	MN	MN^2	MN^2
DDA-NS [49]	MN^2	MN^2	N^2
HNDS [50]	$MN\sqrt{N}$	MN^2	N
BOS [44]	$MNlogN$	MN^2	N^2
MNDS	$NlogN$	MN^2	N^2

Despite presenting a complexity, in the worst case, of $O(MN^2)$, MNDS reduces its complexity as the dominance among the solutions decreases, reaching a complexity of $O(NlogN)$ in the best case. If MNDS is used by a multi-objective algorithm to obtain the population ranking, as the multi-objective algorithm converges towards fronts closer to the Pareto front, the dominance will be reduced and MNDS will tend to reach its best complexity.

This is explained in detail in Chapter 7 where the paper is enclosed.

Chapter 4

General Conclusions, Future Work and Published Articles

This chapter summarises the research carried out in the previous chapters, analyses the research contributions and offers suggestions for future research.

4.1 Conclusions

The objectives set at the beginning of the thesis were as follows:

Objective 1: Study the different multi-objective optimization algorithms.

Objective 2: Study of simulation optimization in software development projects.

Objective 3: Study of multi-objective approaches in antenna design using simulation approaches.

These objectives have been successfully achieved with the publications. Objective 1 has been addressed with in all publications. Objectives 2 and 3 have been achieved with their respective publications.

In this thesis, solutions to two very different real problems were presented, both these problems needed simulators and were previously solved using simulation techniques together with generic metaheuristic algorithms. Anylogic™applies a single objective optimization algorithm, while MONURBS used the Gradient Descent algorithm. The application of multi-objective algorithms to these same problems was shown to be more efficient, both in terms of computational cost and in the solutions provided.

In the case of the simulation applied to IT Service Management, the results using multi-objective algorithms were much better than the ones obtained with the optimizer included with the Anylogic™ tool. Figure 4.1 shows some of the best non-dominated solutions of the Pareto front using jMetal and the single solution (grey square) provided

Table 4.1: Parameters of the best solutions found by Gradient Descent, NSGA-II and SPEA2 algorithms.

Algorithm	Turns	Bottom radius (cm)	Top radius (cm)	Height (cm)	Volume (cm^3)
Gradient Descent	0.831	1.945	1.022	13.8	98.49
NSGA-II	0.777	1.929	1.145	14.072	106.676
SPEA-2	0.84	1.032	0.753	13.593	**34.281**

by Anylogic™. The multi-objective evolutionary algorithms could find better solutions than the ones offered by Anylogic™ in all objectives. Since this proposal benefits from the advantages of the multi-objective optimization approach applied to the results of simulation models, the range of the solutions provided in the Pareto front can help IT managers understand the effect of different management strategies and improve their decision-making towards more efficient processes.

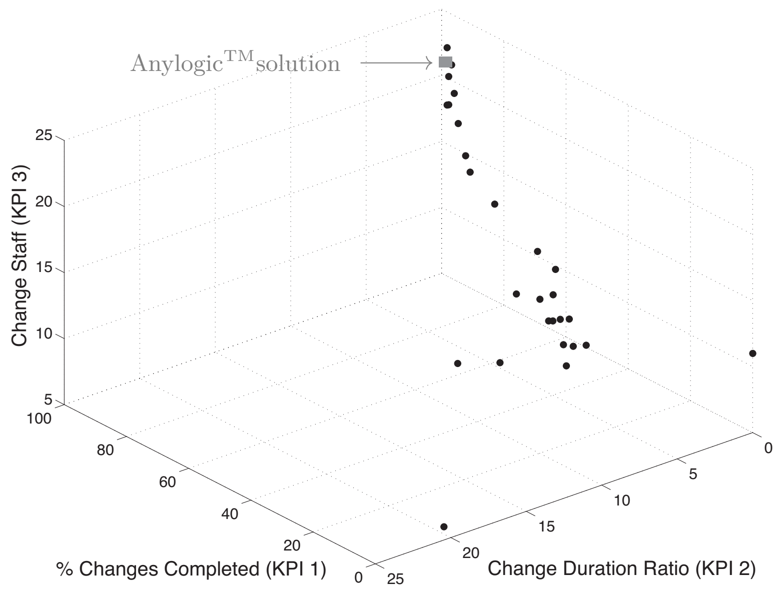


Figure 4.1: Pareto front using jMetal and solution found by Anylogic™optimizer.

In the field of antenna design, a significant time cost reduction was achieved. Furthermore, the simulation–optimization approach allow us to obtain multiple correct solutions that provide some flexibility and can help the decision maker to choose the final design of the antenna. The results obtained with the different algorithms can be seen in the Table 4.1. The experiment using Gradient Descent took two calendar months, while the NSGA-II and SPEA2 experiments took only three weeks. Note that the result obtained by NSGA-II is very similar to that achieved by Gradient Descent, while SPEA2 was able to reduce the volume of the antenna to one third of that obtained by Gradient Descent.

Having more solutions, with different dimensions but all optimal from the radiation point of view, offers more possibilities for the manufacturing not only for the antenna but the rest of elements that are coupled closely to it.

Figure 4.2 shows the parameters of one of the best solutions obtained during the optimization process.

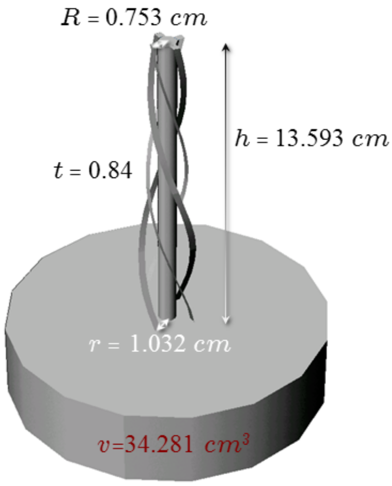


Figure 4.2: Graphical representation of parameters of the best solution, obtained by SPEA2.

As it was observed in these experiments, regardless of the nature of the problem, multi-objective optimization applied to simulation can bring benefits in solving complex problems.

In addition, there were theoretical advances. More concretely, a new sorting algorithm, Merge Non-Dominated Sorting (MNDS), strongly outperforms the current state-of-the-art algorithms in terms of running time and the number of comparisons carried out. Therefore, MOEAs based on the Pareto ranking can benefit greatly from significant time reductions, particularly in the case of using large populations and solving many-objective problems. Therefore, MNDS allows us to relax the stop conditions of the MOEAs algorithms based on Pareto ranking and execute more iterations because when there is no dominance, its computational cost is reduced to $O(MNlogN)$.

4.2 Future Work

The following lines of work where identified as future research in the area of simulation optimization:

- Use other multi-objective algorithms capable of handling the constrains to compare and adapt them to the difficulty of these problem.
- Explore many-objective algorithms to be capable of increasing the number of objectives in order to analyse more complex simulations.
- As the number of variables and solutions in the Pareto can be large, it is necessary to explore visualization and clustering techniques to present the results, as well as defining new multi-criteria decision making methods to choose the solutions to adopt among those in the Pareto front.

- Explore parallel versions of the algorithms, or possible parallelization techniques as usually simulations require very high computational costs.
- Study why SPEA2 has obtained slightly better results than NSGA-II in all the experiments.
- Study the application of memetic algorithms to use cases to achieve even better results.

From a more theoretical point of view and in relation to MNDS, it is possible to improve the way the rankings are calculated by MNDS by the use of different search methods and data structures. For example, using the *Timsort*[1] [51] algorithm instead of *merge sort*, and the sequential search used in finding the ranking of each solution could be replaced by a binary search or a *k-d tree*.

Another line of research would be to adapt the MNDS algorithm so that it can be used with steady-state Evolutionary Algorithms.

4.3 Published Works

Selected Refereed Publications

During the process of elaboration of this thesis, several articles have been published in Journal Citation Reports (JCR) indexed journals. This section cites the JCR articles that appear in the Part II of this thesis as well as other publications.

Ruiz, M., **Moreno, J.** Dorronsoro, B., Rodriguez, D., Using Simulation-based Optimization in the Context of IT Service Management Change Process, *Decision Support Systems*, Vol. 112, August 2018, 35-47, DOI: 10.1016/j.dss.2018.06.004

This journal has the following metrics (year, quartil, category, position/number of journals included in the category):

- 2018 Q1 - Computer Science, Information Systems - 29/155
- 2018 Q1 - Computer Science, Artificial Intelligence - 30/134
- 2018 Q1 - Operations research and Management Science - 12/84
- Impact factor 2018: 3.847

Moreno, J., Gonzalez, D., Rodriguez, D., Design of a TTC Antenna using Simulation and Multi-objective Evolutionary Algorithms, *IEEE Aerospace & Electronics Magazine*, 34(7), 18-31, 2019, DOI: 10.1109/MAES.2019.2916535

[1] https://hal-upec-upem.archives-ouvertes.fr/hal-01212839v2/document

This journal has the following metrics (year, quartile, category, quartile rank/number of journals included in the category):

- 2018 - Q1 - Engineering Aerospace - 6/31
- 2018 - Q3 - Engineering, electrical & electronic - 135/266
- Impact factor: 2.113

Moreno, J., Rodriguez, D., Nebro, A.J., Lozano, J.A., Merge Non-Dominated Sorting Algorithm for Many-Objective Optimization, *IEEE Transactions on Cybernetics*, Accepted, 2020, DOI: 10.1109/10.1109/TCYB.2020.2968301.
(Available on-line)

This journal has the following metrics (year, quartil, category, position/number of journals included in the category):

- 2018 - Q1- Computer Science, artificial intelligence - 4/134
- 2018 - Q1- Computer Science, Cyberntics - 1/23
- 2018 - Q1- Computer Science, Cyberntics - 1/62
- Impact factor: 9.631

Conference Refereed Publications

Moreno, J., Gonzalez, I., Rodriguez, D., Using Simulation and the NSGA-II Evolutionary Multi-Objective Algorithm in the Design of a Compact Dual-band Equatorial Helix Antenna, In Proceedings of 6th International Conference on Space Mission Challenges for Information Technology (SMC-IT 2017), 27-29 September 2017, Alcala de Henares, Madrid, Spain, DOI: 10.1109/SMC-IT.2017.17

Non-refereed Publications

Moreno, J., Rodriguez, D., Nebro, A.J., Lozano, J.A., Merge Non-Dominated Sorting Algorithm for Many-Objective Optimization, arXiv:1809.06106, 2018
https://arxiv.org/abs/1809.06106

Part II

Selected Refereed Publications

Chapter 5

Using Simulation-based Optimization in the Context of IT Service Management Change Process

Today's IT systems and IT processes must be ready to handle change in an efficient and responsive manner to allow businesses to both evolve and adapt to a changing world. In this paper we describe an approach that consists of using simulation based multi-objective optimization to select optimal ITIL change management process strategies that help IT managers achieve process efficiency as a Critical Success Factor (CSF). A multi-method simulation model, which is based on agent-based and discrete-event simulation paradigms, has been built to simulate the whole process life-cycle, since the change initiation until its closure. As most engineering problems,assuring an efficient delivery of the change management process requires optimizing simultaneously the corresponding Key Performance Indicators (KPIs) in which the process-efficiency CSF can be rolled down. In this paper, we show the results of applying two well-known Multi-Objective Evolutionary Algorithms, namely NSGA-II and SPEA2, to obtain a set of optimal solutions for the KPIs associated with delivering process efficiency as a CSF. We also compare the results obtained with the output from the single-objective optimization algorithm provided by the simulation tool. The experimental work included shows how the approach can provide the IT manager with a wide range of high quality solutions to support them in their decision-making towards CSF achievement.

Decision Support Systems 112 (2018) 35–47

Contents lists available at ScienceDirect

Decision Support Systems

journal homepage: www.elsevier.com/locate/dss

Using simulation-based optimization in the context of IT service management change process

Mercedes Ruiz[a,*], Javier Moreno[b], Bernabé Dorronsoro[a], Daniel Rodriguez[b]

[a] Dept. of Computer Science and Engineering, University of Cádiz, Puerto Real 11519, Cádiz, Spain
[b] Dept. of Computer Science, University of Alcalá, Alcalá de Henares 28871, Spain

ARTICLE INFO

Keywords:
Service management
Change process
ITIL
Multi-objective optimization
Evolutionary algorithms
Simulation

ABSTRACT

Today's IT systems and IT processes must be ready to handle change in an efficient and responsive manner to allow businesses to both evolve and adapt to a changing world. In this paper we describe an approach that consists of using *simulation based multi-objective optimization* to select optimal ITIL change management process strategies that help IT managers achieve process efficiency as a Critical Success Factor (CSF). A multi-method simulation model, which is based on agent-based and discrete-event simulation paradigms, has been built to simulate the whole process lifecycle, since the change initiation until its closure. As most engineering problems, assuring an efficient delivery of the *change management process* requires optimizing simultaneously the corresponding Key Performance Indicators (KPIs) in which the process-efficiency CSF can be rolled down. In this paper, we show the results of applying two well-known Multi-Objective Evolutionary Algorithms, namely NSGA-II and SPEA2, to obtain a set of optimal solutions for the KPIs associated with delivering process efficiency as a CSF. We also compare the results obtained with the output from the single-objective optimization algorithm provided by the simulation tool. The experimental work included shows how the approach can provide the IT manager with a wide range of high quality solutions to support them in their decision-making towards CSF achievement.

1. Introduction

In our current digital world, Information Technology (IT) plays a crucial role to help organizations succeed in delivering value to their customers. In response to this growing dependence on technology, today's IT organizations need to be both effective and efficient in transforming resources into valuable services. IT Service Management (ITSM) is defined as the strategic approach to the design, delivery, management, and improvement of the way in which IT is used within an organization.

Since technology and the IT infrastructure are constantly changing and advancing in today's world, organizations such as IT service providers must be ready and able to adapt themselves to evolving conditions if they want to remain competitive and innovative. To this end, the different ITSM guides and standards provide guidance about how to manage change in a productive way, ensuring that new or modified IT services evolve along their lifecycle under a controlled and well-organized manner that keeps these services compliant with the business requirements.

There are different ITSM best-practice guides and standards that help organizations implement an ITSM strategy. A 2017 report, based on a survey of 261 IT leaders in large organizations around the world [18], shows that the top five most used ITSM approaches are: Information Technology Infrastructure Library (ITIL) (47%) [3], Business Process Framework (eTOM) (36%) [48], Control Objectives for Information and Related Technologies (COBIT) (36%) [20], Microsoft Operations Framework (MOF) (34%) [33] and ISO/IEC 20000 (29%) [21].

Among all the existing ITSM guides, we focus in this work on the ITIL proposal, the most widely used one. In fact, it is regarded by many as the de-facto standard for ITSM and its terminology is widely understood and used. ITIL follows a process-driven approach that is grounded in business experience. It offers a set of best practices for IT service management and delivery under an ITSM approach. The ITIL framework comprises the definition of 26 process areas used to describe how IT services evolve through the five main stages of their lifecycle. ITIL consists of five core publications, namely, (i) Service Strategy, (ii) Service Design, (iii) Service Transition, (iv) Service Operation, and (v) Continual Service Improvement, each one dedicated to describe in depth each of the five stages of an IT service lifecycle.

* Corresponding author.
E-mail addresses: mercedes.ruiz@uca.es (M. Ruiz), javier.morenom@edu.uah.es (J. Moreno), bernabe.dorronsoro@uca.es (B. Dorronsoro), daniel.rodriguezg@uah.es (D. Rodriguez).

https://doi.org/10.1016/j.dss.2018.06.004
Received 20 December 2017; Received in revised form 14 June 2018; Accepted 16 June 2018
Available online 27 June 2018
0167-9236/

The ITIL service transition stage is precisely aimed at helping organizations plan and manage the change of state of services in their lifecycle, in a controlled manner, minimizing the risks and ensuring that the services meet the customer's expectations and the business requirements [36]. One of the critical processes within service transition is change management. According to ITIL, “the purpose of the change management process is to control the lifecycle of all changes, enabling beneficial changes to be made with minimum disruption to IT services” [36]. The scope of change management covers changes to service assets and configuration items across the whole service lifecycle. The process addresses all changes at all levels: strategic, tactical and operational.

Today's organizations demand service changes to be performed in less and less time without compromising efficiency. Thus, efficient change management is essential, since the consequences associated with process inefficiency reach the customer. Low quality service changes often lead to new incidents that result in customer dissatisfaction.

Change management needs the involvement of several types of resources, with their corresponding costs. Among them, human resources are crucial for the outcomes of the process. These resources are commonly structured in IT support groups. Each one plays a different role in the process such as change initiators, developers, deployers, and can be part of the IT staff or work for third-party organizations. In practice, all these factors are combined to make the entire change process highly unpredictable, where outcomes depend on complex interactions between different changes, people and groups, each of which, have their own priorities and objectives. Although ITIL and other ITSM frameworks provide important guidance, managing change in real organizations is a very complex process. Also, change management often requires optimizing several objectives simultaneously, such as maximizing the percentage of changes completed on time and minimizing the change duration ratio and the number of resources used.

Therefore, change management is a complex problem that organizations need to handle in an effective way when coping with service transition processes. The goal is to perform the changes with the lowest impact on the Quality of Experience (QoE) perceived (as service disruption, incidents with other live services, or any other issue that might affect customers' satisfaction), but also with the lowest effort for the organization (as minimizing the cost of the change or the resources required). Often, the interests of organization and customers are in conflict, e.g., reducing the time required for the change management process leads to a cost rise for the organization. Optimization techniques can be very valuable tools used to both find high quality solutions to support decision-making and to ensure an efficient change management delivery.

Particularly, multi-objective optimization algorithms can effectively handle the optimization of different conflicting objectives simultaneously, offering a wide and diverse range of trade-off solutions to the problem, helping the IT manager to make the most appropriate decisions. In order to guide the search towards high-quality solutions for such a complex problem involving a large number of processes, interactions among them, and uncertainties, these algorithms must rely on accurate simulations. This approach is explicitly proposed in frameworks to improve decision-making in the ITSM scope, such as the Sim4ITSM framework, which includes activities of simulation optimization in the experimentation phase of the method [39].

The main contribution of this work is a novel application of the multi-objective simulation optimization approach for the IT change management process problem aimed at supporting effective decision making. To this end, we introduce a simulation approach that relies on both the agent-based and discrete event simulation paradigms to model the ITIL change process. We define the problem of optimizing process efficiency according to three main goals: 1) the size of the staff to perform the changes, 2) the change duration ratio, and 3) the percentage of changes completed, which are optimized simultaneously. The problem is tackled with two well-known Multi-Objective Evolutionary Algorithms (MOEAs), NSGA-II and SPEA2. The MOEAs outperform the reference result provided by the single-objective optimization obtained from the simulation software used to build the simulation model. This research represents pioneering work related to the use of multi-objective approaches in simulation-based optimization in the context of ITSM.

The structure of the paper is organized as follows. Section 2 summarizes the works related to our proposal. Section 3 describes the simulation model built for the ITIL change process. Section 4 introduces the MOEAs used in this study and the coupling structure followed to integrate the simulation software with the multi-objective optimization framework. Section 5 describes the experimental work performed. Finally, Section 6 outlines the conclusions and our further work.

2. Related work

In this section, we cover the most relevant works related to our study. Section 2.1 addresses contributions focusing on the design of ITSM simulators. We present the results of a systematic search in the literature of simulation-based optimization in the context of ITSM in Section 2.2.

2.1. Simulation in ITSM

Simulators have been demonstrated to be very useful tools for decision support systems to help decision makers in their activity. We can find many examples in the literature, with different fields of application as logistics [17], planning [41], economics [52,8], or supply chains [19], among many others. The use of simulation modeling to address ITSM problems has been an active topic of research in the last years. The most relevant works in the scope of the service change management process are reviewed next, and we refer the reader to the work by Orta et al. [39] for a more comprehensive literature review.

In [32], the authors describe a System Dynamics model to address the problem of low-performing IT operation by adopting and sustaining IT change and access controls. The simulation model built helped to validate *“underlying observations that change and access controls simultaneously reduce the security risk and increase the efficiency and effectiveness of IT management and operations”*.

Planning and scheduling of changes were identified among the main challenges in IT change management according to the results of a survey carried out by Hewlett Packard in 2006 [40]. The topic of the efficient management of a set of application changes under possible scheduling conflicts was addressed by Luo et al. [31]. The authors use graph and queue modeling to simulate different scheduling heuristics and find their impact on the change completion time and the change capacity of the system. The work finds the limit values for the degree of the scheduling conflicts and the cross-training of executing personnel that help keep the average change delay in the lowest level. In addition to an efficient scheduling of changes, it is also very important to estimate the business impact of operational risk resulting from changes. One proposal that quantifies this impact in terms of financial loss was described by Setzer et al. [42]. In this work, a probabilistic model for analyzing the business impact of changes in a network of services is introduced together with a decision model for service changes scheduling with the aim of reducing the total expected change-related costs. The proposal is then evaluated by using discrete event simulations of different scenarios.

Silva and Yaix [44] propose process simulation as a key element to guide the CIO and CEO in strategic business and IT alignment. They show the benefits of their proposal by simulating the incident and problem management process from the COBIT framework (which are sub-processes of the change management process in the ITIL framework). In this case, the simulation model is built using ADOIT®, which uses its own modeling language and provides basic simulation analysis tools, such as cycle-time and resource-utilization reports.

Simulation has also been used as a means to generate data that help validate conceptual assumptions. One example of this kind of application was reported by Cordeiro et al. [11]. In this work, simulation is used to generate the change logs that are consumed during the mining process whose conceptual and technical feasibility is being proved. Yang et al. [51] is another example of the use of simulated data to evaluate a proposal. In this case, the term *simulation* is used with its most general meaning, that is, the data used to validate "a patch management framework based on SLA-driven patch applicability analysis" is not real but mocked, i.e., simulated. Even though these works are in the context of improving some aspects of the change management process, their main contributions are not aimed at building simulation models. They make use of the term "simulation" to denote that to test their proposals they are using data that imitates real-life ones. Therefore, they differentiate from our work since in our case, the simulation model is the core element of our contribution aimed to design and test effective process improvement initiatives.

To the best of our knowledge, Thanheiser et al. [47] is the only existing work describing the architecture of an agent-based simulation model developed to assess an IT service architecture with respect to service availability and service level management. Our research shares with Thanheiser et al. [47] the use of the agent-based simulation to address problems in the IT service management landscape. However, while Thanheiser et al. 's work aims at building a simulator to help in the "design-time assessment of an IT service architecture", our work is focused on optimizing the execution of one key IT service management process: the *service change management*. Besides, while their work is focused on providing IT management with an assessment tool for particular service-oriented architecture implementations using agent-based simulation, our work is aimed at helping IT management to improve their decision making by optimizing the simulation of the IT management processes, the IT service change management process being the focus of this work.

2.2. Simulation optimization in ITSM

Simulation models allow the user to evaluate the outcomes of different process configurations and can help them find the combination of input values that lead to optimal process performance. When the number of different process configurations the user is interested in evaluating is very large, the number of alternative configurations that need to be simulated and compared grows exponentially. In these cases, it is helpful to integrate simulation with optimization techniques [29]. *Simulation optimization* can be defined as the process of finding the best values for a number of decision variables of a system, where the performance is evaluated based on the output of a simulator that models the system [37]. Simulation optimization is not a novel concept. Metaheuristics addressing problems that involve simulation are known as *simheuristics* [9,25]. They have been widely used to deal with problems that require reproducing the behavior of real systems, where uncertainties can be present in different ways. Simheuristics have also been applied with multi-objective optimization [49]. Some application examples include problems related to mobile networks [15], production planning [35], medicine [30], or water reservoir hedging [45], among many others.

In order to find similar research initiatives aimed at using simulation optimization in the scope of ITIL, and more specifically, in the scope of the change management process, we performed a methodical search of the literature. Our aim is not to perform a systematic literature review study, but to apply a rigorous method to our searches so that the relevant related works can be identified. A set of different categories and keywords, together with their respective synonyms, acronyms, and alternative spellings was designed as shown in Table 1 with the keywords used for the searching process. When adding the keyword *multi-objective*, there were no results.

Table 1
Keywords used for the search.

Category	Keywords
Information Technology Service Management	ITSM, ITIL
Simulation	Simulation, simulating, simulate
Optimization	Optimization, optimization
Multi-objective	Multi-objective

The digital libraries (DL) where the searches were performed were: IEEE Xplore, ACM Digital Library, ScienceDirect, ISI Web of Knowledge, EI Compendex, SpringerLink, Inspec, Scopus and Kluwer.

Only 14 papers where found by the searches conducted. In most of the papers retrieved, the term *optimization* is used in its most general meaning, far from any form of mathematical or meta-heuristic optimization. After removing three papers that were duplicated, the final number of papers retrieved in the search was 11. A first analysis of the papers retrieved, led us to discard 4 papers based on the fact that they were not properly related to the field of simulation optimization within the IT service management scope. This is the case of Xu et al. [50], which describes the current state and development plans for research and education on Services Sciences, Management and Engineering (SSME) in the Harbin Institute of Technology (HIT) or Amin et al. [1], which introduces the ProCEM® method (Process-Centric Enterprise Modeling & Management) that follows the process-centric application systems development style and integrates different services concerned with organization structure and IT applications to support, execute, or even automate the processes. Simulation and Optimization are among the components of this framework. They are also mentioned as part of the method described by Shrinivasan et al. [43] that helps validate the intuitive direction and the polarity of a causal relationship among IT service Key Performance Indicators (KPIs) and estimate the values of the KPIs. Finally, simulation is used as a means to assess a distributed IT management framework as described by Jiang [24].

The exclusion of these non-related works left us with only 7 papers that could be analyzed as truly related to the aim of our work, which is to find existing works that apply simulation-based optimization in the scope of IT service management, and more specifically the ITIL change process.

Most of the regular papers retrieved propose the use of simulation as a means to help decision-making or mitigate risks. The trade-off between energy-efficiency and resilience in communication networks is addressed by Cholda and Jaglarz [10] as a risk mitigation problem. In their work, the authors propose a method to find an optimal solution that mitigates risk by iteratively combining simulations and linear programming. The method is based on repeating a simplified risk management cycle, where the optimization represents the risk control phase and the simulation represents the risk assessment. Risk mitigation and management is among the challenges of IT processes and among the priorities of IT governance activities carried out in organizations. Krey et al. [27] deal with the importance of IT governance in the Swiss healthcare sector. They conclude that IT governance can help optimize the business processes in the medical as well as non-medical areas of a hospital. This work highlights the importance of having specific process models for the health sector to optimize hospital strategies, making adequate business decisions and minimize risks. It does not propose any method of integrating simulation and optimization techniques in the IT processes of the healthcare sector and concludes with the necessity of carrying out a survey to find out the current status and spread of IT governance in the mentioned sector.

The incident management process has been the process that has attracted most of the research done applying simulation techniques. According to ITIL [22], incident management is *"the process through which IT support organizations manage to restore normal service operation after a service disruption"*. Therefore, IT organizations need to measure

Table 2
Summary of relevant works related to ITIL process optimization based on simulation.

Ref.	Simulation scope	Method	Simulation-based optimization
[10]	Risk mitigation in resilient green communication networks	Networks represented as undirected graphs and linear programming to optimize recovery strategies risk and cost	Bi-objective linear programming
[27]	IT process optimization for hospitals	No method integrating simulation & optimization is proposed	No
[4]	Performance analysis and optimization of the incident management process	Discrete-event simulation	No
[5]	Business impact analysis and improvement of the incident management process	Discrete-event simulation	Limited to a number of ad hoc strategies
[6]	Modeling of a single IT support group	Discrete-event simulation with multiple-priority queues	No
[39], [38]	ITIL Incident management process performance	Discrete-event simulation	Single objective
Our work	ITIL Change management process	Discrete-event and agent-based simulation	Multi-objective metaheuristics

the efficiency and effectiveness of their incident management strategies. Simulation can help design and assess the outcomes of different strategies so that the organization can implement the ones that provide the best process results.

SYMIAN is a simulation-based decision support tool that helps to analyze and optimize the incident management process of IT support organizations [4]. SYMIAN is based on a discrete-event simulation model and therefore models the IT support organization as a queuing system. In their work, Bartolini and his colleagues introduce the tool and provide an experimental evaluation of SYMIAN in a fictitious situation. Although the experimental results are intended to show how the tool optimizes the process outputs, there is no evidence of the application optimization techniques, and the results come from what seems to be scenario comparison rather than proper simulation optimization. In a later work [5], the same authors introduce HANNIBAL, a discrete-event simulation tool that works together with SYMIAN for business impact analysis and improvement of the incident management process. We share a similar motivation with this work since both initiatives aim to find the values for process options that optimize the process outcomes. However, the simulation optimization methods used are substantially different. In [5], the user defines a set of business strategies and is interested in finding the one that best aligns with a business objective. A business objective is a function that can aggregate different objectives into a single function using weights that represent the importance that the user gives to each objective. The tool performs a sequence of activities that starts with a complete simulation for each business strategy and calculates its cost. After that, the optimization process consists of selecting the business strategy that has the best alignment to the business objective. In our approach, the user first sets the acceptable range for the decision variables and their constraints and then selects the outputs whose values are to be either maximized or minimized together. Then, it is the multi-objective evolutionary algorithm, and not the user, who defines the best strategy, relying on simulations to obtain the process performance of all tentative solutions generated (i.e., process configurations). The process performance obtained by simulation is used to calculate the fitness of the tentative solution within the solution domain. Therefore, our approach does not evaluate several ad hoc strategies as in [5], but it automatically generates and evaluates several tens of thousands of them.

Bartolini et al. [6] deepen in the simulation modeling of a single IT support group. In this case, the support group is modeled by using a discrete-event mode with multiple-priority queues. This component is then integrated into SYMIAN simulation core to allow the users to simulate *what-if* scenarios. In addition, the tool counts with an optimizer component materialized as an R application. The optimizer allows to find the values for the model parameters that *"enable the most accurate reenactment of a real life support group"*. To do that, the optimizer relies on a non-linear optimization algorithm based on the Limited memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm [34] to minimize the Wilcoxon distance [34] between the simulation outputs and the real values collected from historic data. In this work, optimization is only used to guide the inference of model parameters that make the simulation outputs reproduce closely the historic data collected, i.e. model calibration. The application of simulation-based optimization to improve process performance, which is our goal, is not among the aims of this work.

The remaining related works in the scope of the IT service management simulation optimization have been proposed by Orta et al. [39], and Orta and Ruiz [38]. In their first work [39], the authors offer a comprehensive review of the research papers that apply simulation modeling in the ITSM context, propose a decision-making framework based on simulation modeling to improve ITSM and illustrate the usefulness of this framework with two application cases including model simulations and optimization experiments to determine the optimal process configuration. In a following contribution, the authors propose a business-process and simulation-based method to support ITIL implementation and present the results of an application case in the context of the incident management process of a real company. In both works, the simulation optimization experiments were set in the scope of the ITIL incident management process and made use of the optimization engine implemented in OptQuest® to find the best process configuration that maximizes the number of incidents solved. These works serve as clear examples of single-objective simulation optimization in the ITSM domain.

After the analysis of the previous works describing the use of simulation optimization in the scope of IT service management, we can conclude that this is a topic where very little research effort has been applied (see Table 2). In the small number of cases describing an application of simulation optimization, this has been of the single-objective category, despite the multi-objective nature of the problem: in IT service management, it is usual that conflicting objectives (i.e., the KPIs) interact with each other in nonlinear ways. As a result, a multi-objective optimization approach seems to be more adequate in this domain. To the best of our knowledge, which is based on the findings of the systematic searches performed, our proposal is the first one that aims at using simulation-based multi-objective optimization to improve decision making in ITSM domain and, more specifically, in the scope of the change management process.

3. Simulation model for the ITIL change management process

This section introduces briefly the change management process abstraction and describes the simulation model built following Law's methodology [28]. The layout of the description is partly based on Kellner's proposal for describing simulation models [26].

3.1. Process abstraction

This section provides a short introduction to the ITIL change management process with a focus on those aspects of the process that are especially relevant to the purpose of the model. A comprehensive description of the process can be found in the ITIL Service Transition

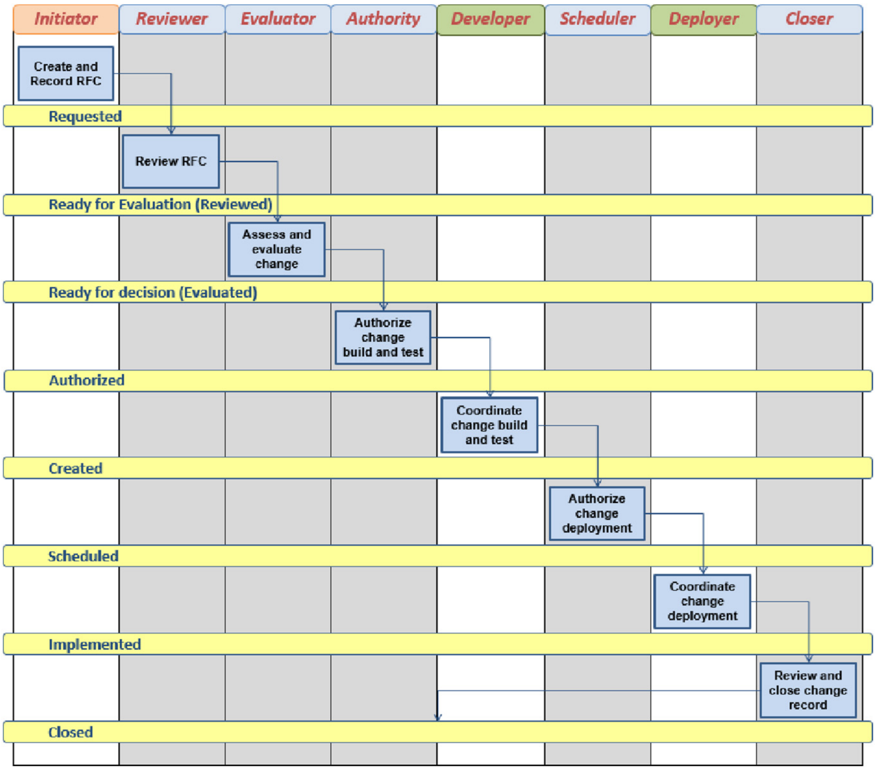


Fig. 1. ITIL change management process.

Guide [36].

The ITIL definition of change is "the addition, modification or removal of anything that could have an effect on IT services" [36]. Changes are requested by a formal proposal called RFC (Request for Change). A change can be requested by an individual or a business unit. ITIL describes three types of service change: a) Standard, a relatively common low-risk and pre-authorized change, b) Emergency, an urgent change to solve a critical situation, and c) Normal, a change that is neither standard or urgent. Additionally, changes can be categorized depending on their cost and risk into major, significant and minor changes. This classification helps in identifying the suitable authority level to authorize and manage them. Once a RFC is produced, a change record is created. A change record is a record that contains all the information of the lifecycle of a particular change.

At a very high level, the ITIL change management process can be described as the processing and managing of change records from an open state to a closed state performed by different resources. Fig. 1 shows the typical activities[1] needed to manage an individual change, the roles involved in each one and the different states of the change lifecycle. These activities are: (a) Create and record the RFC; (b) Review the RFC, so that incomplete or wrongly routed changes can be consequently addressed; (c) Assess and evaluate the change, so that the business justification, impact, cost, etc. and relevant areas of interest and authority are evaluated; (d) Authorize the change, so that the authorization/rejection for the change can be communicated, especially to the change initiator; (e) Plan updates, so that the scheduling information for task development and testing is completed; (f) Coordinate change implementation, so that the activities needed to implement the change are properly performed; and (g) Review and close change, so that the change and its documentation can be reviewed, the lessons learned gathered and the change can be formally closed.

[1] In order to provide an overall view of the process, the figure has been simplified. The implementation of the model takes into account the feedback loops in the management of each change.

The coordinator of the change team and the main responsible for the process is the change manager who needs to ensure that all the activities are undertaken, documented and reviewed in an appropriate manner. The decisions of the change manager are supported by the Change Advisory Board (CAB) that gives approval to the requested changes and assistance in the assessment and prioritization of changes.

3.2. Purpose and scope of the simulation model

The purpose of the simulation model is to help decision-making in the change process management of the service transition stage of the IT service lifecycle as defined by ITIL, with the aim of supporting decisions towards process efficiency. The model can be configured via a set of input parameters to present the operational characteristics of real-world change management processes and provide output information related to the process typical Key Performance Indicators (KPIs). Accordingly, the scope of the model is the whole of the process as described by ITIL: from change initiation to change closure.

3.3. Input parameters

The input parameters of the simulation model are used to tune the operation of the model to mimic that of the real-world change management process carried out in an organization. Different sets of parameter values enable the simulation of wide range of change processes using different scenarios. Since our intention is to make this simulation model highly configurable to simulate ITIL-based change management process from different organizations and be able to replicate multiple adjustments to these change processes, the model accepts a large number of input parameters (a total of 136, between scalar and vectorial inputs). These input parameters help the model user tune the model to each particular process implementation, adjusting the model's behavior to different situations and thus improving its flexibility. The model parameters have been selected by analyzing carefully the description of the IT service change process in the ITIL Service Transition guide [36] and our experience with simulation model building for ITIL implementations [39,38], with the aim of providing a sufficiently flexible and highly adaptable model assuring that every parameter has a real-world counterpart.

In order to design the scenario the user is interested in simulating, the input parameters need to receive values. These values can be obtained from different sources. The most realistic simulation outputs will be obtained when the model parameters are fed with real data that the organization has already collected. However, sometimes organizations do not count with either reliable data about the execution of their processes or the data available is not enough to provide all the input parameter values. In the absence of real data or in the case that the model is used to provide insights about a hypothetical situation, the users can provide subjective estimates for the model inputs. On the other hand, when real world data is available, running optimization experiments can help users to find the values of the input parameters that fit such data, following what is known as a calibration process. Practically, the input parameters used to set a simulation experiment with this model are configured in an Excel spreadsheet that is loaded into the model at run time. Because it is not feasible to describe each of them individually in this paper, we outline the different categories of inputs that the model accepts and provide some examples of parameters in each category.

- *Change types.* This group helps define the input stream of changes to be processed. Examples of inputs in this group are the change type, the change complexity factor and the range of configuration items affected by the change.
- *Processing durations.* The parameters grouped in this category help define the duration of each task of change processing, desegregated by change type and category. Providing an estimate for the duration of every task within a process execution implies a high level of uncertainty given the multiple factors of different nature that lead to a wide range of possible completion times. This is particularly true in tasks that can be performed within feedback loops, like the ones in the service change process, since the number of repetitions of the task and the completion time of each execution is unpredictable. The problem of uncertain task durations has been traditionally overcome in management by estimating the probability distribution of the duration times. The triangular distribution of three-point distribution is commonly used to solve this estimation problem since it is easy to use and requires only three estimates of minimum, most likely and maximum durations which should be within the estimating capability of the IT manager [12,29]. For this reason, tasks durations are modeled as a three-point estimation for the lower limit, upper limit and mode of a triangular distribution. At the beginning of a simulation run, the value for each task duration is stochastically calculated from its corresponding triangular distribution. The tasks included in the model are the ones represented in Fig. 1.
- *Change resources.* This group helps define the size and features of the resource pools available to process changes. Examples of inputs in this group are the number of change reviewers, the number of change developers, their processing efficiency, the number and timetable of each working shift and the resource allocation to working shifts. By processing efficiency, we mean how well the processing task is performed, i.e. does it introduce errors or delays into the process? This is directly related to the skill of the resources performing the change processing tasks, i.e. how well they do their jobs. The value is provided as the probability that the resource has of introducing errors or delays in the process, measured in percent.
- *Task parameters.* This group helps define the tasks features such as their priorities in the resource request queues. Examples of inputs in this group are the weighting factors applied to the priority of requests and to the priority of long running tasks, so that, during the simulation, resource monopolization can be avoided.
- *Process decisions.* This group of parameters help define the decisions made by the different roles with responsibility in the change management process, for instance, whether a change is authorized or rejected. Examples of inputs in this group are the statistical distributions that define the activation probability of each decision component coded in the model implementation.

There are other input parameters that help configure the general settings of the model such as the model time, and the randomness. Since these are not particular inputs of the change process management model, but general inputs applicable to every simulation model, they have not been described in this section.

3.4. Output variables

The output variables are the information elements needed to fulfill the purpose of the model. Change process success can be studied under different and complementary dimensions. First, the value dimension, which means that the process must assure the reduction of the negative impacts of change over the business. Second, the efficiency dimension, which means the process must be carried out in a timely and costly effective way. Third, the risk dimension, which means that changes must be handled in a risk-controlled way. Our simulation model is suitable to explore the efficiency dimension of the process as described above.

In order to provide meaningful outputs to help improve process efficiency, the outputs selected for this model are intended to serve as a basic for the calculation of the typical Key Performance Indicators (KPIs) suggested for the ITIL change management process [36]. It is important to notice that since this is a dynamic simulation model, the outputs do not only contain the values achieved at the end of the simulation, but are being updated dynamically during a simulation run. The user interface of the model updates and represents graphically the evolution of the output variables selected by the user during the model run. The model outputs are grouped into the following categories:

- *Change counts by state.* This category groups the outputs regarding the total number of change records that are currently in each change state (i.e. new, authorized, scheduled. See Fig. 1 for the complete list of change states).
- *Average durations of completed changes by state.* This category groups the outputs regarding the 4 M (Minimum, Maximum, Mean and Median) values for time that all completed change records have spent in each change estate.
- *Percentage of changes completed on time by type.* This category groups the outputs regarding the ratio of changes completed on time divided by the total number of changes completed, both values separated by change type (i.e. major, significant, and minor).
- *Percentage of closed changes by state.* This category groups the outputs regarding the number of closed changes by state divided by the total number of closed changes.
- *Number of completed changes by type.* This category groups the outputs regarding the number of completed changes by type.
- *Overall percentage of changes completed on time.* This category groups the outputs regarding the total number of changes completed on time divided by the total number of changes completed.
- *Overall process step count.* This category groups the outputs regarding the 4 M values for the number of process steps taken by completed changes by change type.
- *Change duration.* This category groups the outputs regarding the 4 M values for the duration of all completed changes for each change type.
- *Overall actual /predicted duration ratio.* This category groups the outputs regarding the 4 M values of the ratio of actual change duration divided by predicted duration for all completed changes.
- *Change success rate.* Percentage of changes deemed successful at

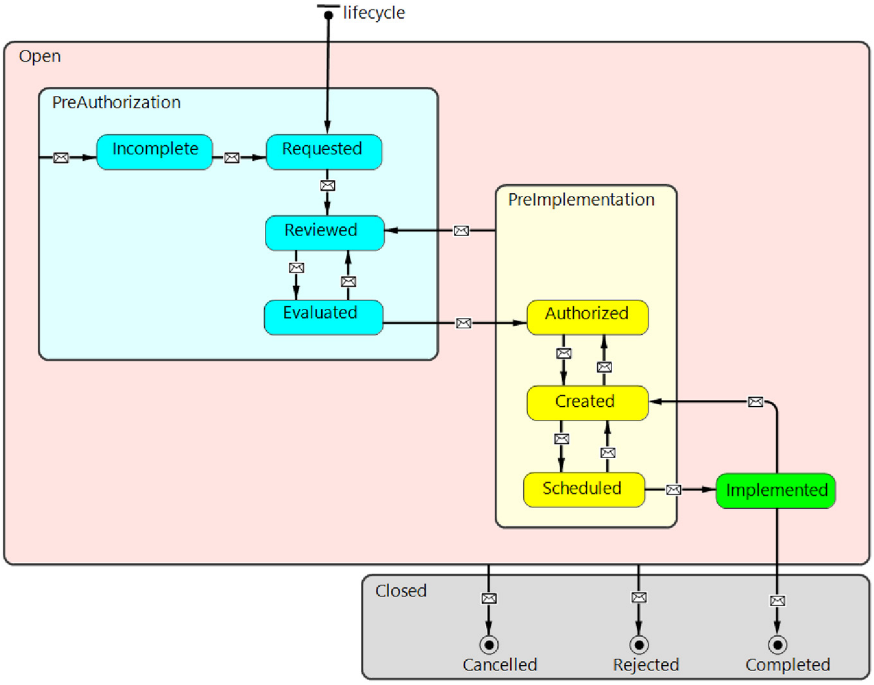


Fig. 2. Change Record agent type — Lifecycle statechart.

review/number of RFCs approved.

- *Change resource utilization.* This category groups the outputs regarding the percentage utilization of each resource type. It is measured as the ratio of the resource's busy time divided by its available working time.
- *Total change hours required.* This category groups the outputs regarding the sum of the number of hours used per resource type across all work shifts.

For the aggregated variables, their values are based on a one-year window of data, i.e. at any moment in time the output variable value will be based on the last 365 days of a closed change data.

3.5. Model implementation

The model is built under a multi-method simulation approach with components built under the agent-based and the discrete-event simulation approaches. The AnyLogic™ modeling and simulation software has been used to seamlessly integrate these components into a single simulation model. The combined usage of the different simulation methods allows us to take advantage of the strengths of each of the simulation approaches in modeling the ITIL change management process. After developing early prototypes under different simulation approaches (System Dynamics, Discrete-event and Agent-based), we decided that the agent-based approach was the most suitable to model the process. The early prototypes developed using the other simulation methods did not provide as good representation of the reality of the change management process and resulted in excessively complex models, with little flexibility and difficult to update.

In this model, there are several agent types, each one representing each of the model's logical sections. These agent types can be grouped logically into two groups: a) Change Record agent type, and b) Change Processing Role agent type. The event- and time-driven behaviors of the agents in the model have been implemented using statecharts and action charts. Additionally, the discrete-event simulation method is used to implement the utilization of resources by the agents. The following subsections provide detailed information about the different agent types and their behavior, as well as the different resource pools of the discrete-event model.

3.5.1. Change Record agents

In ITIL, a Change Record is created after receiving an RFC. It contains all the information regarding a single change and it is updated as the change lifecycle progresses. In our model, whenever an RFC is received, a new Change Record agent of the Change Record agent type is created and instantiated with the data of the RFC.

A Change Record agent type is implemented by two statecharts. Fig. 2 shows the statechart that describes the Change Record agent lifecycle. This statechart is based on the process flow for a normal change suggested by ITIL [36]. The boxes represent all the attainable states for a change record and the arrows represent all the allowable transitions between those states. Both forward and backward transitions between states are included to cover all possible outcomes (i.e. successful or non-successful) in the normal change process. Composite states have been added to give a logical grouping of states and to add clarity to some of the state transitions. Additionally, a second statechart is used to control the processing steps inside each change lifecycle state (see Fig. 3). This statechart keeps track of the progress of its enclosing change record. It allows processing to be paused (e.g. due to the end of the scheduled working periods or interruptions to process higher priority tasks), resumed again at a later time (e.g. when a suitable resource is available), until it is verified that the processing step has completed.

3.5.2. Change Processing Role agents

There are seven generalized roles that describe the tasks in the ITIL change management process: 1) initiator, who raises the request for a change, 2) practitioner, who submits requests for evaluation, 3) approver, who formally authorizes changes, 4) scheduler, who plans and schedule changes, 5) developer, who procures, develops, builds and tests whatever is needed for the change, 6) deployer, who takes built and tested solutions and implements them in the target environment, and 7) closer, who reviews the change and formally closes it. A comprehensive list of the tasks performed by each role can be found in the

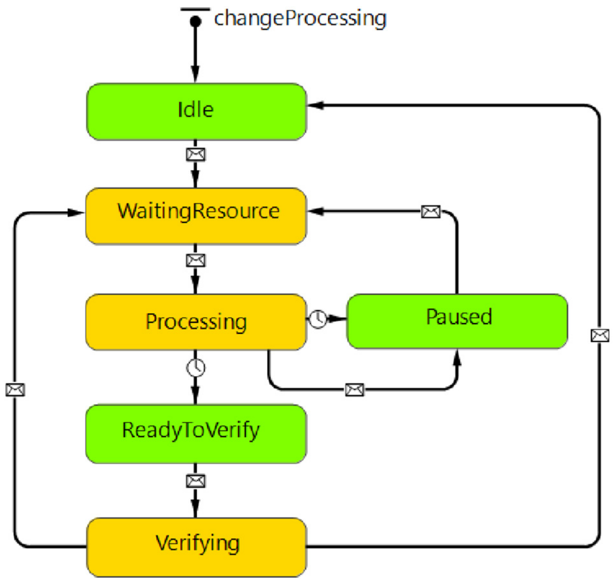


Fig. 3. Change Record agent type — Processing steps statechart.

ITIL Service Transition guide [36].

To model the tasks performed by each change processing role, a particular agent type, with its correspondent statecharts, has been created. Change record processing is carried out by a population of agents of these agent types. For a given change, its current state change record lifecycle state determines which change processing role and therefore which change processing agent should be processing the change to make it progress to the next lifecycle state.

To add realism, task processing within the simulation model can only occur when there are resources available and only during the resource's scheduled work times. The agents compete with each other to obtain resources from their assigned resource pools in order to perform the required change processing tasks on the change records. The model reproduces the common situation that occurs when the processing of longer running tasks get interrupted by higher priority tasks and lower priority tasks have to wait for others to complete before they can be advanced. In addition, at the end of each scheduled work period, processing will stop and will not start again until the start of the next scheduled period.

3.5.3. Change Processing resources

In the model, each of the change roles requiring change management resources sends a resource request to a queue. From there, as resources become available, the resource requests are selected in priority order, the resource is assigned and the task processing is performed. When the processing of the task has completed, the resource is released and becomes available again for any other queued requests. This queuing and task assignment functionality has been developed as a discrete-event model and embedded inside the agent-based model.

Each resource pool is constrained by the number of people compromising the group and by the group work schedule. The actual number of resources in each resource pool and their different shifts are configured using the input parameters of the model and thus can be varied during the simulation experiments.

During a simulation run, the user interface represents the real-time evolution of the main output variables as well as allows the user to make changes to the most relevant input parameters so that questions of the *What if...?* type can be analyzed.

Simulation modeling serves as a valuable technique towards ITIL implementation, especially in the process design, analysis and improvement phases [38]. In the particular case of the change management process, the model presented in this work enables the IT change manager to perform different types of simulation studies such as interactive simulations, parameter variation experiments, sensitivity analyses and optimization experiments that can help such managers improve their decisions in real-life situations, such as deciding on the prioritization strategies applied to evaluate the change requests, the size and features of the resources used or the activities designed to improve the duration of the change processing tasks.

4. Metaheuristics and simulation-based optimization

Metaheuristics [46] are a family of approximate techniques for solving optimization problems, capable of providing accurate solutions to difficult problems in reasonable time. Evolutionary Algorithms (EAs) are particularly suitable metaheuristics to solve Multi-objective Optimization Problems (MOP) [13]. One reason for that is that they deal simultaneously with a set of tentative solutions (the so-called population) that is evolved, allowing them to capture the dominance relations among solutions, helping to efficiently guide the search towards the Pareto-optimal front. Indeed, EAs can find good approximations of Pareto optimal set in a single run [13].

4.1. Metaheuristics algorithms

As previously stated, a large number of problems within the software engineering domain can be solved with metaheuristic techniques. Among their many different applications, they can be used to solve MOP, which are those involving multiple and conflicting objective functions simultaneously. In general, the solutions for MOPs form a Pareto front of non-dominated solutions, which can be formally defined as follows.

Given the minimization of n components $f_k, k = 1,\ldots,n$, of a vector function $\mathbf{f}$ of a vector variable $\mathbf{x}$ in $\mathscr{D}$, i.e., $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ldots,f_n(\mathbf{x}))$, and subject to inequality and equality constraints ($g_j(\mathbf{x}) \geq 0, j = 1,\ldots,J$ and $h_k(\mathbf{x}) = 0, k = 1,\ldots,K$):

Definition 1. Pareto Dominance. A vector $\vec{u} = (u_1, \ldots, u_k)$ dominates a vector $\vec{v} = (v_1, \ldots, v_k)$, denoted by $\vec{u} \preccurlyeq \vec{v}$ iff $\vec{u}$ is partially less than $\vec{v}$, i.e., $\forall i \in (1,\ldots,k), u_i \leq v_i \wedge \exists i \in (1,\ldots,k) : u_i < v_i$ (assuming minimization of all objectives).

The *Pareto front* is the set of optimal solutions, for which no objective can be improved without worsening at least one of the other objectives.

We have selected two well-known multi-objective evolutionary algorithms from the literature to solve the proposed problem, NSGA-II [14] and SPEA2 [53]. Although, there are more recent algorithms discussed in the literature, these two are among the most frequently used, and are still considered "state of the art" even though they were published more than fifteen years ago.

The Non-dominated Sorting Genetic Algorithm II, NSGA-II [14], is a Genetic Algorithm (GA) that assigns a fitness value to individuals according to their dominance level (through *Ranking* method) and diversity (thanks to *Crowding* technique). In every generation, a new population (with the same size as the original one) is created through the iterative application of the genetic operators. The next generation population is created by merging the two populations using the *Ranking* and *Crowding* methods to select the most promising solutions for the search process. Ranking orders solutions according to the dominance concept. *Crowding* assigns higher fitness to those solutions that are more isolated, representing the areas of the Pareto front approximation that are less explored.

The Strength Pareto Evolutionary Algorithm 2, SPEA2 [53], makes use of an external archive to store the best non-dominated solutions found. The size of the archive is limited, therefore the algorithm implements a mechanism to keep the most promising solutions when it

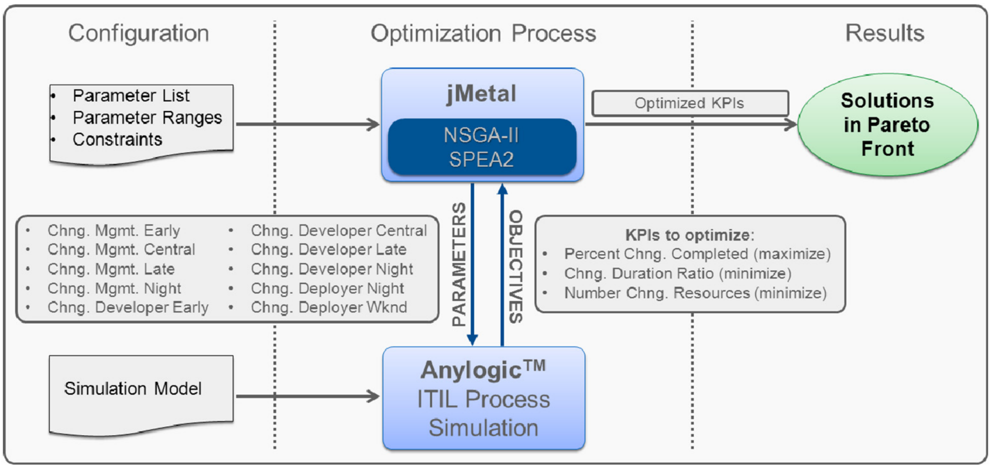


Fig. 4. Interaction between jMetal and Anylogic™.

becomes full. For that, the strength of individuals is defined in terms of the number of other individuals they dominate in the population. A fitness value is assigned to every individual, computed as the sum of its strength raw fitness and a density estimation, so that individuals with the lowest fitness can be discarded, if needed. The algorithm evolves the population through the iterative application of the variation operators on the individuals, storing all generated non-dominated solutions in the archive. After every generation, the population of the next generation is built from the current population and the archive, using the previously defined fitness to discard less promising solutions.

In this work, we use the implementation provided by the jMetal[2] framework [16], a metaheuristic algorithm framework that implements many of the current state of the art MOEAs, including NSGA-II and SPEA2. We also used the parameter settings proposed in the original papers, with the exception of the population and archive sizes that was set to 50 solutions. The termination condition of the algorithms was set to 5000 iterations.

4.2. Linking Anylogic™ and jMetal

The interaction we built between Anylogic™ simulator and jMetal optimization framework is shown in Fig. 4. As previously said, we use in this work NSGA-II and SPEA2 algorithms, among those provided in jMetal. These algorithms follow an iterative process in which new solutions (a solution in our context is an assignment of values to the decision variables, presented later in Table 3) are continuously being created by applying stochastic genetic operators on the solutions in the population. Every time a new solution is generated, Anylogic™ parameters are configured as described by the solution, and the simulation is run. Once the simulation is finished, jMetal takes the values of the defined KPIs from Anylogic™, and use them as the fitness of the solution.

The communication between jMetal and AnyLogic™ is carried out using the Google Protocol Buffers[3] to serialize the data to be exchanged between both applications as a TCP/IP client-server framework. In this way, AnyLogic™ acts as the server and, after starting the execution, executes via shell command the jMetal experiment and waits for a jMetal message. When AnyLogic™ receives a jMetal message, it runs a simulation experiment with the received variables and returns the simulation output to jMetal, that waits for the simulation results in order to evaluate the solution. This loop is repeated until jMetal reaches the stopping criteria and closes the communication. This approach has been also applied by other authors such as Beham et al. [7] to link Anylogic™ with HeuristicLab[4], another optimization framework.

5. Experimental work

In this section, we describe the experimental work carried out to perform multi-objective simulation optimization on the simulation model created. First, we introduce the motivation and formulation of the problem; next, we show the optimal solutions found by both (i) Anylogic™ (as standalone tool with its built-in optimizer) and (ii) combining Anylogic™ and jMetal for multi-objective simulation optimization.

5.1. Problem formulation

ITIL recommends each organization to define their own Critical Success Factors (CSFs) to achieve their particular mission, objectives or goals. The ITIL Service Transition Glossary defines a CSF as *"Something that must happen if a Process, Project, Plan, or IT Service is to succeed"* [36]. Basically, a CSF can be understood as a high-level goal critical for the success of the organization. Given the strategic nature of CSFs, they are often tracked and measured in terms of a set of Key Performance Indicators (KPIs). KPIs can be defined as the most important metrics used to report on process performance that contributes to measure the achievement of CSFs. Thus each CSF will have a small set of KPIs associated. Frequently, CSFs are qualitative and meant to answer the question *What should be done to achieve success?* Complementarily, KPIs are quantitative and meant to answer the question *Are we successful?*

For the purpose of this study, we will consider an IT service organization with a set of CSFs defined. Let us assume that among their CSFs, there is one related to delivering process efficiency. This decision is based upon the widely accepted assumption that every organization should succeed in delivering efficient processes.

The problem of IT change process efficiency can be defined as another instance of the well-known time-cost-quality triangle, in which three conflicting factors need to be optimized in order to achieve efficiency [2,23]. Under this consideration, the KPIs that can be associated with delivering an effective IT service change process are:

- *KPI 1:* Percentage of changes completed successfully within the time estimates, as a measure of the time factor. The higher the value of this KPI, the better for the process efficiency.

[2] https://github.com/jMetal/jMetal.
[3] https://developers.google.com/protocol-buffers/.
[4] http://dev.heuristiclab.com/.

Table 3
Solutions selected from the Pareto front and solution obtained by AnyLogic™.

	Decision variable	Solution 1	Solution 2	Solution 3	AnyLogic™
Input parameters	*Change Management Early*	0	0	0	1
	Change Management Central	1	1	1	1
	Change Management Late	2	2	0	1
	Change Management Night	1	1	1	1
	Change Developer Early	1	7	7	5
	Change Developer Central	4	4	3	6
	Change Developer Late	9	2	2	1
	Change Developer Night	1	0	0	1
	Change Deployer Night	3	3	4	3
	Change Deployer Weekend	1	1	1	2
KPIs	*Percent Changes Completed OnTime*	94.77	93.67	93.07	92.83
	Change Duration Ratio	0.70	0.84	1.05	0.88
	Number of Change Resources	23	21	19	22

- *KPI 2:* Actual change duration/estimated change duration, as a measure of internal quality. The lower the value of this KPI, the better for the process efficiency.
- *KPI 3:* Overall number of resources utilized, as a measure of cost. The lower the value of this KPI, the better for the process efficiency.

Before using the simulation model to show how multi-objective simulation optimization can help decision-makers in achieving efficiency-based CSF, it is necessary to establish the simulation scenario that will serve as the foundation for this experiment. One of the critical decisions that change process managers need to make relates to the configuration of the process staff, as different staff configurations may lead to different outcomes. We assume that the staff involved in the fulfillment of the change management process have the following roles (besides the initiator, the practitioner, the approver, and the closer): a) change management, b) change developer, and c) change deployer. For each role, let us assume there are four possible basic shifts:

- Central, from 9 am to 6 pm, with one-hour break at 1 pm.
- Early, from 6 am to 3 pm, with one-hour break at 10 am.
- Late, from 2 pm to 11 pm, with one-hour break at 6 pm.
- Night, from 10 pm to 7 am, with a break at 2 am.

For change developers, there is a special shift:

- Weekend, Saturdays and Sundays, 7 am to 10 pm, with one-hour breaks at noon and 5 pm.

We are interested in finding the optimal combination of staff resulting in the best possible solution towards the achievement of the CSF measured in terms of the KPIs previously described. An optimization process can help in finding these values by running repetitive simulations of the model, each with different values in the input parameters and locating the values that solve the problem. For this particular study, the optimization parameters, i.e., decision variables, selected are the ones defining the number of people per role who are working on each of the working shifts.

In order to propose a realistic simulation scenario, and based on our own experience, we set the problem variables and their range of allowed values as listed below. However, users will probably need to adjust the values of these parameters in order to tailor the simulation to his/her own particular needs.

- Bounds for change management central staff: $cmC \in [1,3]$
- Bounds for change management early staff: $cmE \in [0,2]$
- Bounds for change management late staff: $cmL \in [0,2]$
- Bounds for change management night staff: $cmN \in [0,1]$
- Bounds for change developer central staff: $cDvC \in [3,14]$
- Bounds for change developer early staff: $cDvE \in [0,1]$
- Bounds for change developer late staff: $cDvL \in [0,11]$
- Bounds for change developer night staff: $cDvN \in [0,3]$
- Bounds for change deployer night staff: $cDpN \in [1,5]$
- Bounds for change deployer weekend staff: $cDpW \in [1,5]$

Not every combination of values for the optimization parameters is acceptable in a real-life situation. For this reason, we also need to add some constraints upon the values of the optimization parameters. The constraints set for the optimization experiments performed is formally defined as:

- The change management staff cannot exceed 3 workers: $cmC + cmE + cmL + cmN \le 3$.
- The change developer staff cannot exceed 14 workers: $cDvC + cDvE + cDvL + cDvN \le 14$.
- The change deployer staff cannot exceed 6 workers: $cDpN + cDpW \le 6$.

Formally, our problem is defined as the optimization of the following three functions:

$$f_{KPI1} = \arg\max \left\{ \frac{\sum_{i=1}^{|C|} x_i}{|C|} \times 100 \;\middle|\; x_i = \begin{cases} 1 & \text{if} \quad t_i \le et_i \\ 0 & \text{otherwise} \end{cases} \right\} \tag{1}$$

$$f_{KPI2} = \arg\min \left\{ \sum_{i=1}^{|C|} \frac{t_i}{et_i} \right\} \tag{2}$$

$$f_{KPI3} = \arg\min \; \{cmC + cmE + cmL + cmN + cDvC + cDvE + cDvL + cDvN + cDpN + cDpW\} \tag{3}$$

where C is the set of required changes, t_i is the real time required to process change i and et_i is the estimated time to process change i.

It is important to note that, as described in Section 3.3, the number of working shifts, their respective timetable and the allocation of staff to each shift can be configured by using the different input parameters of the simulation model. The values shown in this section are only intended to illustrate a possible configuration of the process within an IT organization that is based on the authors' experience and the information found in the available literature. Similarly, the configuration of the parameters range and constraints is also illustrative and can be adapted to the values of interest in any given particular situation.

5.2. Anylogic™ optimization

Anylogic™ simulation software comes with OptQuest™ [5] optimization engine. This optimization tool helps to find the values of model

[5] http://www.opttek.com/OptQuest

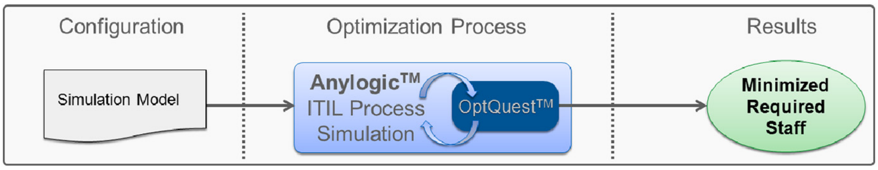


Fig. 5. Anylogic™ optimization via OptQuest™ software.

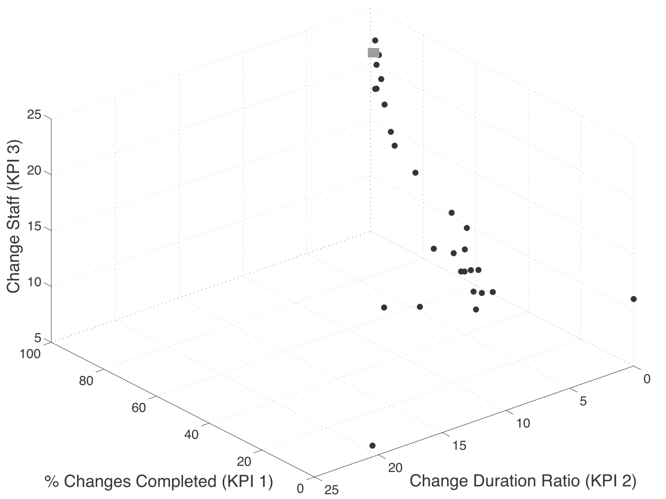


Fig. 6. Pareto front of the best non-dominated solutions found. The solution found by Anylogic™ optimizer is represented as a gray square.

parameters that maximize or minimize the model's objective function. Fig. 5 depicts this process.

We used Anylogic™ optimization facilities to find a solution for the problem previously stated. However, the type of simulation-based optimization problem that one can solve with Anylogic™ falls in the category of single-objective optimization. For this reason, we created a simulation optimization experiment in Anylogic™ that minimizes the total number of staff involved in the change process, while assuring that at least 90% of the changes are completed on time.

5.3. Multi-objective optimization

We executed six independent runs of NSGA-II and SPEA2 to find accurate trade-off solutions to our problem. The simulation process is shown in Fig. 4. From all the solutions found in the different runs of the two algorithms, we built a single Pareto front containing all best non-dominated solutions. The resulting Pareto front is composed of 27 highly accurate solutions, and it is shown in Fig. 6 (the solution provided by the optimizer embedded in Anylogic™ is plotted as a gray square for reference). In this figure, we can observe that 89% of the solutions require a change duration ratio less than 5.0. Additionally, it can be seen that those solutions with the highest percentage of changes completed are, as it could be expected, the more expensive ones (i.e., the ones requiring the highest number of staff). The solutions with more than 80% changes completed on time require at least 15 persons.

Furthermore, we are interested in analyzing our solutions, and compare them versus the one reported by AnyLogic™. To do so, we select a subset of three solutions we consider interesting, out of the 27 ones in the Pareto front approximation we computed, just like a decision maker would do. We also defined and followed some ad hoc criteria. The selected solutions are shown in Table 3, and we consider they are interesting solutions for the organization for the following reasons:

- **Solution 1**. The one with the best percentage of completed changes on time (*KPI 1*).
- **Solution 2**. We first select the 30% best solutions from the Pareto front, according to the percentage of changes completed on time (*KPI 1*). From those solutions, we keep the 30% best solutions that minimize the duration ratio (*KPI 2*). Finally, from this resulting set, the solution that needs the minimum number of staff was selected as Solution 2 (*KPI 3*).
- **Solution 3**. We choose the 30% best solutions from the Pareto front in terms of the duration ratio (*KPI 2*) and, among them, we further select the 30% best solutions guided by the number of staff that were selected (*KPI 3*). Solution 3 is the one maximizing the percentage of changes completed on time (*KPI 1*) from the selected ones.

From the ITSM point of view, all these solutions are among the most expensive ones for the organization, as they require a large number of personnel. However, the solutions found that needed a low number of personnel were always poor solutions that do not lead to the process-efficiency CSF achievement. If we compare the solutions provided by our approach versus that of Anylogic™ (shown in Table 3), we can see that the latter cannot outperform any of the selected solutions from the multi-objective algorithms. Indeed, Solution 2 (from the proposed multi-objective approach) outperforms the solution provided by Anylogic™ for the three objectives.

It is beyond the scope of this work to conduct a detailed comparison of the two used multi-objective optimization algorithms. However, we noticed that the three best solutions (those presented in Table 3) were all found by the SPEA2 algorithm. Both algorithms implement the same selection, recombination, and mutation operators. In addition, the number of non-dominated solutions in the Pareto front approximations do not reach the limit in any of the two algorithms (set to 50 in our experiments), so the strength raw fitness operator of SPEA2 does not have any effect on the performance of the algorithm (note that it is used to select the solutions to discard from the archive when its limit is exceeded). Therefore, we suspect that the use of an external archive of solutions benefits SPEA2 against NSGA-II. Implementing an external archive allows SPEA2 keeping a more diverse population with respect to NSGA-II, which requires a stronger elitist policy to avoid missing any non-dominated solutions from the population.

6. Conclusions and future work

IT service management frameworks provide important guidance for change management in IT organizations. However, decision-making in this area is a complex process that involves a large number of difficult decisions which have a crucial impact on the achievement of the organization's Critical Success Factors (CSFs). Similarly to many other engineering problems, efficient change management requires optimizing several objectives simultaneously, such as maximizing the percentage of changes completed on time and minimizing the change duration ratio and the number of resources used.

In this paper, to help change process managers make better decisions, we have described a proposal based on using multi-objective optimization to optimize the outputs of a multi-method simulation model of the ITIL change process. The simulation model built is based on the agent-based and discrete-event simulation paradigms and simulates the whole process lifecycle, from change initiation to change closure, allowing the study of the average yearly performance of the process. Google Protocol Buffers has been used to facilitate the exchange of information between Anylogic™ software, which runs the simulation model, and the jMetal framework, which implements the multi-objective optimization algorithms used in this study.

To illustrate how multi-objective simulation optimization can help improve decision-making in this area, we have formulated a problem consisting of helping to achieve a very common CSF aimed at ensuring change process efficiency, i.e. the process is carried out in a timely and cost-effective way. The problem of IT change process efficiency has been defined as another instance of the well-known time-cost-quality triangle, in which three conflicting variables need to be optimized in order to achieve efficiency. After setting a particular simulation scenario, selecting the decision variables and setting the constraints for the optimization experiment, the problem was solved with two well-known multi-objective evolutionary algorithms, i.e. NSGA-II and SPEA2. Three solutions were selected from the best non-dominated solutions found in our experiments. These solutions were selected for being the ones leading to optimal values for the Key Performance Indicators (KPIs) associated with the CSF previously mentioned. Furthermore, the solutions found by the multi-objective evolutionary algorithms were compared with the solution provided by the Anylogic™ built-in optimizer. As a result, the multi-objective evolutionary algorithms could find better solutions than the ones offered by Anylogic™ in all objectives.

Even though our experimental work has been done in the scope of the ITIL change management process, our proposal is clearly extensible to other ITIL processes or other processes defined in other ITSM frameworks. In fact, we plan to continue the development of the simulation models of the ITIL processes that interact with the change management process such as: configuration management, problem management and incident management, being the last one already built [38], to give a more complete support to decision-making towards process efficiency in the Service Transition stage of the IT service lifecycle. Since this proposal benefits from the advantages of the multi-objective optimization approach applied to the results of simulation models, the range of the solutions provided in the Pareto front can help IT managers understand the effect of different management strategies and improve their decision-making towards more efficient processes.

As future work, we also intend to apply multiple algorithms to this new optimization problem we have defined in this work, and carry out a thorough comparison of their performance. Also, we need to consider parallel versions of the algorithms, because the simulations require a high computational cost. In this sense, our architecture with Google *protobuffer* allows us to easily distribute the load to multiple machines. Finally, as the number of variables and solutions in the Pareto can be large, we will need to explore visualization and clustering techniques to present the results, as well as defining new multicriteria decision making methods to choose the solutions to adopt among those in the Pareto front.

Acknowledgments

This research was partly supported by the Spanish Ministry of Science and Innovation and the ERDF funds under projects BadgePeople (TIN2016-76956-C3-3-R), SAVANT (TIN2014-60844-R), the Ramón y Cajal fellowship with contract RYC-2013-13355 and the Andalusian Plan for Research, Development and Innovation (TIC-195). The authors also thank David W. Gawn for his collaboration in an earlier version of this work, the Universities of Cadiz and Alcala and the anonymous reviewers for their help in improving the manuscript.

References

[1] T. Amin, T. Grollius, E. Ortner, Language-Critical Development of Process-Centric Application Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 31–46.
[2] R. Atkinson, Project management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteria, International Journal of Project Management 17 (1999) 337–342.
[3] Axelos,. ITIL® - IT Service Management. URL: https://www.axelos.com/best-practice-solutions/itil (Accessed: June 2018).
[4] C. Bartolini, C. Stefanelli, M. Tortonesi, SYMIAN: a simulation tool for the optimization of the IT incident management process, Managing large-scale service deployment, IFIP; IEEE, 200819th IFIP/IEEE International Workshop on Distributed Systems - Operations and Management.
[5] C. Bartolini, C. Stefanelli, M. Tortonesi, Business-impact analysis and simulation of critical incidents in IT service management, IFIP/IEEE Int. Symp. on Integrated Network Management, 2009, pp. 9–16.
[6] C. Bartolini, C. Stefanelli, M. Tortonesi, Modeling IT support organizations using multiple-priority queues, 2012 IEEE Network Operations and Management Symposium, 2012, pp. 377–384.
[7] A. Beham, E. Pitzer, S. Wagner, M. Affenzeller, K. Altendorfer, T. Felberbauer, M. Bäck, Integration of flexible interfaces in optimization software frameworks for simulation-based optimization, Genetic and Evolutionary Computation Conference, GECCO, 2012, pp. 125–132.
[8] T. Chesney, S. Gold, A. Trautrims, Agent based modelling as a decision support system for shadow accounting, Decision Support Systems 95 (2017) 110–116.
[9] M. Chica, A.A. Juan Pérez, O. Cordon, D. Kelton, Why Simheuristics? Benefits, Limitations, and Best Practices When Combining Metaheuristics with Simulation, (2017) https://ssrn.com/abstract=2919208 SSRN.
[10] P. Chołda, P. Jaglarz, Optimization/simulation-based risk mitigation in resilient green communication networks, Journal of Network and Computer Applications 59 (2016) 134–157.
[11] W. Cordeiro, G. Machado, F. Andreis, J.A. Wickboldt, R. Lunardi, A. dos Santos, C. Both, L. Gaspary, L. Granville, D. Trastour, C. Bartolini, CHANGEMINER: a solution for discovering IT change templates from past execution traces, IFIP/IEEE Symp. on Integrated Network Management, 2009, pp. 97–104.
[12] R. Dawson, C. Dawson, Practical proposals for managing uncertainty and risk in project planning, International Journal of Project Management 16 (1998) 299–310.
[13] K. Deb, D. Kalyanmoy, Multi-Objective Optimization Using Evolutionary Algorithms, John Wiley & Sons, Inc., New York, NY, USA, 2001.
[14] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002) 182–197.
[15] B. Dorronsoro, P. Ruiz, G. Danoy, Y. Pigné, P. Bouvry, Evolutionary Algorithms for Mobile Ad Hoc Networks, Wiley/IEEE Computer Society, 2014.
[16] J.J. Durillo, A.J. Nebro, jMetal, A Java framework for multi-objective optimization, Advances in Engineering Software 42 (2011) 760–771.
[17] M.P. Fanti, G. Iacobellis, W. Ukovich, V. Boschian, G. Georgoulas, C. Stylios, A simulation based Decision Support System for logistics management, Journal of Computational Science 10 (2015) 86–96.
[18] Forbes Insights, Delivering Value to Today's Digital Enterprise - The State of IT Service Management, 2017, Technical Report, Forbes, 2017.
[19] X. Fu, M. Dong, S. Liu, G. Han, Trust based decisions in supply chains with an agent, Decision Support Systems 82 (2016) 35–46.
[20] ISACA COBIT 5 - A Business Framework for the Governance and Management of Enterprise IT, URL: www.isaca.org/cobit (Accessed: June 2018).
[21] ISO/IEC 20000-1:2011. Information technology - Service management - Part 1: Service management system requirements. URL: https://www.iso.org/standard/51986.html (Accessed: June 2018).
[22] ITIL, ITIL Continual Service Improvement 2011 Edition, The Stationery Office, Norwich, 2011.
[23] A.H.J.B. Ebbesen, Re-imagining the iron triangle: embedding sustainability into project constraints, PM World Journal 2 (2013) 1–13.
[24] C. Jiang, H. Shi, H. Zhang, A novel quantitative model for system performance based on distributed IT management structure, IEEE Int. Conf. on Communication Software and Networks (ICCSN), 2015, pp. 449–453.
[25] A.A. Juan, J. Faulin, S.E. Grasman, M. Rabe, G. Figueira, A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems, Operations Research Perspectives 2 (2015) 62–72.
[26] M.I. Kellner, R.J. Madachy, D.M. Raffo, Software process simulation modeling: Why? What? How? Journal of Systems and Software 46 (1999) 91–105.
[27] M. Krey, B. Harriehausen, M. Knoll, S. Furnell, IT governance and its impact on the Swiss Healthcare, 12th International Conference on Computer Modelling and Simulation, 2010, pp. 340–345.
[28] A.M. Law, How to build valid and credible simulation models, Proceedings of the 40th Conference on Winter Simulation, Winter Simulation Conference, 2008, pp. 39–47.
[29] A.M. Law, Simulation Modeling and Analysis, 5th ed., McGraw-Hill, 2014.
[30] R.-C. Lin, M.Y. Sir, K.S. Pasupathy, Multi-objective simulation optimization using data envelopment analysis and genetic algorithm: specific application to determining optimal resource levels in surgical services, Omega 41 (2013) 881–892.
[31] X. Luo, K. Kar, S. Sahu, P. Pradhan, A. Shaikh, On improving change management process for enterprise IT services, IEEE International Conference on Services Computing (SCC2008), 2008, pp. 341–348.
[32] A. Moorer, R. Antao, Improving management of information technology: system dynamics analysis of IT controls in context, Proceedings of the 24th International Conference of the System Dynamics Society, 2006.
[33] MS Operation Framework (MOF) and Service Management. URL: blogs.technet.microsoft.com/mof/ (Accessed: June 2018).
[34] J. Nocedal, S.J. Wright, Numerical optimization, Springer, 2006.
[35] D.C. Novak, C.T. Ragsdale, A decision support methodology for stochastic multi-criteria linear programming using spreadsheets, Decision Support Systems 36 (2003) 99–116.
[36] C. Office, ITIL Service Transition, The Stationery Office, London, 2011.
[37] S. Ólafsson, J. Kim, Simulation optimization: simulation optimization, Proceedings

of the 34th conference on Winter simulation: exploring new frontiers, Winter Simulation Conference, 2002, pp. 79–84.
[38] E. Orta, M. Ruiz, Met4ITIL: a process management and simulation-based method for implementing ITIL, Computer Standards & Interfaces (2018).
[39] E. Orta, M. Ruiz, N. Hurtado, D. Gawn, Decision-making in IT service management: a simulation based approach, Decision Support Systems 66 (2014) 36–51.
[40] R. Rebouças, R. Santos, J. Sauvé, A. Moura, IT Change Management Challenges - Results of 2006 Web Survey, Technical Report HPL-TR-2006-3, Computing Systems Department, Federal Univ., Campina Grande, 2006.
[41] D. Schmaranzer, R. Braune, K.F. Doerner, A discrete event simulation model of the Viennese subway system for decision support and strategic planning, Winter Simulation Conference (WSC), 2016, pp. 2406–2417.
[42] T. Setzer, K. Bhattacharya, H. Ludwig, Change scheduling based on business impact analysis of change-related risk, IEEE Transactions on Network and Service Management 7 (2010) 58–71.
[43] Y.B. Shrinivasan, G.B. Dasgupta, N. Desai, J. Nallacherry, A Method for Assessing Influence Relationships among KPIs of Service Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 191–205.
[44] E. Silva, Y. Chaix, Business and IT governance alignment simulation essay on a business process and IT service model, 41st Annual Hawaii International Conference on System Sciences (HICSS 2008), 2008 434–434.
[45] K. Srinivasan, K. Kumar, Multi-objective simulation-optimization model for long-term reservoir operation using piecewise linear hedging rule, Water Resources Management 32 (2018) 1901–1911.
[46] E.-G. Talbi, Metaheuristics: From Design to Implementation, Wiley Publishing, 2009.
[47] S. Thanheiser, L. Liu, H. Schmeck, SimSOA - an approach for agent-based simulation and design-time assessment of SOC-based IT systems, ACM Symposium on Applied Computing, 2009, pp. 2162–2169.
[48] TMForum Business Process Framework (eTOM). URL:www.tmforum.org/business-process-framework (Accessed: June 2018).
[49] K.O. Willis, D.F. Jones, Multi-objective simulation optimization through search heuristics and relational database analysis, Decision Support Systems 46 (2008) 277–286.
[50] X.-F. Xu, Z.-J. Wang, T. Mo, The Current State and Development Plan of Research and Education on SSME in Harbin institute of Technology, Springer US, Boston, MA, 2008, pp. 219–224.
[51] B. Yang, S. Zeng, N. Ayachitula, R. Puri, SLA-driven applicability analysis for patch management, 12th IFIP/IEEE International Symposium on Integrated Network Management (IM 2011), 2011, pp. 438–445.
[52] M.A. Zaffar, R.L. Kumar, K. Zhao, Diffusion dynamics of open source software: an agent-based computational economics (ACE) approach, Decision Support Systems 51 (3) (2011) 597–608.
[53] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Technical Report 103. Comp. Eng. & Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), (2001).

Mercedes Ruiz is an associate professor of software and services engineering and advanced database technologies at the University of Cádiz (Spain). She holds a degree in Computer Science and a PhD in Software Engineering from the University of Seville (Spain) (2003). Dr. Ruiz leads the Software Process Improvement and Formal Methods Research Group at the University of Cadiz. Her research interests are simulation-based decision support applied in software and services engineering, quality improvement in people-driven processes, gamification and simulation-based learning and assessment. She is the author or co-author of multiple papers in international conferences and impact journals and serves as reviewer for several impact journals and conferences.

Javier Moreno received the degree in engineering (2008) from the University of Alcalá (Spain). He has worked for four years as department head in Indra, in the area of IT and transport. Previously, he worked as team leader for R&D traffic projects for four years. His main research interests include data mining, multi/many-objective optimization and their application to solve real-world problems.

Bernabé Dorronsoro received the degree in engineering (2002) and the Ph.D. in Computer Science (2007) from the University of Málaga (Spain), and he is currently working at the University of Cádiz (Spain). He worked for five years as a post-doc at the University of Luxembourg, and as a Marie Curie Fellow at the University of Lille (France) for two more years. His main research interests include sustainable computing, Grid computing, ad hoc networks, the design of new efficient metaheuristics, and their application for solving complex real-world problems in the domains of logistics, telecommunications, bioinformatics, combinatorial, multiobjective, and global optimization. He has published over 30 articles in high impact journals and two authored books. Dr. Dorronsoro has been a member of the organizing committees of a large number of conferences and workshops, and he usually serves as reviewer for leading impact journals and conferences

Daniel Rodríguez is currently an associate professor at the Computer Science Department of the University of Alcala, Madrid, Spain. In the past, he has been a lecturer at the University of Reading (2001–2006). He earned his degree in Computer Science at the University of the Basque Country and Ph.D. degree at the University of Reading, UK in 2003. His research interest include software engineering in general and the application of data mining and optimization techniques to software engineering problems in particular. He is a member of IEEE and ACM associations.

Chapter 6

Design of a TTC Antenna using Simulation and Multi-objective Evolutionary Algorithms

The design of a Compact Dual-band Equatorial helix antenna using Computational Electromagnetic Methods (CEM) together with multi-objective optimization algorithms is presented. These antennas are used for Telemetry, Tracking, and Control (TTC) of satellites from the terrain base station. In order to optimize the parameters an antenna, a simulation-optimization process is shown along a real case study. The parameters of the antenna that fulfills the radiation patterns needed for the communication are obtained using a simulation tool called MONURBS together with two well-known multi-objective algorithms: NSGA-II and SPEA-2. In this work, a comparison with previous designs and the antenna prototype are presented, showing that this approach can obtain multiple valid solutions and accelerate the design process.

Feature Article:

DOI. No. 10.1109/MAES.2019.2916535

Design of a TTC Antenna Using Simulation and Multiobjective Evolutionary Algorithms

***Javier Moreno, Iván González, Daniel Rodríguez,* University of Alcalá, Madrid, Spain**

INTRODUCTION

In this paper, we present a case study about how to apply simulation–optimization, i.e., the application of simulation together with a multiobjective algorithm can help us to optimize the design parameters of an antenna with very stringent constrains. The objective is to obtain a compact dual-band helical antenna for Telemetry, Tracking, and Control (TTC) of satellites. A TTC subsystem provides the communication between a satellite and a ground station, as illustrated in Figure 1. The Telemetry system monitors the satellite retrieving its health and status of other subsystems and sending data to the station. The Tracking subsystem manages the satellite position in its orbit while the Control subsystem allows us to command the satellite, reconfiguring it if necessary. Within a TTC system, the antenna is undoubtedly the most critical part. The antenna must guarantee a proper operation within the established parameters and due to its constraints, its design can be extremely complex.

In our case study, the antenna needed to be able to operate in the S-Band at 1.81 and 2.55-GHz frequencies:

- Minimizing the cross-polarization level.
- Maximizing the gain for the Right-Hand Circular Polarization (RHCP).

In our case study, the parameters of the antenna that fulfills the radiation patterns needed for the communication are obtained using a simulation tool called MONURBS [1] together with two well-known multiobjective algorithms: Non-dominated Sorting Genetic Algorithm (NSGA-II) [2] and the Strength Pareto Evolutionary Algorithm-2 (SPEA-2) [3].

In this paper, a comparison with previous designs and the antenna prototype is presented, showing that simulation–optimization can obtain multiple valid solutions and accelerate the design process.

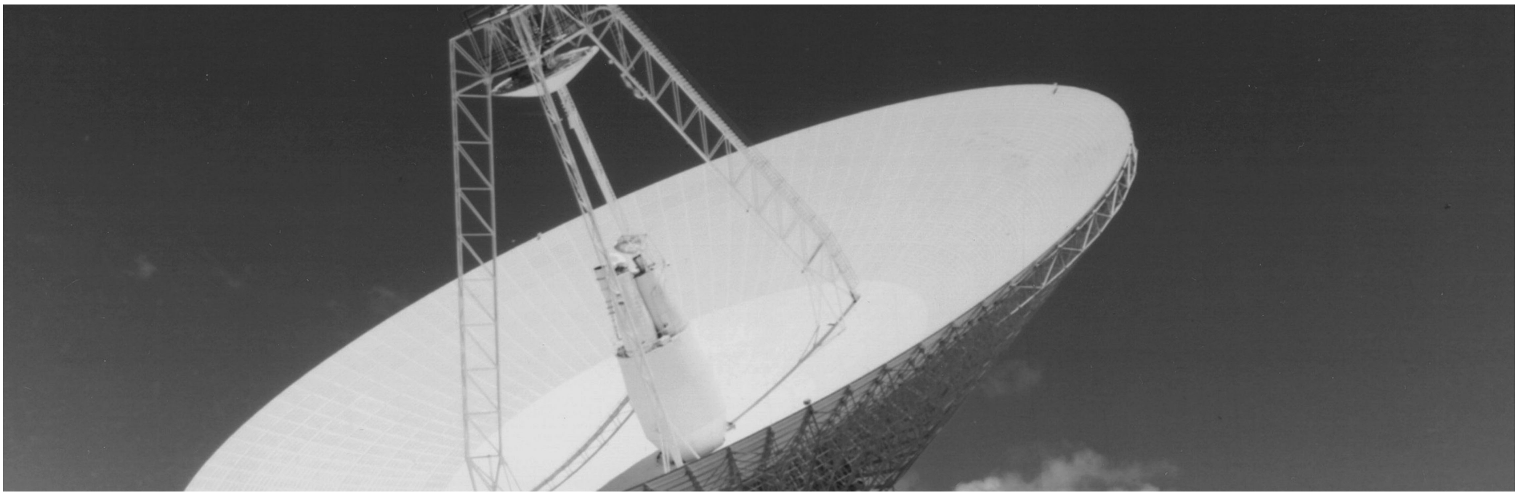
Authors' current address: J. Moreno, I. González, and D. Rodríguez, Department of Computer Science, University of Alcalá, Madrid 28805, Spain, E-mail: (javier.morenom@edu.uah.es).
Manuscript received August 30, 2018, revised December 13, 2018, and ready for publication May 10, 2019.
Review handled by M. D. R-Moreno.
0885-8985/19/$26.00 © 2019 IEEE

PREVIOUS EXPERIENCE AND MOTIVATION

In a previous work, González et al. [4] presented the design of a compact dual-band helical antenna for TTC applications in satellites. In [5], we proposed the use of NSGA-II to reduce the cost of time and optimize the design of the helical antenna. Here, we extend our proposal by applying the SPEA-2 algorithm to obtain multiple valid solutions and to expedite the process in future designs. The initial work was immersed in a ESA project 20995/NL/ST/na, "S-Band Toroidal Antenna," where the main contractor was RYMSA.[1]

Although the geometric model is quite simple, it needs to be parametrized according to rigorous requirements where there are several objectives that the optimization process has to deal with. In the previous work [4], the optimization process was carried out applying the *Gradient Descent* (GD) algorithm with a simulation tool called MONURBS to analyze and obtain the radiation pattern of the antenna. This GD method was used with a cost function that depended on the antenna requirements. However, it resulted in a very complex problem with a large number of maximums and minimums where the application of the GD method was difficult and not appropriate (it was more like a random sampler in the search space). A huge number of simulations were needed to obtain a valid solution that satisfied all the requirements simultaneously. It was, therefore, an extremely CPU intensive task that needed a very large time span (several months). As a consequence, we started tackling this problem as a case study applying multiobjective optimization techniques.

[1] http://www.tryo.es/



Although there are a large number of multiobjective algorithms, we selected the two most popular and well-known ones, NSGA-II and SPEA-2, for our case study. Both algorithms are by far the most popular and referenced in the multiobjective literature.

PROBLEM DEFINITION

For the purpose of optimization, we can divide the problem into two parts:

(1) The problem parameters, which define the antenna geometry.

(2) The problem objectives, which define how good is the antenna according to the specified requirements.

A helix antenna is formed by one or more strips wrapped helically. The geometrical model of a helix antenna is defined by a truncated cone. The antenna has four rolled strip in the form of a helix from the bottom circle to the topside circle. The strips are short-circuited in the top of the antenna. Finally, a post is set internally to the four strips to be mechanically strong enough. Therefore, the antenna geometry can be defined by four parameters (see Figure 2):

- Bottom radius (r).
- Top radius (R).
- Height (h).
- Number of turns of the helix (t).

The most important electromagnetic requirements were stated as follows:

- Dual Band operation at 1.81 and 2.55 GHz in the S Band (two frequencies).
- RHCP, the main electrical field that radiates the antenna.
- Peak maximum gain greater than 2 dBi for the RCHP polarization.
- Minimum gain of 0 dBi in the range coverage for the RHCP polarization.
- Cross-polar polarization level had to be smaller than −12 dB (difference between LHCP—Left Hand Circular Polarization—and RHCP), this is difficult to obtain.
- The above specifications in an equatorial radiation pattern had to be satisfied in the elevation angle with a range between 70° and 110°.

Figures 3 and 4 show these requirements graphically. The mask has to be satisfied for radiation pattern in the desired

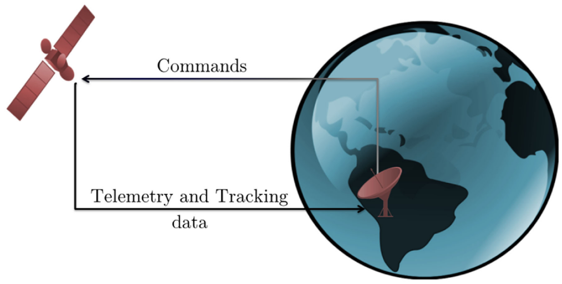


Figure 1.
TTC communication system.

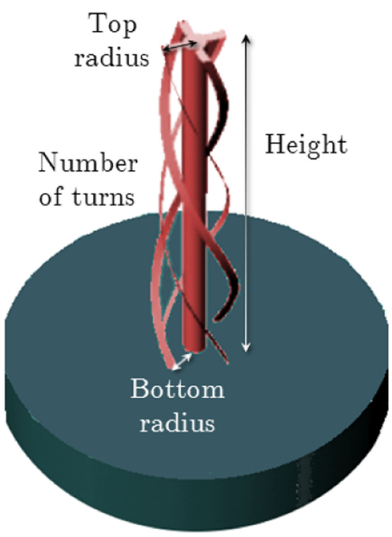


Figure 2.
Geometrical parameters of a helical antenna.

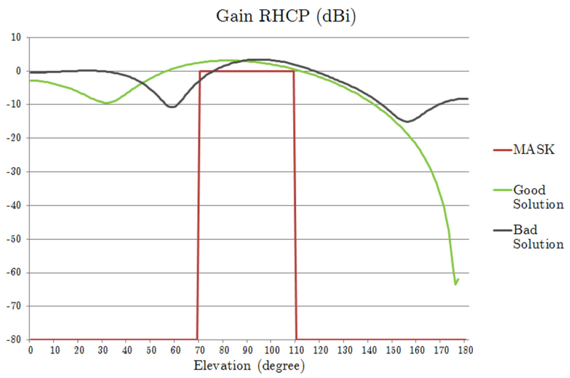


Figure 3.
Gain objective.

directions for the main (RHCP) as well as for the cross-polar components (difference between LHCP-RHCP).

Additionally, the weight of the prototype had to be as small as possible, therefore it was important to reduce the volume of the antenna. The volume of the antenna can be calculated using the truncated cone volume formula:

$$v = \frac{1}{3} \cdot \pi \cdot h \cdot (R^2 + r^2 + R \cdot r). \tag{1}$$

Due to the difficulty of the problem, we decided to take out the volume requirement of the optimization process. The problem objectives will be:

- Maximize the RHCP gain for 1.81-GHz frequency.
- Minimize the cross-polar polarization level for 1.81-GHz frequency in the range between 70° and 110°. In this range, gain must be above 0 dBi.
- Maximize the RHCP gain for 2.55-GHz frequency.
- Minimize the cross-polar polarization level for 2.55-GHz frequency in the range between 70° and 110°. In this range, gain must be above 0 dBi.

Once the optimization process ends, solutions that meet all requirements are filtered and the volume of each solution is computed.

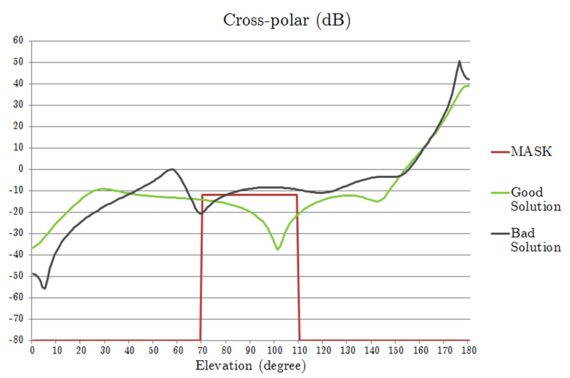


Figure 4.
Cross-polar objective.

Finally, joining problem parameters and problem objectives, we define a *problem solution* or simply a *solution* s, as a tuple of two vectors, $s = (s_p, s_o)$, where:

- s_p defines the four geometric parameters, i.e.: $s_p = (s_{p_1}, \ldots, s_{p_m}) : m = 4$.
- s_o defines the four radiation objectives, i.e.: $s_o = (s_{o_1}, \ldots, s_{o_n}) : n = 4$.

THEORETICAL BACKGROUND

ANALYSIS OF ANTENNAS: NUMERICAL METHODS

Before manufacturing, the antenna must be designed and optimized to satisfy the requirements that have been imposed in Section "PROBLEM DEFINITION." This is not an easy task, because the antenna is not a canonical object and there is not a simple formula that can be used to obtain the radiation parameters. Then, it is necessary to apply advanced numerical methods in computers to obtain the behavior of the antenna under test. These numerical methods are known as computational electromagnetics methods (CEM) [6] and they are applied to a variety of complex problems: antennas, radar cross section (RCS), propagation, radomes, electromagnetic compatibility, communications, etc.

When an electromagnetic signal impinges with an object or is used to feed a structure like an antenna, then a current is induced on it generating an electromagnetic field that is radiated in all the space. This is known as the scattering phenomena. This phenomena follows the Maxwell Equations [7] that relate the time domain and spatial variation of the electric and magnetic field generated by the currents. To solve these equations, several advanced numerical techniques were developed as can be seen in Figure 5.

Every technique has its own advantages and disadvantages, but the question is which method is used to solve this problem? Mainly, this depends on the size of the object compared with the frequency of operation. According to the main classification of Figure 5, there are basically two:

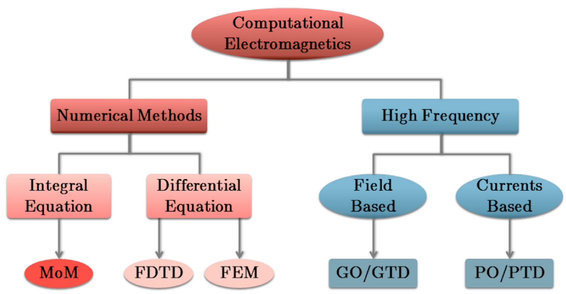


Figure 5.
Classification of numerical techniques.

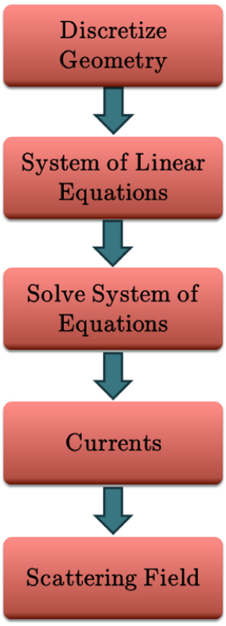


Figure 6.
Analysis of the antenna using the MOM technique.

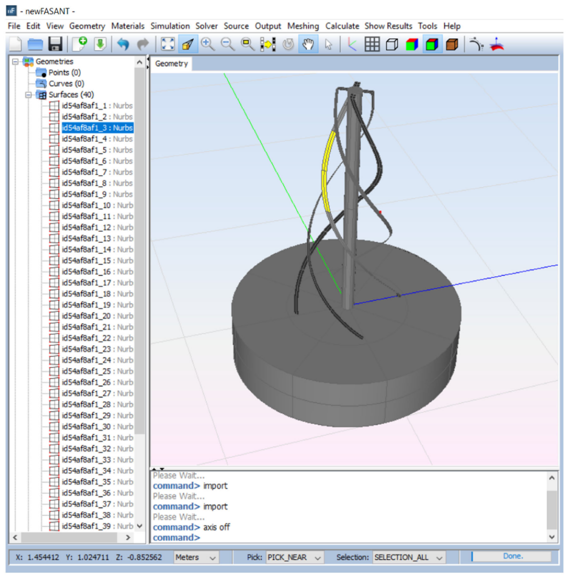


Figure 7.
Example of geometrical model of the antenna and a NURBS selected.

- High Frequency methods (Asymptotic Methods), where it is necessary that the object must be electrically large compared with the frequency (about several wavelengths). This is an approximation of the Maxwell equations, and in this case the current is assumed to be local character, that is, the current in one part of the structure is independent of other part, there is not coupling between structure parts. This is not the case, because at 2 GHz the wavelength is 0.15 m and the antenna size is close to this value. The techniques than can be used in this classification are Geometrical Optic (GO) [8] combined with the Geometrical Theory of Diffraction (GTD) [9] that are based on obtaining the Electrical Field of every ray that impinges the structure; and the Physical Optic (PO) [10]/Physical Theory of Diffraction (PTD) [11] that are based on calculating the currents on the object to obtain after the scattering field.
- Numerical Methods (Rigorous Methods), which does not matter the size of the object compared with the frequency, but they have the problem that when the frequency increases, more computation resources (memory and CPU) are needed. The main characteristic of the currents of the object is they are strongly coupled with other parts of the object. Mainly there are three techniques depending on the kind of Maxwell Equations they are using: Moment of Methods (MoM) [12] that solves the integral Maxwell Equations, and Finite Elements (FEM) [13], and Finite Difference Time Domain (FDTD) [14] that solves the differential partial Maxwell equations. These techniques can be applied to this case without any problem, and we have chosen MoM because this is the technique that the Electromagnetic Computing Group (GEC) [15] has been working to solve electromagnetic problems for more than 20 years.

When the object has arbitrary shape, there is not analytic solution as has been mentioned before. Then, a numerical technique must be applied to solve the problem. The MoM technique has to be selected and it is going to be applied to the analysis of the antenna. The process that is going to be applied to the analysis of the antenna is shown in Figure 6.

The geometrical model of the antenna is represented using Non-Uniform Rational B-Splines [16] (see Figure 7), a kind of parametric surfaces that are very common to model complex objects with a little information that are able to represent very accurately the real shape of the object, avoiding the use of planar facets models that are not so good when curved surfaces are presented. In this example of antenna, with only 40 NURBS surface is enough to represent the geometry.

The Maxwell equation cannot be applied directly to the geometrical model of the antenna, so a first step of discretization of the geometry must be done. This is done using a mesher [17], a code that preprocess and discretize the geometry obtaining small pieces of the surfaces named elements. The elements shall have a size lower than the wavelength, typically an edge size between wavelength/8 or wavelength/10 to obtain an accurate representation of the current. Thus, it can be seen that when the object of

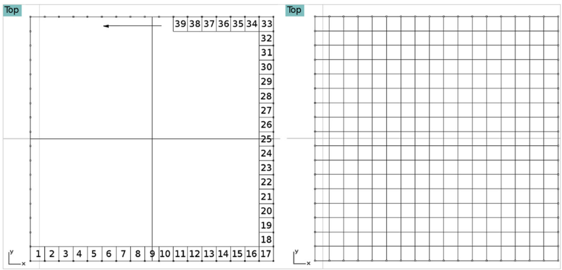


Figure 8.
Example of discretization procedure.

the size is bigger or the frequency is increased, more elements are needed and then more computational resources would be necessary. The discretization process is done according to the Paving Technique [17] that is based on dividing the edges of the surfaces according to the previous size and then fill the original surface with quadrangular elements from the edges to the inner part of the surface. Figure 8 shows an example of the Paving algorithm apply to a plane surface where 289 elements have been obtained.

The right image of Figure 8 is the mesh of the surface and it is very important that the mesh is continuous and the elements have more or less the same size and not be an irregular mesh. It is so important to have a good mesh as to apply an accurate numerical technique to obtain good results. Figure 9 shows the continuous mesh obtained with the mesher when it has been applied to the helix antenna.

When the elements have been obtained, it is necessary to define the basis functions that will model the current on the element, the amplitude of these basis functions will be the unknown current that will be necessary to calculate. The unknowns or subdomains will be defined by two elements that share a common edge. This common edge

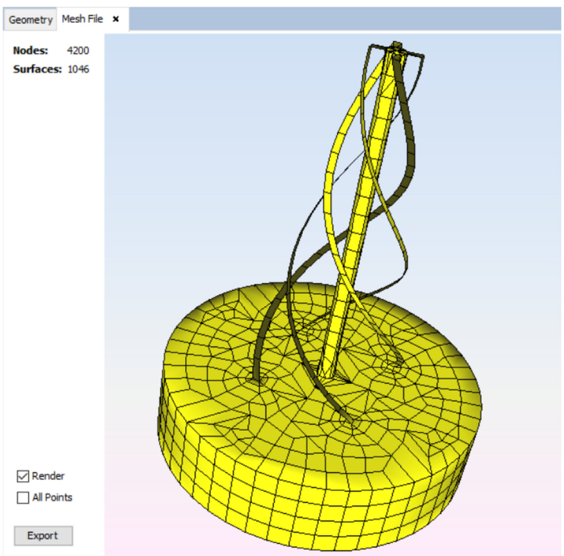


Figure 9.
Mesh of the helix antenna.

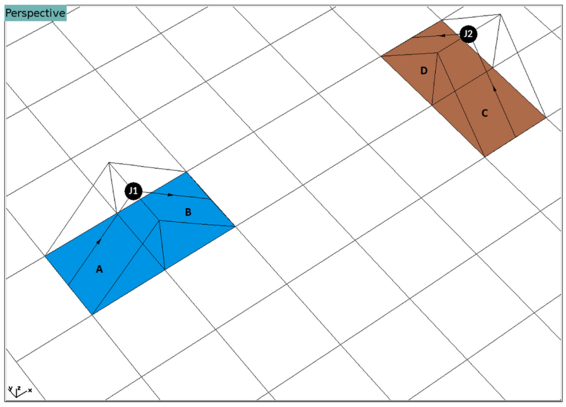


Figure 10.
Definition of the subdomains or current elements.

will have the maximum of the amplitude current and will be the part of the geometry where the Maxwell equation will be applied to build the System of Linear Equations. Thus, a basis function will be defined as a conformed rooftop that models the current that flows on the subdomain. The current starts from one edge of the first element, reach the maximum on the common edges of both elements and ends in the edge of the second element. Figure 10 shows the definition of the basis function between two elements that share a common edge. Elements A and B define a current element where the J1 amplitude of the current is unknown and the elements C and D define another current element where the amplitude J2 is unknown. The J1 is a current element according to X direction and J2 according Y. In both elements, it can be seen that the current is born in patch A or C and the current dies in element B or C depending on the subdomain considered.

Figure 11 shows all subdomains obtaining from the mesh of the helix antenna. In this example, there are 1929 subdomains or current elements where the Maxwell equation will be applied and a linear system of 1929 equations with 1929 unknowns must be solved to know the amplitude of the currents and then to be able to calculate the scattering field to obtain the radiation patterns.

The equation that must satisfy every subdomain of the antenna must be the Electric Field Integral Equation (EFIE) [7], [12] that has the following expression when the surfaces of the geometry are Perfect Electrical Conductors (PEC):

$$\hat{n} \times \vec{E}^{\text{imp}} = \left[\hat{n} \times \frac{j\omega\mu}{4\pi} \iint_S \vec{J}_S(\vec{r}')G(\vec{r},\vec{r}')dS' \right. \\ \left. + \frac{j}{4\pi\omega\epsilon} \nabla \iint_S \nabla' \cdot \iint_S \vec{J}_S(\vec{r}')G(\vec{r},\vec{r}')dS' \right] \tag{2}$$

where $\vec{E}^{\text{imp}}$ is the impressed electrical field (electrical field that feed the antenna), $\hat{n}$ is the normal vector on every surface point, $\vec{J}_S(\vec{r}')$ is the current density defined in every

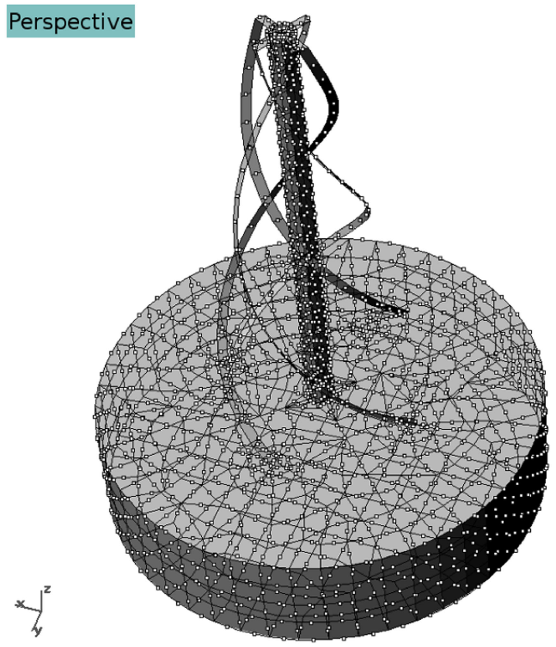


Figure 11.
Subdomains obtained from the mesh for the helix antenna.

subdomain, and $G(\vec{r}, \vec{r}')$ is the green function that gives an idea that how a subdomain $(\vec{r}')$ couples with other subdomain $(\vec{r})$. Each subdomain couples with the other subdomains and this defines a system of equations of this way

$$[Z][I] = [V] \tag{3}$$

where $[Z]$ is the coupling matrix of $N \times N$ elements, where every element is known, $[I]$ the unknown amplitudes of the currents, vector of N elements, and $[V]$ is the independent term, a vector of N elements that it is known and depends on the impressed electrical field. Then, the amplitude of the currents can be obtained solving the system with direct methods, for instance, inverting the matrix $[Z]$ or with iterative methods like Biconjugate Gradient Stabilized Method (BICGSTAB) [18] or Generalized Minimal Residual Method (GMRES) [19].

To feed the helix antenna, it is necessary to fix an impressed voltage in every helix with a difference of phase of 90°. Thus, in this way, the RHCP or LHCP can be obtained. Figure 12 shows the location of the impressed voltage between the bottom end of every helix and the top lid of the cylinder. In the right of the figure, the voltage values are assigned in this way.

When the system of equations has been solved, the currents can be visualized as in Figure 13. Red color shows where the amplitude is maximum. These currents will scatter the electrical and magnetical field of the antenna.

When the amplitudes of the currents are known, then all the scattering fields of the antenna can be obtaining in every part of the space, according to the following

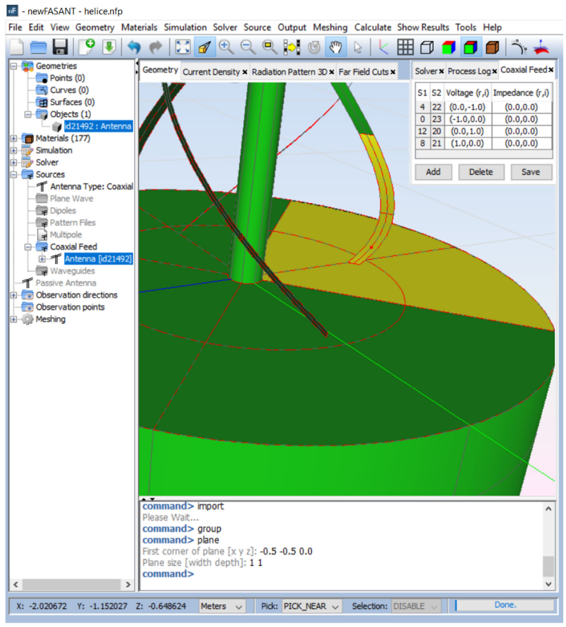


Figure 12.
Assign of the impressed voltage to every helix.

expression in far field region [7]:

$$\vec{E}^{\text{scatt}}(\vec{r}) = -\frac{j\omega\mu}{4\pi}\frac{e^{-jkr}}{r}\iint_S \vec{J}_S(\vec{r}')e^{-jk\hat{r}\cdot\vec{r}'}dS'. \tag{4}$$

Figure 14 shows the 3-D radiation pattern when the equatorial radiation pattern can be noticed.

MULTIOBJECTIVE OPTIMIZATION PROBLEMS

Multiobjective Optimization Problems (MOOP) are those that involve multiple and conflicting objective functions.

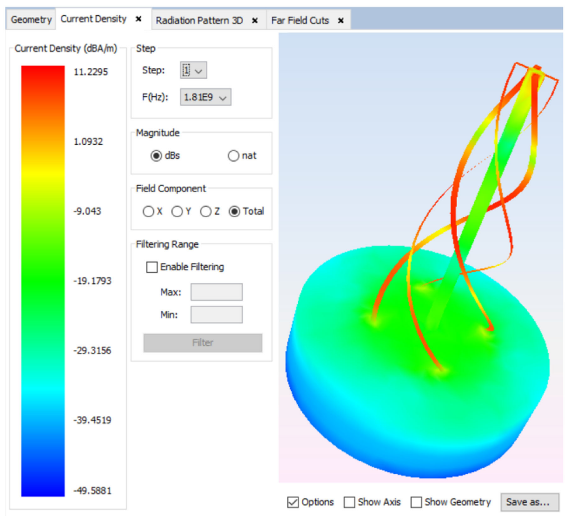


Figure 13.
Current distribution of the antenna.

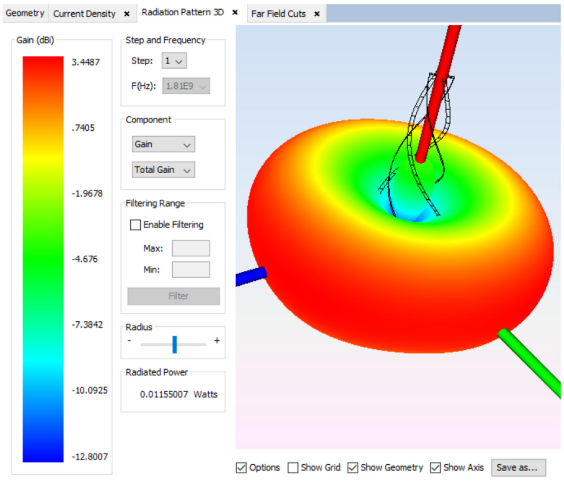


Figure 14.
3-D radiation pattern of the antenna.

In general, there are multiple valid solutions that are defined using the concept of *Pareto-optimal Front*. The Pareto-optimal Front is the set of the best possible solutions for the problem. In Section "PROBLEM DEFINITION," we defined the solution objectives s_o as a vector of n objective values, i.e., $s_o = (s_{o_1}, \dots, s_{o_n})$. To obtain the Pareto-optimal Front of a problem, it is necessary to sort all solutions according to their relationship of dominance. We say that a solution s dominates a solution v, denoted as $s \preceq v$ if the objective values of s are partially less (at least one less and equal the rest) than the objectives values of v, i.e., $\forall i \in (1, \dots, n),\ s_{o_i} \leq v_{o_i} \wedge \exists i \in (1, \dots, n) : s_{o_i} < v_{o_i}$. This definition considers that we are minimizing all objective values. To maximize, just change the less than operator by greater than operator. Note that the relationship of order $\preceq$ is partial and therefore there may be solutions that do not dominate each other. A set of solutions that do not dominate each other is said to belong to the same front. Those solutions that are not dominated by any other, belong to the first front, called Pareto-optimal Front. The solutions dominated by those belonging to the first front, but which do not dominate each other, form the second front. And so, successively, all the solutions are grouped in different fronts. To illustrate the previous concepts, let us provide an example with the problem that concerns us. In our case, we have four objectives to optimize, i.e., cross-polar polarization level (dB) and gain RHCP (dBi) for 1.81 and 2.55-GHz frequencies. Table 1 shows the objective values of six solutions obtained from the experimentation. Solutions 1 and 2 correspond to solutions SPEA-2 5 and *SPEA-2 6* shown in Table 2. Last column of the table shows the front to which each solution belongs. All solutions in the first front belong to the Pareto-optimal Front.

Note that solutions in the same front do not dominate each other, but they do dominate solutions in lower fronts. Solution 1, for example, dominates solution 2 for the 1.81-GHz values, but is dominated for the 2.55-GHz values by solution 2. Figure 15 shows a 4-D chart (the fourth axis is the color range) with values obtained from the experimentation (some of them are shown in Table 1). Solutions in the Pareto-optimal Front (front 1) correspond to the most top-left plane.

METAHEURISTICS AND EVOLUTIONARY ALGORITHMS

Metaheuristics are a family of approximate optimization techniques for solving the computational problem. There are multiple metaheuristic techniques available for solving MOOPs.

Evolutionary algorithms (EAs) are a set of algorithms inspired in the biologic evolution. Algorithm 1 shows the pseudocode of a standard EA. At each generation (loop iteration), an auxiliary population (with the same size as the original one) is generated by iteratively applying the genetic operators (crossover and mutation), then, both the current and the auxiliary populations are merged into one single new population. Worst individuals of the new

Table 1.

Objective Values of Six Solutions

	Cross Polar Level (dB)		Gain RHCP (dBi)		
Solution	**1.81 GHz**	**2.55 GHz**	**1.81 GHz**	**2.55 GHz**	**Front**
1	−19.13	−13.82	5.81	3.82	1
2	−12.87	−19.2	3.33	4.02	1
3	−9.2	−8.83	−1.7	−3.1	2
4	−5.9	−9.01	−5.2	−5.0	2
5	−1.3	−5.2	−8.2	−5.1	3
6	−3.9	−0.4	−5.3	−9.75	3

Table 2.

Parameters of Best Solutions Found by Using jMetal and MONURBS					
Algorithm	**Turns**	**Bottom radius (cm)**	**Top radius (cm)**	**Height (cm)**	**Volume (cm^3)**
Gradient Descent	0.831	1.945	1.022	13.8	98.49
NSGA-II 1	0.787	2.458	1.001	13.732	136.672
NSGA-II 2	0.787	2.339	1.02	13.732	127.915
NSGA-II 3	0.787	2.458	1.001	13.732	136.672
NSGA-II 4	0.791	2.458	1.001	14.505	144.346
NSGA-II 5	0.777	1.929	1.145	14.072	106.676
NSGA-II 6	0.787	2.339	1.02	13.732	127.915
NSGA-II 7	0.777	1.929	1.145	14.072	106.676
SPEA-2 1	0.859	2.409	0.701	14.433	120.677
SPEA-2 2	1.119	2.168	0.734	16.898	120.837
SPEA-2 3	0.782	2.385	0.709	14.859	122.676
SPEA-2 4	0.801	2.494	0.748	14.697	133.022
SPEA-2 5	0.84	1.032	0.753	13.593	**34.281**
SPEA-2 6	0.637	1.092	0.72	13.275	34.701
SPEA-2 7	0.637	1.344	0.704	13.145	44.739

population are removed (i.e., the best solutions are selected), until the size of the new population is reduced by half. For us, each individual is a problem solution, so individual and solution are equivalent concepts.

EAs are particularly desirable to solve MOOPs, primarily because of their population-based nature. This enables them to capture the dominance relations in the population as a way to guide the search toward the Pareto-optimal Front.

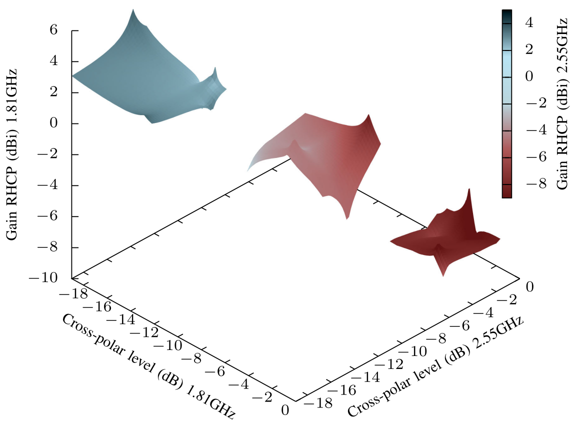


Figure 15.
Four-dimensional representation of solutions obtained in the experimentation.

EAs usually contain several parameters that need to be tuned for each particular application at the same time considering:

(1) Nonconflicting objectives, i.e., achieve a single optimal solution satisfies all objectives simultaneously.

(2) Competing objectives, i.e., cannot be optimized simultaneously.

In addition, since the EAs are stochastic optimization techniques, different runs tend to produce different results. Therefore, multiple runs of the same algorithm on a given problem are needed to statistically describe their performance on that problem. For a more detailed discussion of the application of EAs in multiobjective optimization, the reader is referred to Coello et al. [20] and Deb et al. [2]. Multiobjective EAs need to fulfill two primary roles:

(1) Guiding the search toward the Pareto-optimal Front set to accomplish optimal or near-optimized solutions.

(2) Maintaining a diverse population to achieve a well-distributed nondominated front, thereby fully exploring the solution space.

Algorithm 1: Evolutionary Algorithm

```
Create random initial population P
Evaluate population P
while Stopping criteria not reached do
    Select population P
    Create empty population Q
    for Population size/2 times do
        Select two parents from P
        Perform crossover & Mutation
        Insert children into Q
    end for
    Evaluate population Q
    P ← Q
end while
return P
```

THE NONDOMINATED SORTING GENETIC ALGORITHM-II (NSGA-II)

This algorithm was developed by Deb et al. [2] as an extension of an earlier proposal by Srinivas and Deb [21].

The population individuals (solutions) are evaluated (i.e., they are assigned fitness values) in relation to how close they are to the *Pareto-optimal Front* and a *crowding* measure.

The NSGA-II algorithm also considers the sparsity (density) of the individuals belonging to the same rank using a crowding measure (the Manhattan distance among individuals), with the idea of promoting diversity within the fronts (the larger the sparsity, the better). In addition, the NSGA-II includes elitism in order to maintain the best solutions from the *Pareto-optimal Front* found. The rank of each individual is based on the level of nondomination. Therefore, each solution has two attributes: i) nondomination rank (front to which the solution belongs) and ii) crowding distance. In other words, between two solutions with differing nondomination ranks, the solution with the lower rank is preferred. Otherwise, if both solutions belong to the same front, then the solution that is located in a less crowded region is preferred. The pseudocode of NSGA-II is shown in Algorithm 2. Note that lines 2 and 6 are calls to the simulator to evaluate the solution parameters. NSGA-II sends the parameters of the solution to the simulator and the simulator returns the objective values for that solution.

STRENGTH PARETO EVOLUTIONARY ALGORITHM-2 (SPEA-2)

This algorithm was proposed by Zitzler et al. [3]. In this algorithm, the strength of an individual is defined in terms of the number of solutions it dominates in the population. A fitness value is assigned to every individual, and it is defined as the sum of its strength raw fitness and a density estimation. The algorithm evolves the population through the iterative application of the variation operators on the solutions. All generated nondominated solutions are stored in an external archive. After every iteration, all nondominated solutions (from both the population and the archive) are copied into a new generation population. If its size is larger than the population size, the algorithm applies a truncation operator to discard solutions. It is based on the distances to the k*th* nearest neighbors (a crowding measure), so that those solutions having the largest distances to the other solutions (i.e., the most isolated ones) are selected. Algorithm 3 presents a pseudocode of SPEA-2. Note that lines 4 and 5 are calls to the simulator to evaluate each solution parameters. SPEA-2 sends the parameters of the solution to the simulator and the simulator returns the objective values for that solution.

DESIGNING THE ANTENNA

In order to obtain the radiation patterns of the antenna shown in Figure 2, it is necessary to use a simulation computer program with the input of the four parameters of the antenna, then does:

(1) builds the geometrical model of the antenna;

(2) discretize the model according the wavelength;

(3) simulates the antenna to obtain the radiation patterns for both frequencies in order to be processed by the multiobjective algorithms.

Figure 16 shows the block diagram of the electromagnetic simulation stage.

The geometrical model is built using a software by which the previously stated parameters can create a geometrical file in AutoCAD DXF (Drawing Interchange Format) [22] with the surfaces that define the antenna. This file must be previously processed to be simulated with a mesher that discretize the antenna parameters as input to the simulator that calculates the radiation pattern using a simulation software called MONURBS [1].

This simulation software is being developed by the Electromagnetic Computing Group, University of Alcalá, and it is included in as part of an electromagnetic suite, newFASANT [23]. This suite can be used in many applications like electromagnetic field analysis of any complex 3-D structures such as reflectors, horns, microstrip passive devices, periodical structures, antenna on board, etc. Also, the RCS of complex platforms with arbitrary materials and the compatibility between different devices mounted on the same platform. Figure 17 shows the User Interface of the code with all the modules that can be used for several electromagnetic applications.

From this suite, the MONURBS and mesher code have been extracted and used as a black box to implement the procedure of Figure 16. The MONURBS code is based on the Moment Method Technique (MoM) that is a full-wave solution. When the object to be analyzed is large, this technique is both CPU and memory consuming and cannot

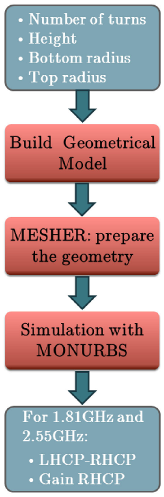


Figure 16.
Electromagnetic simulation stage block diagram.

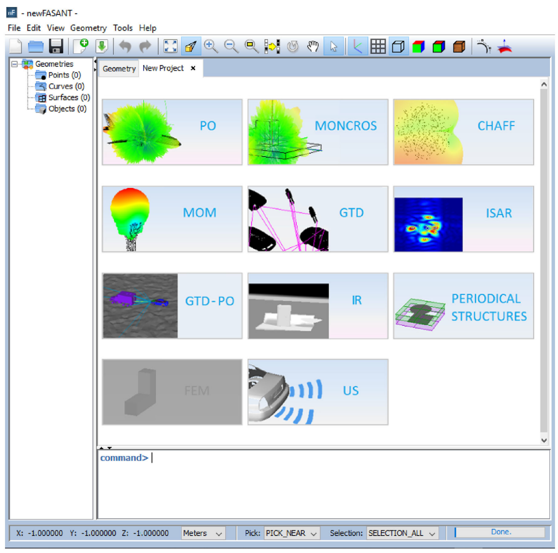


Figure 17.
Electromagnetic suite newFASANT.

be applied if the resources of the machine are not high. To overcome this, several techniques have been implemented to speed up the simulation while using less memory: (i) Fast Multipole Multilevel Method [12], [24] and (ii) the Characteristics Basis Function Method [25], [26]. Also, the Message Passing Interface (MPI) and OpenMP paradigms have also been implemented to solve the problem using less CPU time with multiprocessor machines [27].

LOOKING FOR THE OPTIMAL PARAMETERS

In this paper, we used the implementation of NSGA-II and SPEA-2 provided in the jMetal[2] framework [28] for

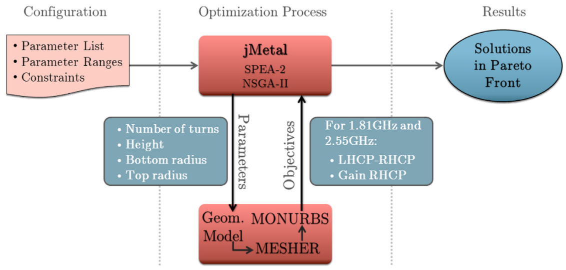


Figure 18.
Antenna parameters optimization.

multiobjective optimization together with a simulation software of antenna radiation, MONURBS, as previously described.

Algorithm 2: NSGA-II Algorithm [2]

1: $P \leftarrow$ **makeInitalRandomPopulation**()
2: $P \leftarrow$ **antennaSimulator**(P) ▷ Call to evaluate P
3: $t \leftarrow 0$
4: **while** $t \leq max\_generations$ **do**
5: $Q \leftarrow$ **makeNewPopulation**(P)
6: $Q \leftarrow$ **antennaSimulator**(Q) ▷ Call to evaluate Q
7: $R \leftarrow P \cup Q$ ▷ Combine parents and offsprings
8: $\mathcal{F} \leftarrow$**fastNonDominatedSort**(R) ▷ Calculate *Fronts*
9: $P \leftarrow \emptyset \wedge i \leftarrow 1$
10: **while** $|P| + |\mathcal{F}_i| \leq N$ **do**
11: $P \leftarrow P \cup F_i$ ▷ Add ith rank to population
12: $i \leftarrow i + 1$
13: **end while**
14: **if** $|P| \neq N$ **then**
15: **crowdingDistance**($\mathcal{F}_i$) ▷ Calc. crowding measure in $\mathcal{F}_i$
16: $P \leftarrow P \cup$ **bestCrowdingSolutions**($\mathcal{F}_i, |P| - N$)
17: **end if**
18: $t \leftarrow t + 1$
19: **end while**
20: $\mathcal{F} \leftarrow$ **fastNonDominatedSort**(R)
21: **return** $\mathcal{F}_1$ ▷ Return first front; i.e., *Pareto-optimal Front*

As jMetal is being developed in Java, the communication is also handled using the Java runtime API to simulate the antenna radiation using the antenna parameters generated by the multiobjective algorithms. Therefore, to perform the data exchange between jMetal and MONURBS, it was necessary to implement a specific method, called *AntennaSimulation(P)* (Algorithm 4) to perform the population evaluation. The calls are carried out in lines 2 and 6 in the NSGA-II (Algorithm 2), and lines 4 and 5 in the SPEA-2 (Algorithm 3). Figure 18 illustrates the communication between jMetal and MONURBS implemented in Algorithm 4.

[2] https://github.com/jMetal/

Algorithm 3: SPEA-2 Algorithm [3]

$P_0 \leftarrow$ **makeInitalRandomPopulation**()
$Q_0 \leftarrow \emptyset$ ▷ Initial empty archive of size M
while $t \leq max\_generations$ **do**
  $P_t \leftarrow$ **antennaSimulator**(P_t) ▷ Call to evaluate P_t
  $Q_t \leftarrow$ **antennaSimulator**(Q_t) ▷ Call to evaluate Q_t
  $Q_{t+1} \leftarrow$ **copyNonDominatedSolutions**(P_t, Q_t)
  if $|Q_{t+1}| > M$ **then** ▷ Q_{t+1} exceeds archive size
    $Q_{t+1} \leftarrow$ **truncate**(Q_{t+1})
  else if $|Q_{t+1}| < M$ **then** ▷ Fills with dominated solutions
    $Q_{t+1} \leftarrow$ **copyDominatedSolutions** $(P_t, Q_t, M - |Q_{t+1}|)$
  end if
  $P_{t+1} \leftarrow$ **selectPopulation**(Q_{t+1}) ▷ Mating selection
  $P_{t+1} \leftarrow$ **variationOperators**(P_{t+1}) ▷ Apply recombination and mutation
  $t \leftarrow t + 1$
end while
$\mathcal{F} \leftarrow$ **copyNonDominatedSolutions**(Q)
return $\mathcal{F}$ ▷ Return the *Pareto-optimal Front*

Algorithm 4: antennaSimulator(P)

for each solution s in population P **do**
  if s does not violates problem constraints **then**
    Invoke MONURBS process with s parameters (s_p vector)
    Wait until MONURBS process finalizes
    Parse MONURBS return
    Update s evaluation values (s_o vector)
  end if
end for
return P

RESULTS

In this section, we show solutions found by both i) using MONURBS (as standalone tool using its built-in *GD* optimizer) and ii) combining MONURBS and jMetal. The computer and software used to carry out the experimentation were:

- Windows Server 2016 Standard, 64 bits.
- Java version: 1.8.0-121, 64 bits.
- 8 Quad-Core AMD Opteron Processor 8356 2.29 GHz.
- 256 GB of RAM memory.

The configuration of the problem ranges, i.e., the antenna parameters was

- Number of turns: [0.2, 3].
- Bottom radius: [0.1 cm, 50 cm].

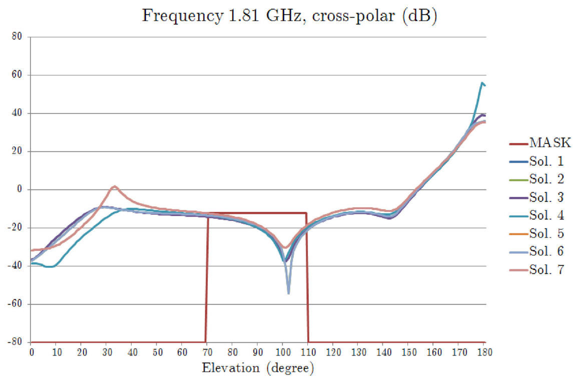


Figure 19.
1.81-GHz cross-polar objective results using NSGA-II.

- Top radius: [0.1 cm, 50 cm].
- Height: [0.1 cm, 50 cm].

Also, the configuration of NSGA-II and SPEA-2 was as follows:

- Population size: 50.
- Maximum number of algorithm iterations: 5000
- Crossover operator: Simulated binary crossover
  - Crossover probability: 90%
  - Crossover distribution index: 20.
- Mutation operator: Polynomial mutation
  - Mutation distribution index: 20
  - Mutation probability: 25%.

The results are shown in Table 2. The first column shows the solutions found by each algorithm. Four next columns show the parameters defined by each solution and the last column shows the volume of the antenna. All results where rounded to three decimals. As it can be observed, all solutions found are very close to each other, especially those obtained by NSGA-II where some of them are practically equivalent. The first row shows the result obtained with the MONURBS *GD* in order to compare such results with the ones obtained by the NSGA-II and SPEA-2 algorithms in the next rows. Figures 19 to 26 show the objective values graphically. It can be observed that all solutions met the constraints defined for this problem. The results obtained by SPEA-2 are better dispersed than those obtained by NSGA-II. Please note that the best solution obtained by SPEA-2 reduces the volume of the solution obtained using GD by 35%. Figure 27 shows the geometrical model of this solution. With this new approach, the CPU time to obtain a suitable solution has been reduced considerably. The solution obtained with GD took several months, while using NSGA-II or SPEA-2 have taken three weeks. In a future work, we will need to apply other multiobjective algorithms and techniques to explore if there are other parameters that are significantly different.

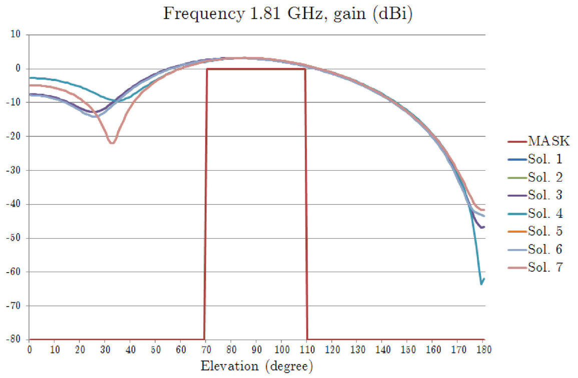


Figure 20.
1.81-GHz gain objective results using NSGA-II.

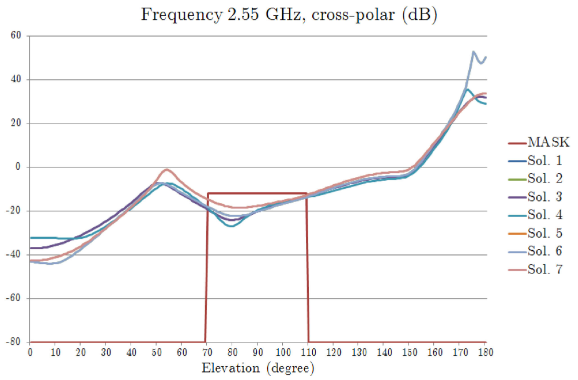


Figure 21.
2.55-GHz cross-polar objective results using NSGA-II.

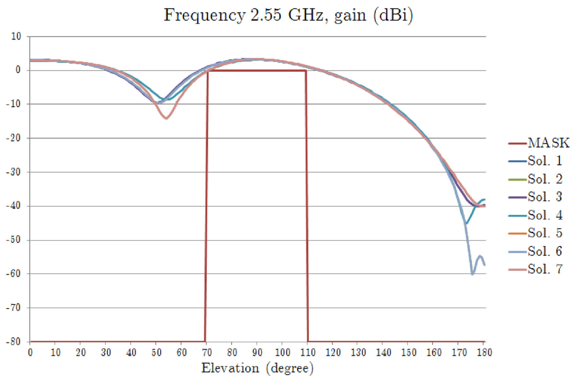


Figure 22.
2.55-GHz gain objective results using NSGA-II.

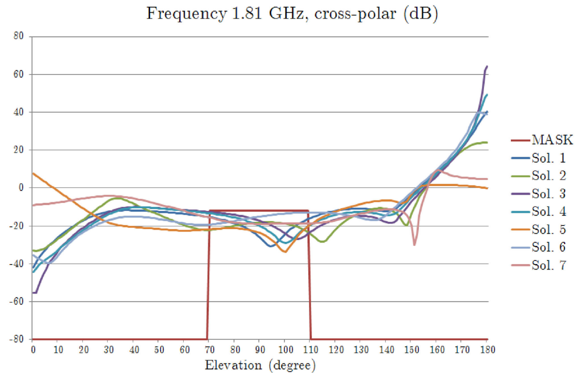


Figure 23.
1.81-GHz cross-polar objective results using SPEA-2.

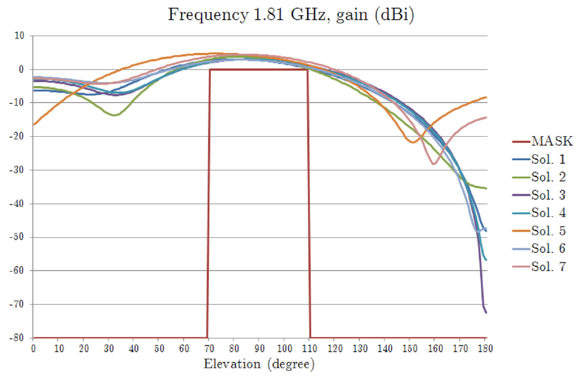


Figure 24.
1.81-GHz gain objective results using SPEA-2.

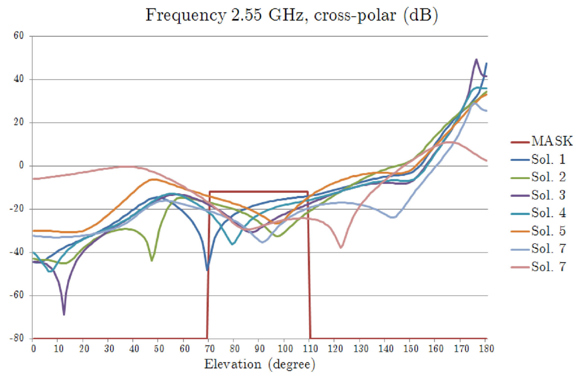


Figure 25.
2.55-GHz cross-polar objective results using SPEA-2.

CONCLUSIONS AND FUTURE WORK

In this paper, we presented a simulation–optimization approach to the design of helical antennas. This is a very complex problem with several restrictions that must be met in two frequencies and, additionally, compacting the antenna dimensions as possible. To address the problem, we used two well-known multiobjective algorithms and still state-of-the-art algorithms, NSGA-II and SPEA-2, that were capable of improving the time and effort needed to find valid solutions (antenna shape and dimensions) when compared with finding solutions using the GD as a searching technique together a simulator tool. The use of multiobjective algorithms reduced the time cost of algorithm execution when compared with a previous approach using the GD. Also, the simulation–optimization approach allow us to obtain multiple correct solutions that provide some flexibility and can help us to choose the final design of the antenna. Having more solutions, with different dimensions but all optimal from the radiation point of view, offers more

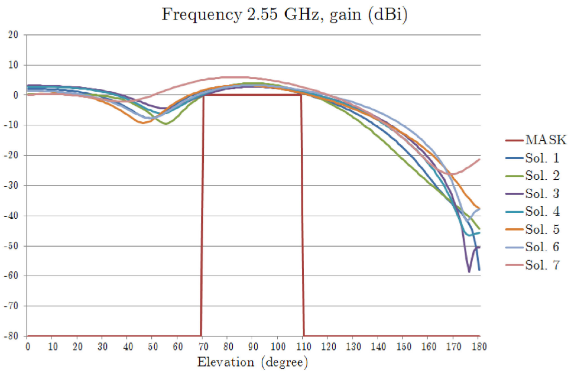


Figure 26.
2.55-GHz gain objective results using SPEA-2.

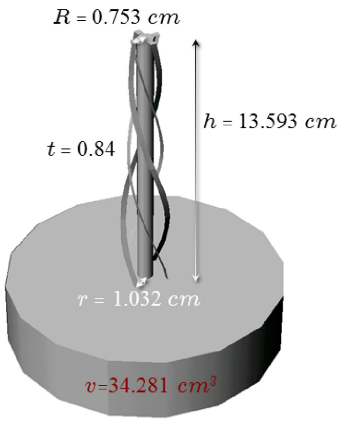


Figure 27.
Geometrical model of the solution with a lower volume (SPEA-2 5).

possibilities for the manufacturing not only for the antenna but the rest of elements that are coupled closely to it.

Future works include the use other multiobjective algorithms capable of handling the constrains to compare and adapt them to the difficulty of this problem. We will also explore many-objective algorithms as we are handling five objectives in this paper.

ACKNOWLEDGMENT

This work was supported by the University of Alcalá, Projects BadgePeople (TIN2016-76956-C3-3-R), and the Spanish Department of Science, Innovation and Universities, Project TEC2017-89456-R.

REFERENCES

[1] I. González, E. Garcia, F. Sáez de Adana, and M. F. Cátedra, "Monurbs: A parallelized multipole multilevel code for analyzing complex bodies modeled by nurbs surfaces," *Appl. Comput. Electromagn. Soc. J.*, vol. 23, no. 2, pp. 134–142, 2008.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[3] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Comput. Eng. Netw. Lab., Swiss Federal Inst. Technol., Zürich, Switzerland, Tech. Rep. 103, 2001.

[4] I. González, J. Gómez, A. Tayebi, and M. F. Cátedra, "Optimization of a dual-band helical antenna for TTC applications at s band," *IEEE Antennas Propag. Mag.*, vol. 54, no. 4, pp. 63–77, Aug. 2012.

[5] J. Moreno, I. Gonzalez, and D. Rodriguez, "Using simulation and the NSGA-II evolutionary multi-objective algorithm in the design of a compact dual-band equatorial helix antenna," in *Proc. 6th Int. Conf. Space Mission Challenges Inf. Technol.*, Sep. 2017, pp. 56–60.

[6] P. Sumithra and D. Thiripurasundari, "A review on computational electromagnetics methods," *Adv. Electromagn.*, vol. 6, no. 1, pp. 42–55, 2017.

[7] C. A. Balanis, *Advanced Engineering Electromagnetics*. Hoboken, NJ, USA: Wiley, 1989.

[8] R. G. Kouyoumjian, "Asymptotic high-frequency methods," *Proc. IEEE*, vol. 53, no. 8, pp. 864–876, Aug. 1965.

[9] R. G. Kouyoumjian and P. H. Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface," *Proc. IEEE*, vol. 62, no. 11, pp. 1448–1461, Nov. 1974.

[10] J. A. Stratton, *Electromagnetic Theory*. New York, NY, USA: MacGraw-Hill, 1941.

[11] P. Y. Ufimtsev, "Elementary edge waves and the physical theory of diffraction," *Electromagnetics*, vol. 11, no. 2, pp. 125–160, 1991.

[12] W. Chew, E. Michielssen, J. M. Song, and J. M. Jin, Eds., *Fast and Efficient Algorithms in Computational Electromagnetics*. Norwood, MA, USA: Artech House, 2001.

[13] O. L. Zienkiewichhz, *The Finite Element Method*. New York, USA: MacGraw-Hill, 1979.

[14] A. Taflove, "Application of the finite-difference time-domain method to sinusoidal steady-state electromagnetic-penetration problems," *IEEE Trans. Electromagn. Compat.*, vol. EMC-22, no. 3, pp. 191–202, Aug. 1980.

[15] "Computational-electromagnetics research group." 2018. [Online]. Available: https://www.uah.es/en/investigacion/unidades-de-investigacion/grupos-de investigacion/Computacional-electromagnetics/#Miembros.

[16] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. New York, NY, USA: Academic, 1988.

[17] J. M. Garrido, M. A. Díaz, I. G. Diego, and M. F. Cátedra Pérez, "Design and evaluation of the multilevel mesh generation mode for computational electromagnetics," *Appl. Comput. Electromagn. Soc. J.*, vol. 30, no. 6, pp. 578–588, 2015.

[18] H. A. van der Vorst, “Bi-CGSTAB: A fast and smoothly converging variant of bi-CG for the solution of nonsymmetric linear systems,” *SIAM J. Sci. Statist. Comput.*, vol. 13, no. 2, pp. 631–644, Mar. 1992. [Online]. Available: http://dx.doi.org/10.1137/0913035

[19] Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems,” *SIAM J. Sci. Statist. Comput.*, vol. 7, no. 3, pp. 856–869, Jul. 1986. [Online]. Available: http://dx.doi.org/10.1137/0907058

[20] C. A. C. Coello, “A comprehensive survey of evolutionary-based multiobjective optimization techniques,” *Knowl. Inf. Syst.*, vol. 1, pp 269–308, 1998.

[21] N. Srinivas and K. Deb, “Muiltiobjective optimization using nondominated sorting in genetic algorithms,” *Evol. Comput.*, vol. 2, pp. 221–248, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1162/evco.1994.2.3.221

[22] “Dxf reference,” 2012. [Online]. Available: http://images.autodesk.com/adsk/files/autocad_2012_pdf_dxf-reference_en%u.pdf

[23] “Fasant.” 2018. [Online]. Available: http://www.fasant.com

[24] E. García, C. Delgado, I. G. Diego, and M. F. Cátedra, “An iterative solution for electrically large problems combining the characteristic basis function method and the multilevel fast multipole algorithm,” *IEEE Trans. Antennas Propag.*, vol. 56, no. 8, pp. 2363–2371, Aug. 2008.

[25] C. Delgado, M. Cátedra, and R. Mittra, “Application of the characteristic basis function method utilizing a class of basis and testing functions defined on nurbs patches,” *IEEE Trans. Antennas Propag.*, vol. 56, no. 3, pp. 784–791, Mar. 2008.

[26] C. Delgado, M. F. Cátedra, and R. Mittra, “Efficient multilevel approach for the generation of characteristic basis functions for large scatters,” *IEEE Trans. Antennas Propag.*, vol. 56, no. 7, pp. 2134–2137, Jul. 2008.

[27] “MPI and OpenMP introduction,” 2018. [Online]. Available: http://www.hpc.cineca.it/content/introduction-parallel-computing-mpi-an% d-openmp

[28] A. J. Nebro, J. J. Durillo, and M. Vergne, “Redesigning the jmetal multi-objective optimization framework,” in *Proc. Companion Publication Annu. Conf. Genetic Evol. Comput.*, 2015, pp. 1093–1100. [Online]. Available: http://doi.acm.org/10.1145/2739482.2768462

Chapter 7

Merge Non-Dominated Sorting Algorithm for Many-Objective Optimization

Many Pareto-based multi-objective evolutionary algorithms require to rank the solutions of the population in each iteration according to the dominance principle, what can become a costly operation particularly in the case of dealing with many-objective optimization problems. In this paper, we present a new efficient algorithm for computing the non-dominated sorting procedure, called Merge Non-Dominated Sorting (MNDS), which has a best computational complexity of $\Theta(NlogN)$ and a worst computational complexity of $\Theta(MN^2)$. Our approach is based on the computation of the dominance set of each solution by taking advantage of the characteristics of the merge sort algorithm. We compare the MNDS against four well-known techniques that can be considered as the state-of-the-art. The results indicate that the MNDS algorithm outperforms the other techniques in terms of number of comparisons as well as the total running time.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Merge Nondominated Sorting Algorithm for Many-Objective Optimization

Javier Moreno, Daniel Rodriguez, *Member, IEEE*, Antonio J. Nebro, and Jose A. Lozano, *Senior Member, IEEE*

***Abstract*—Many Pareto-based multiobjective evolutionary algorithms require ranking the solutions of the population in each iteration according to the dominance principle, which can become a costly operation particularly in the case of dealing with many-objective optimization problems. In this article, we present a new efficient algorithm for computing the nondominated sorting procedure, called merge nondominated sorting (MNDS), which has a best computational complexity of $O(N \log N)$ and a worst computational complexity of $O(MN^2)$, with N being the population size and M being the number of objectives. Our approach is based on the computation of the *dominance set*, that is, for each solution, the set of solutions that dominate it, by taking advantage of the characteristics of the merge sort algorithm. We compare MNDS against six well-known techniques that can be considered as the state-of-the-art. The results indicate that the MNDS algorithm outperforms the other techniques in terms of the number of comparisons as well as the total running time.**

***Index Terms*—Evolutionary algorithms (EAs), many-objective problems, multiobjective optimization, nondominated sorting.**

I. Introduction

Evolutionary algorithms (EAs) have been successfully applied in the solution of multiobjective optimization problems (MOPs) in the last two decades. These approaches can be mainly classified into indicator-based, decomposition-based, and Pareto-based EAs. Indicator-based EAs are characterized by guiding the search by a quality indicator such as the hypervolume that measures the performance of solutions. In the second group, the idea is to decompose an MOP into a set of related single-objective subproblems that are optimized at the same time. Most of the algorithms belonging to the Pareto-based group, which includes the nondominated sorting genetic algorithm (NSGA)-II [1], SPEA2 [2], and many others [3], require ranking the population in the selection and replacement phases according to the dominance principle [4]. In this article, we focus on this third group of techniques, specifically on the efficient implementation of the nondominance sorting.

The nondominated ranking procedure can be computationally significant in the total computing time of a multiobjective EA (MOEA), particularly when dealing with many-objective problems, and large populations.

In this article, we present the merge nondominated sorting (MNDS) algorithm aimed at efficiently performing the nondominated ranking. MNDS takes advantage of the characteristics of the merge sort algorithm to calculate the *dominance set*, that is, the set of solutions that dominate other solutions, for each solution. MNDS achieves the best computational complexity of $O(N \log N)$, while the worst case is $O(MN^2)$, where N corresponds to the population size and M is the number of objectives. As usually happens with these kinds of algorithms, there is a time versus memory tradeoff. In our algorithm, the storage of the *dominance set* of each solution allows a reduction of the computational time.

The remainder of this article is organized as follows. Section II briefly presents current works aiming to reduce the computational cost of the nondominated sorting problem. Section III describes our proposal in detail. Experimental work and results are provided in Section IV. Finally, Section V highlights the conclusions and outlines future work.

II. Background and Related Work

Nondominated sorting is based on the concept of Pareto dominance between vectors (or solutions, in the context of EAs). Let P be a population of N solutions, $\{s_1, \ldots, s_N\} \in P$, where each solution contains a vector of M objectives to minimize, $(f_1(s_i), \ldots, f_M(s_i))$ $\forall i \in \{1, \ldots, N\}$. A solution s_i dominates solution s_j, denoted by $s_i \preceq s_j$, if the vector of objectives of s_i is partially less than the vector of objectives of s_j, that is, $\forall m \in \{1, \ldots, M\}, f_m(s_i) \leq f_m(s_j) \wedge \exists m' \in$

Manuscript received March 3, 2019; revised September 11, 2019 and December 13, 2019; accepted January 14, 2020. The work of Javier Moreno was supported by the University of Alcala. The work of Daniel Rodriguez was supported by the University of Alcala under Project TIN2016-76956-C3-3-R, Project uDEVOPS 823992, and Project RED2018-102472-T. The work of Antonio J. Nebro was supported by the Spanish Ministry of Education and Science under Grant TIN2017-86049-R. The work of Jose A. Lozano was supported in part by the Basque Government through the BERC 2018–2021 program and Research Group under Grant IT1244-19, and in part by the Spanish Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation under Grant SEV-2017-0718 and Grant TIN2016-78365-R. This article was recommended by Associate Editor Q. Zhang. *(Corresponding author: Javier Moreno.)*

Javier Moreno and Daniel Rodriguez are with the Department of Computer Science, University of Alcala, 28805 Alcala de Henares, Spain (e-mail: javier.morenom@edu.uah.es; daniel.rodriguezg@uah.es).

Antonio J. Nebro is with the Department of Computer Languages and Computing Sciences, Institute for Software Technologies and Software Engineering, Biomedical Research Institute of Malaga, University of Malaga ETSI Informática, Campus de Teatinos, 29071, Maálaga, Spain (e-mail: ajnebro@uma.es).

Jose A. Lozano is with the Department of Computer Science and Artificial Intelligence, University of the Basque Country, 20018 Donostia-San Sebastian, Spain, and also with the Basque Centre for Applied Mathematics, 48009 Bilbao, Spain (e-mail: ja.lozano@ehu.eus).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2020.2968301

TABLE I
COMPLEXITY OF NONDOMINATED SORTING ALGORITHMS REPRESENTATIVE OF THE STATE-OF-THE-ART

Algorithm	Complexity		
	Best Case	Worst Case	Space
FNDS [1]	MN^2	MN^2	N^2
Dominance Tree [6]	$MNlogN$	MN^2	MN
Deductive Sort [7]	$MN\sqrt{N}$	MN^2	N
Corner Sort [8]	$MN\sqrt{N}$	MN^2	N
ENS-SS [9]	$MN\sqrt{N}$	MN^2	1
ENS-BS [9]	$MNlogN$	MN^2	1
ENS-NDT [10]	$MNlogN$, *if* $M > logN$ $Nlog^2N$, *rest of cases*	MN^2	$NlogN$
M-Front [11]	MN	MN^2	MN^2
DDA-NS [12]	MN^2	MN^2	N^2
HNDS [13]	$MN\sqrt{N}$	MN^2	N
BOS [3]	$MNlogN$	MN^2	MN
MNDS	$NlogN$	MN^2	N^2

$\{1, \ldots, M\}$ s.t. $f_{m'}(s_i) < f_{m'}(s_j)$ (we assume minimization without loss of generality). Given a set of solutions, those solutions which are nondominated by any other are assigned rank 1. If these solutions are removed, then those solutions, which are nondominated by any other, are assigned rank 2, and so on. Finding these ranks is called nondominated sorting. Kung *et al.* [5] were the first to propose a method based on the divide-and-conquer idea to find maximal elements of a set of vectors, paving the way for further studies.

Reducing the complexity of nondominated sorting is a matter of active research. The original implementation of NSGA [4] had a complexity of $O(MN^3)$. A later version, in NSGA-II [1], the fast nondominated sorting (FNDS) reduced the cost to $O(MN^2)$. Table I shows both computational and spatial costs of the most representative algorithms for nondominated sorting and how they compare against the two variants of our current proposal (MNDS).

We briefly summarize next the different strategies used by each of these algorithms (they are fully described in the provided references).

1) *FNDS* [1] compares each solution with the rest of the solutions of the population to obtain their dominance relationship. While carrying out this comparison, each solution stores those solutions that it dominates in a list. Once the comparisons are done, the lists of dominated solutions are traversed to rank them.
2) *Dominance Tree* [6] uses a divide-and-conquer strategy to obtain the dominance relationships among the population solutions. These relationships are stored in a tree-like data structure called the dominance tree.
3) *Deductive Sort* [7] iterates through the population repeatedly, comparing the solutions one by one. Nondominated solutions are assigned to the corresponding rank and eliminated from the population.
4) *Corner Sort* [8] reduces the number of comparisons using two strategies: 1) as *Deductive Sort*, it avoids comparing solutions marked as dominated and 2) shows a preference for comparing corner solutions when determining the dominance between solutions.
5) Efficient nondominated sort (ENS) [9] calculates the rank of each solution at a time. To do so, it sorts the first objective using the lexicographical comparison.[1] Then, it looks for the rank of each solution using a sequential search strategy (version ENS-SS) or a binary search (version ENS-BS).
6) *M-Front* [11] proposes modifying the typical MOEA's structure to improve its performance. In order to reduce the number of comparisons among solutions, the M-Front algorithm applies the geometric and algebraic properties of the Pareto dominance to perform interval queries using a nearest-neighbor search. M-Front defines a special data structure called *archive* where all nondominated individuals are stored. In addition, M-Front stores all solutions in lists and uses a K-d tree for the nearest-neighbor search.
7) Hierarchical nondominated sorting (HNDS) [13] minimizes the number of comparisons of objectives by ordering the population by the first objective and then by comparing the first solution with the rest of the solutions. These solutions are moved to an auxiliary list if they are not dominated by the first solution or a list of dominated solutions otherwise. The first solution is assigned to its corresponding rank and then the algorithm iterates until all of the solutions are assigned their corresponding rank.
8) The Dominance degree approach for nondominated sorting (DDA-NS) [12] is based on the concept of the dominance degree matrix, which is an $N \times N$ square matrix where each column and row represents a solution $s_i : i \in \{1, \ldots, N\}$ and the cells contain the number of objectives in which each solution s_i dominates the other solution s_j $\forall s_i \neq s_j$ with $s_i, s_j \in P$. Once the dominance degree matrix is obtained, DDA-NS traverses the matrix to find the maximum values of each column, gets their corresponding solution, and assigns a rank to it.
9) Best order sort (BOS) [3] sorts the population by each objective, resolving ties by means of lexicographical comparison. For each objective and solution s_i, it searches those solutions that are not worse than s_i. These solutions are stored in a set T associated with s_i. BOS will look at T for the s_j solution with the worst rank r. The rank of s_i will be $r + 1$.
10) ENS with nondominated tree (ENS-NDT) [10] extends the ENS-BS [9] algorithm using a new data structure, a variant of a bucket k-d tree, called nondominated tree (NDTree). ENS-NDT is similar to ENS-BS but in the binary search, it uses an NDTree instead of an array to store the fronts, speeding up the domination checking.

We must note that there are also some approaches related to the efficient computation of nondominated sorting in steady-state MOEAs [14]. However, our focus here is on the most general case of generational algorithms such as the standard NSGA-II. Those techniques should probably need to be

[1]The lexicographical comparison between two solutions compares the value of the objectives of both solutions starting from the first one. If the values are the same, then the second objectives are considered. This is carried out iteratively until the values are different or their objectives are exactly the same.

adapted to work properly in this context, so we have not considered them in this article.

As we can observe in Table I, all algorithms have a computational complexity, in the worst case of $O(MN^2)$. Kung *et al.* [5] developed an algorithm that reaches an $O(MN \log N)$ computational complexity but is applied only to two objectives. Thus, the difference between the remaining algorithms lies in the average case and not in the worst case. In this sense, different algorithms apply different strategies to reduce the computational time.

In order to determine the dominance between two solutions, it is necessary to compare the values of their objectives, and a common quality indicator is the number of comparisons performed by an algorithm. For example, FNDS [1] compares all solutions among them and stores the result of each comparison to obtain the rank of each solution. HNDS [13] and ENS [9], [10] versions sort the population by the first objective value and use different data structures to differentiate a solution from the rest, and finally assign it a rank. BOS [3] sorts the population by each objective and assigns, to each solution s, M sets, where it stores those solutions that are not worse than s in each objective. DDA-NS [12] sorts the population by each objective, and stores in their dominance degree matrix the number of objectives in which one solution dominates the rest.

The MNDS strategy is quite straightforward. MNDS associates each solution with a total ordered set called *dominance set* ($s.ds$ in algorithms) containing the solutions that dominate it. More formally, given a solution $s_i \in P$, its *dominance set*, $s_i.ds$ contains all solutions that dominate s_i, that is, each $s_j \in P$ such that $s_j \preceq s_i$. Table II shows a set of solutions along their dominance set and rank. For example, solution s_{10} is dominated by solutions s_2, s_3, s_4, and s_6; therefore, the s_{10} dominance set is represented as $s_{10}.ds = \{s_2, s_3, s_4, s_6\}$. Once the dominance set of all solutions is computed, their rank is obtained as follows. Those solutions with an empty dominance set belong to the first rank. The rank of a solution with one or more elements in its dominance set will be calculated adding one to the largest rank of the solutions that compose the dominance set. Following our example, solutions s_2, s_3, s_4, and s_5 are not dominated and correspond to rank 1. Solution s_6 is dominated by s_3, that is, $s_6.ds = \{3\}$, and as a result belongs to the second rank. Finally, the rank of s_{10} is $\mathrm{Max}(R(s_2), R(s_3), R(s_4), R(s_6)) + 1 = 3$. Fig. 1 shows a 3-D representation of the example population in Table II. The three planes correspond to the three ranks. The higher the plane, the higher the rank, and the lowest plane corresponds to the first rank.

To obtain the dominance set of each solution, MNDS sorts the population by each objective. The order of the solutions corresponds to their dominance relationship for that objective, that is, the first solution is not dominated, the second is dominated by the first, and so on. To do so, MNDS creates the objective dominance set (ods). Once the population has been sorted by an objective, the ods is constructed traversing the ordered solutions and adding a solution to ods in each iteration. Therefore, we could define a function ods(*solution*, *objective*) that returns the content of the ods considering the position of a solution sorted for an objective. The dominance set of a solution s will be $s.ds = \mathrm{ods}(s, 1) \cap \mathrm{ods}(s, 2) \cap \cdots \cap \mathrm{ods}(s, M)$.

TABLE II
EXAMPLE OF A POPULATION WITH THEIR DOMINANCE SETS AND RANK

Population	Solution DS	Rank
s_1 = {34,30,41}	$s_1.ds = \{s_5\}$	2
s_2 = {33,34,30}	$s_2.ds = \emptyset$	1
s_3 = {32,32,31}	$s_3.ds = \emptyset$	1
s_4 = {31,34,34}	$s_4.ds = \emptyset$	1
s_5 = {34,30,40}	$s_5.ds = \emptyset$	1
s_6 = {36,33,32}	$s_6.ds = \{s_3\}$	2
s_7 = {35,31,43}	$s_7.ds = \{s_1, s_5\}$	3
s_8 = {37,36,39}	$s_8.ds = \{s_2,s_3,s_4,s_6,s_9\}$	3
s_9 = {35,34,38}	$s_9.ds = \{s_2,s_3,s_4\}$	2
s_{10} = {38,38,37}	$s_{10}.ds = \{s_2,s_3,s_4,s_6\}$	3
s_{11} = {39,37,31}	$s_{11}.ds = \{s_2,s_3\}$	2
s_{12} = {37,36,39}	$s_{12}.ds = \{s_2,s_3,s_4,s_6,s_9\}$	3

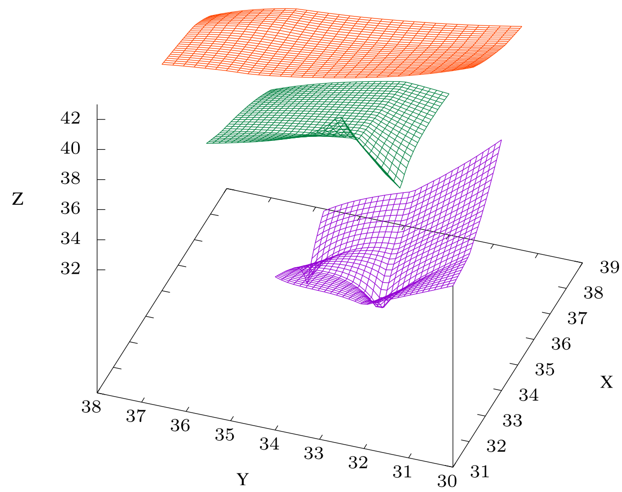


Fig. 1. 3-D representation of the example population.

TABLE III
EXAMPLE POPULATION SORTED BY OBJECTIVE 1

Population	ods	Solution DS
s_4 = {**31**,34,34}	$\emptyset$	$\emptyset$
s_3 = {**32**,32,31}	$\{s_4\}$	$\{s_4\}$
s_2 = {**33**,34,30}	$\{s_3,s_4\}$	$\{s_3,s_4\}$
s_5* = {**34**,30,40}	$\{s_2,...,s_4\}$	$\{s_2,...,s_4\}$
s_1* = {**34**,30,41}	$\{s_2,...,s_5\}$	$\{s_2,...,s_5\}$
s_7* = {**35**,31,43}	$\{s_1,...,s_5\}$	$\{s_1,...,s_5\}$
s_9* = {**35**,34,38}	$\{s_1,...,s_5, s_7\}$	$\{s_1,...,s_5, s_7\}$
s_6 = {**36**,33,32}	$\{s_1,...,s_5, s_7, s_9\}$	$\{s_1,...,s_5, s_7, s_9\}$
s_8* = {**37**,36,39}	$\{s_1,...,s_7, s_9\}$	$\{s_1,...,s_7, s_9\}$
s_{10} = {**38**,38,37}	$\{s_1,...,s_9\}$	$\{s_1,...,s_9\}$
s_{11} = {**39**,37,31}	$\{s_1,...,s_{10}\}$	$\{s_1,...,s_{10}\}$

*In case of ties, and only for the first objective, the lexicographic ordering is applied, as explained in Section III.

Tables III–V show the results of sorting the population on objectives 1–3, respectively. Each table also shows the objective dominance set (ods) and the dominance set ($s.ds$) for each solution. It is worth noting that solution s_{12} does not appear in Tables III–V as this solution is a duplicate of s_8 and MNDS removes duplicate solutions when sorting objective 1. In this way, the size of the population is reduced, avoiding unnecessary operations. In the end, those duplicate solutions are added back to the population with their rank (already calculated for the similar solution kept in the population). It can be observed that the dominance set in Table V corresponds to the actual one shown in Table II.

TABLE IV
EXAMPLE POPULATION SORTED BY OBJECTIVE 2

Population	ods	Solution DS
s_5* = {34,**30**,40}	$\emptyset$	$\emptyset$
s_1* = {34,**30**,41}	$\{s_5\}$	$\{s_5\}$
s_7 = {35,**31**,43}	$\{s_1, s_5\}$	$\{s_1, s_5\}$
s_3 = {32,**32**,31}	$\{s_1, s_5, s_7\}$	$\emptyset$
s_6 = {36,**33**,32}	$\{s_1, s_3, s_5, s_7\}$	$\{s_1, s_3, s_5, s_7\}$
s_4* = {31,**34**,34}	$\{s_1, s_3, s_5,..,s_7\}$	$\emptyset$
s_2* = {33,**34**,30}	$\{s_1, s_3,..,s_7\}$	$\{s_3, s_4\}$
s_9* = {35,**34**,38}	$\{s_1,..,s_7\}$	$\{s_1,..,s_5, s_7\}$
s_8 = {37,**36**,39}	$\{s_1,..,s_7, s_9\}$	$\{s_1,..,s_7, s_9\}$
s_{11} = {39,**37**,31}	$\{s_1,..,s_9\}$	$\{s_1,..,s_9\}$
s_{10} = {38,**38**,37}	$\{s_1,..,s_9, s_{11}\}$	$\{s_1,..,s_9\}$

*To equal values, merge sort keeps the order obtained from the previous objective. This is explained in detail in Section III.

TABLE V
EXAMPLE POPULATION SORTED BY OBJECTIVE 3

Solution	ods	Solution DS
s_2 = {33,34,**30**}	$\emptyset$	$\emptyset$
s_3 = {32,32,**31**}	$\{s_2\}$	$\emptyset$
s_{11} = {39,37,**31**}	$\{s_2, s_3\}$	$\{s_2, s_3\}$
s_6 = {36,33,**32**}	$\{s_2, s_3, s_{11}\}$	$\{s_3\}$
s_4 = {31,34,**34**}	$\{s_2, s_3, s_6\ s_{11}\}$	$\emptyset$
s_{10} = {38,38,**37**}	$\{s_2,..,s_4, s_6, s_{11}\}$	$\{s_2,..,s_4, s_6\}$
s_9 = {35,34,**38**}	$\{s_2,..,s_4, s_6, s_{10}, s_{11}\}$	$\{s_2,..,s_4\}$
s_8 = {37,36,**39**}	$\{s_2,..,s_4, s_6, s_9,..,s_{11}\}$	$\{s_2,..,s_4, s_6, s_9\}$
s_5 = {34,30,**40**}	$\{s_2,..,s_4, s_6, s_8,..,s_{11}\}$	$\emptyset$
s_1 = {34,30,**41**}	$\{s_2,..,s_6, s_8,..,s_{11}\}$	$\{s_5\}$
s_7 = {35,31,**43**}	$\{s_1,..,s_6, s_8,..,s_{11}\}$	$\{s_1, s_5\}$

This strategy has two important advantages: 1) it minimizes the number of comparisons among objective values of the solutions and 2) it performs an early detection of nondominance among the solutions of the population. When the dominance sets of all the solutions are empty, there is no dominance and the algorithm ends. This property is particularly advantageous when MNDS is used within MOEAs, such as NSGA, where the dominance among solutions decreases as the number of generations of the algorithm increases.

III. MERGE NONDOMINATED SORTING

As already stated in the previous section, the overall idea behind our proposal is to obtain the dominance set of each solution in the population, and then calculate their rank based on their corresponding dominance set. To obtain the dominance set for each solution s_i, it is necessary to sequentially sort the population by each of the objectives. The output obtained during the sorting of the mth objective is the input for objective $m + 1$th. A key point of our approach is the treatment of ties. In the case of the first objective, a lexicographical[1] comparison is used to break ties. If there are ties for all objective values, the second solution is considered as duplicate. For the rest of the objectives, ties are broken using the output of the previous iteration, that is, the order obtained with the previous objective. This is automatically done by the merge sort[2] algorithm.

We now formalize this approach and provide a step-by-step example to illustrate our algorithm. In population P, where each solution contains a vector of M objective values, the dominance set of solutions $s_i \in P$ can be obtained by sorting P iteratively by each objective as follows.

1) For the first objective ($m = 1$), the individuals are sorted taking into account the objective function value of the first objective. When there are ties, a lexicographical order is used to rank the individuals. Once the population is sorted by the first objective, each solution keeps its ordinal position in a variable s_i.index. The index is used to identify each solution and to create the dominance set in each algorithm. It is worth noting that we do not need to create ods for the first objective.[3] Finally, duplicate solutions are removed from the population.
2) For the second objective ($m = 2$), individuals are sorted by the objective function value of the second objective. In case of a tie, both solutions maintain the order obtained during the sorting of the first objective. Next, the dominance set of all solutions is initialized. The dominance set of each solution $s.ds$ consists of the indices contained in the set ods$(s, 2)$ whose values are lower than the solution index s.index, that is, $s.ds = \{u|u.\text{index} < s.\text{index and } u \in \text{ods}(s, 2)\}$. Finally, dominance between solutions is checked and when there is no dominance, MNDS stops as all solutions belong to the first rank.
3) For the remaining objectives ($1 < m \leq M$), we sort the population (previously sorted by objective $m-1$) by each objective m. In case of a tie, both solutions maintain the order from the previous ($m - 1$ objective) sorting. The dominance set of the ith solution s_i, in this order, is obtained by the ods(s, m) intersected with the previous dominance set of s_i, that is, $s_i.ds = \text{ods}(s_i, m) \cap s_i.ds$. As before, dominance is checked to decide whether to stop MNDS (all solutions belong to the first rank).

After sorting by the last objective, the dominance set of each solution $s_i.ds$ contains all the indices of the solutions that dominate s_i. The rank of a solution $s_i \in P$ will be the next rank to the highest rank of all the solutions $s_j \in s_i.ds$. In case that $s_i.ds$ is empty, s_i is assigned rank 1. Tables VI–VIII show the result of applying the previous steps to the example population of Table II. It is worth noting that when sorting by the first objective, solutions s_1 and s_5 have the same objective function value. Due to merge sort being a stable sorting algorithm, when comparing s_1 and s_5 lexicographically, we obtain that $s_5 \preceq s_1$ and the relationship between s_1 and s_5 will be maintained in case of ties when sorting by the next objectives. The treatment of duplicated is illustrated with individual s_{12}. In our example, solution s_{12} is a duplicate of solution s_8 and as a result, it is removed from the population while carrying out the sorting but added back again to the population after obtaining the ranking (due to most MOEAs need to keep their population size fixed).

[2]Merge sort is a stable sorting algorithm, that is, when it rearranges the population and there is a tie between two solutions, the relative position of both solutions in the population is maintained.

[3]For the first objective, the index of a solution $s \in P$ corresponds with its ordinal and therefore $s.ds = \{u|u.\text{index} < s.\text{index and } u \in P\}$.

TABLE VI
EXAMPLE POPULATION. SORTFIRSTOBJECTIVE()

	Population	Index
Algorithm steps:	s_4 = {**31**,34,34}	0
- Sort population by first	s_3 = {**32**,32,31}	1
objective	s_2 = {**33**,34,30}	2
- Remove duplicated solutions	s_5 = {**34**,30,40}	3
- Assign a solution index	s_1 = {**34**,30,41}	4
	s_7 = {**35**,31,43}	5
	s_9 = {**35**,34,38}	6
	s_6 = {**36**,33,32}	7
	s_8 = {**37**,36,39}	8
	s_{12} = {**37**,36,39}	duplicated
	s_{10} = {**38**,38,37}	9
	s_{11} = {**39**,37,31}	10

TABLE VII
EXAMPLE POPULATION. SORTSECONDOBJECTIVE()

	Population	Idx	ods	Sol. DS
Algorithm steps:	s_5 = {34,**30**,40}	3	∅	∅
- Sort population by	s_1 = {34,**30**,41}	4	{3}	{3}
second objective	s_7 = {35,**31**,43}	5	{3,4}	{3,4}
- For each solution:	s_3 = {32,**32**,31}	1	{3,..,5}	∅
- Compute	s_6 = {36,**33**,32}	7	{1,3,..,5}	{1,3,..,5}
dominance set	s_4 = {31,**34**,34}	0	{1,3,..,5,7}	∅
- Check global	s_2 = {33,**34**,30}	2	{0,1,3,..,5,7}	{0,1}
dominance	s_9 = {35,**34**,38}	6	{0,..,5,7}	{0,...,5}
	s_8 = {37,**36**,39}	8	{0,..7}	{0,...,7}
	s_{11} = {39,**37**,31}	10	{0,...8}	{0,...,8}
	s_{10} = {38,**38**,37}	9	{0,..,8,10}	{0,...,8}

TABLE VIII
EXAMPLE POPULATION. SORTRESTOFOBJECTIVES()

	Population	Idx	ods	Sol. DS
- For obj = 3 to 3:	s_2 = {33,34,**30**}	2	∅	∅
- Sort population	s_3 = {32,32,**31**}	1	{1,2}	∅
- For each sol.:	s_{11} = {39,37,**31**}	10	{1,2,10}	{1,2}
- Compute	s_6 = {36,33,**32**}	7	{1,2,10}	{1}
dominance set	s_4 = {31,34,**34**}	0	{1,2,7,10}	∅
- Check global	s_{10} = {38,38,**37**}	9	{0,..,2,7,10}	{0,..,2,7}
dominance	s_9 = {35,34,**38**}	6	{0,..2,7,9,10}	{0,..,2}
	s_8 = {37,36,**39**}	8	{0,..2,6,7,9,10}	{0,..,2,6,7}
	s_5 = {34,30,**40**}	3	{0,..2,6,..,10}	∅
	s_1 = {34,30,**41**}	4	{0,..3,6,..,10}	{3}
	s_7 = {35,31,**43**}	5	{0,..4,6,..,10}	{3,4}

TABLE IX
SUMMARY EXAMPLE

Order obj. 1→		Order obj. 2→		Order obj. 3→		Rank	
	Index	Index	Sol. DS	Index	Sol. DS	Index	Rank
s_4	0	3	∅	2	∅	2	1
s_3	1	4	{3}	1	∅	1	1
s_2	2	5	{3,4}	10	{1,2}	10	2
s_5	3	1	∅	7	{1}	7	2
s_1	4	7	{1,3,..,5}	0	∅	0	1
s_7	5	0	∅	9	{0,..,2,7}	9	3
s_9	6	2	{0,1}	6	{0,..,2}	6	2
s_6	7	6	{0,...,5}	8	{0,..2,6,7}	8	3
s_8	8	8	{0,...,7}	3	∅	3	1
s_{10}	9	10	{0,...,8}	4	{3}	4	2
s_{11}	10	9	{0,...,8}	5	{3,4}	5	3

A summary of the steps performed by MNDS with the sample population is shown in Table IX. The first two columns show the solutions sorted by the first objective and their associated index. The next four columns show the value of each solution dominance set after sorting by objectives 2 and 3. The last two columns show the index and rank of each solution, respectively. To further describe the example and following the steps previously described, the dominance set of solution s_7, for example, is obtained as follows.

1) *Objective 1:* The population is sorted (from lowest to highest) using this first objective. After sorting the population, an index is assigned to each solution. In our example, s_7 appears ordered in the fifth position, s_7.index $= 5$. This means that all solutions with an index value less than 5 dominate s_7 in the first objective. These solutions are s_4.index $= 0$, s_3.index $= 1$, s_2.index $= 2$, s_5.index $= 3$, and s_1.index $= 4$.
2) *Objective 2:* After sorting the population by objective 2, solutions with indices 3 and 4 dominate s_7, as a result, the dominance set of s_7 is initialized as $s_7.ds = \{3, 4\}$.
3) *Objective 3:* After sorting the population by objective 3, solution s_7 is dominated by the remaining solutions. Therefore, $s_7.ds = s_7.ds \cap \text{ods}(s_7, 3) = \{3, 4\} \cap \{0, \ldots, 4, 6, \ldots, 10\} = \{3, 4\}$.

Finally, the ranks of the solutions are obtained based on the dominance sets and duplicates are inserted again with their corresponding rank. In our example, solution s_7 is dominated by solutions with indices 3 and 4, which are s_5 and s_1, respectively. The rank of $s_7 = \text{Max}\{R(s_5), R(s_1)\} + 1 = \text{Max}\{1, 2\} + 1 = 3$.

A. Formalization of the MNDS Algorithm

As it can be observed in Algorithm 1, MNDS receives the population to sort as the only parameter. The process followed by MNDS can be divided into the following four phases.

1) Sort the population by the *first objective* and assign the solution ordinal to the index variable (s.index $= ord(s)$). Ties are broken using lexicographical comparison and the duplicated solutions are moved to a list of *duplicates* solutions (Algorithm 1, line 2). This list consists of tuples (*duplicate solution, original solution*). Note that although this requires more memory than just keeping the original solution with a list of duplicates, our solution is faster because it avoids searching through such a list. The rank of duplicate solutions is assigned at the end of Algorithm 1 (line 6).
2) Sort the population by the *second objective* and initialize the dominance set of each solution. If there is an iteration where all the dominance sets are empty, there is no dominance and MNDS ends since all solutions belong to the first rank.
3) Iteratively sort the population by the rest of the objectives $2 < m \leq M$. In case that in any iteration all the dominance sets are empty, that is, there is no dominance, MNDS ends since all solutions belong to the first rank.
4) Calculate the rank of each solution.

Lines 2–4 in Algorithm 1 correspond to the first three phases, respectively. These phases are in turn further described in Algorithms 2–4, respectively. The calculation of the ranking of each solution (phase 4) corresponds to lines 5 and 6.

Algorithms 2–4 sort the population P by the objective O using Algorithm 6, *MergeSort*(P, O). As previously stated, this algorithm is based on the merge sort algorithm. When sorting by the first objective ($O = 1$), in case of ties,

Algorithm 1 MNDS(P)

```
Input: population P
Output: ranking for each solution R
 1: R ← ∅
 2: duplicates ← SortFirstObjective(P)
 3: if SortSecondObjective(P) then
 4:     if SortRestOfObjectives(P) then
 5:         R ← GetRanking(P)
 6:         Update the rank of each duplicates solution with
            the rank of its original solution
 7:     end if
 8: end if
 9: return R
```

Algorithm 2 SortFirstObjective(P)

```
Input: population P
Output: population P, duplicate solutions duplicates
 1: ods ← ∅        ▷ Dominance set for this objective. ods is
    implemented with a bitset
 2: duplicates ← ∅
 3: MergeSort(P, 1)
 4: u ← P[1]                          ▷ auxiliary solution u
 5: ordinal ← 1
 6: u.index ← ordinal
 7: for s : P do                    ▷ s_i ∈ P, ∀i ∈ {2, ..., |P|}
 8:     ordinal ← ordinal + 1
 9:     if s ≠ u then
10:         s.index ← ordinal
11:     else
12:         duplicates ← duplicates ∪ s
13:         P ← P − s
14:     end if
15:     u ← s
16: end for
17: return P, duplicates
```

the lexicographical comparison is applied (see Algorithm 6, line 2).

The method *SortFirstObjective*(P) shown in Algorithm 2 implements the sorting by the first objective (phase 1). Line 3 sorts the population P by its first objective using the lexicographic rule in case of ties. Next, the loop (from lines 7 to 16) calculates the *index* of each solution (lines 8 and 10) and moves the duplicate solutions (see lines 12 and 13) to the *duplicates* list.

The method *SortSecondObjective*(P) shown in Algorithm 3 implements the sorting by the second objective (phase 2). The loop (from lines 6 to 8) initializes the solution dominance set ($s.ds$) with solutions in ods with an index lower than s.index. Note that, at each iteration i, the dominance set ods contains the solutions that dominate s_i solution for this objective.

The method *SortRestOfObjectives*(P) shown in Algorithm 4 implements the third phase. The first loop iterates through all objectives except the first two. The calculation of the dominance sets is carried out by the internal loop (lines 7–13), which also evaluates if there is dominance among the solutions

Algorithm 3 SortSecondObjective(P)

```
Input: population P
Output: population P, hasDominance Boolean with whether
    there is dominance
 1: ods ← ∅                ▷ Dominance set for this objective.
 2: hasDominance ← false
 3: MergeSort(P, 2)
 4: for s : P do
 5:     s.ds ← ∅
 6:     if subSet(ods, 1, s.index − 1) ≠ ∅ then
 7:         top ← Min(s.index − 1, ods.max)
 8:         s.ds ← ods.subSet(ods.min, top)
 9:         hasDominance ← true
10:     end if
11:     ods ← ods ∪ s.index
12: end for
13: return P, hasDominance
```

Algorithm 4 SortRestOfObjectives(P)

```
Input: population P
Output: population P, hasDominance Boolean with whether
    there is dominance
 1: hasDominance ← true
 2: Obj ← 3
 3: while Obj ≤ M ∧ hasDominance do
 4:     if MergeSort(P, Obj) then
 5:         hasDominance ← false
 6:         ods ← ∅            ▷ Dominance set for this objective
 7:         for s : P do
 8:             s.ds ← s.ds ∩ ods
 9:             ods ← ods ∪ s.index
10:             if s.ds ≠ ∅ then
11:                 hasDominance ← true
12:             end if
13:         end for
14:     end if
15:     Obj ← Obj + 1
16: end while
17: return P, hasDominance
```

(line 10). When there is no further dominance, the method ends.

The last phase, the calculation of the population ranking, is implemented by the method *GetRanking*(P). In this method, the variable *maxRank* always contains the highest rank value of all evaluated solutions. Note that the rank of a solution s is always in the range $[1, maxRank + 1]$. The internal loop (lines 6–16) traverses the dominance set $s.ds$, obtaining the rank ($iR[i]$, line 7) of each solution in the current dominance set. If that value is greater than current *rank*, the *rank* value is increased to $iR[i] + 1$ (line 8). Likewise, if the value of the *rank* variable is greater than *maxRank* (line 10), the *rank* value is assigned to *maxRank* and the search ends. Note that all dominance sets contain indices to solutions and the rank $iR[]$ is calculated taking into account these indices. The ranking of all solutions is stored in R in line 13.

Algorithm 5 GetRanking(P)

```
Input: Population P
Output: Population Ranking R
R ← ∅
iR ← ∅                  ▷ Ranking considering solution indices
maxRank ← 0
for s : P do
    rank ← 0            ▷ Ranking of population solutions
    for i : s.ds do     ▷ for each solution index i in s.ds
        if iR[i] ≥ rank then
            rank = iR[i] + 1
        end if
        if rank > maxRank then
            maxRank ← rank
            iR[i] ← rank
            R[s] ← rank
            break
        end if
    end for
end for
return R
```

Algorithm 6 MergeSort(P, O)

```
Input: population P, objective O
Output: population P sorted by objective O,
    isSorted Boolean with whether input P is already ordered
if O = 1 then
    isSorted ← Sort P by objective O. In case of ties,
    apply lexicographical order for objectives O + 1 to M
else
    isSorted ← Sort P by objective O.
end if
return P, isSorted
```

B. Implementation Considerations

We make use of *bitsets* to deal with sets operations. The motivation behind using bitsets to represent sets is their capability of maintaining the set sorted to facilitate the insertion of elements with a complexity of $O(1)$ while in other implementations, such as lists, their cost is $O(\log N)$. It is worth noting that we use *sorted* sets to speed up the intersection operation between sets. Furthermore, our implementation of bitsets considers the range of the values in the set $[\min, \max]$, the intersection between two sets a and b is only applied within the range $[\mathrm{Max}(a.\min, b.\min), \mathrm{Min}(a.\max, b.\max)]$. Therefore, the intersection in Algorithm 4, line 8 will not be calculated if the solutions in ods do not dominate the solution s.

C. Computational and Spatial Complexity

MNDS is based on the merge sort algorithm (see Algorithm 6) which has the best and worst computational complexity of $O(N \log N)$. The computational complexity of MNDS (Algorithm 1), in the worst case scenario, is the sum of the complexities of the methods shown in Algorithms 2, 4, and 5, which are calculated as follows.

1) *Algorithm 2:* The worst case belongs to the sorting which has a complexity of $O(N \log N)$.
2) *Algorithm 3:* Sorting by objective 2 has a complexity $O(N \log N)$. The loop initializes the dominance set ($s.ds$) of all solutions, so its complexity is $O(N)$. Each $s.ds$ is initialized with the indices of those solutions in ods whose index is less than s.index. The worst case occurs when the first solution dominates the second, the second to the third, and so on. In that case, the complexity is $O(N)$, so the complexity of Algorithm 3, in the worst case, is $O(N^2)$. The best case occurs when there is no dominance between solutions, or each solution is dominated only by another. In that case, the initialization of every $s.ds$ is $O(1)$, and the best case of Algorithm 3 is $O(N \log N)$.
3) *Algorithm 4:* The inner loop calculates the dominance of all solutions in P ($O(N)$), which computes the intersection $s.ds \cap$ ods ($O(N)$) so this loop has a worst complexity of ($O(N^2)$). The external loop sorts ($O(N \log N)$) the population P for each objective except the first two, that is, objectives 3 to M. Therefore, the complexity for this algorithm is $O((M-2)(N \log N + N^2))$. The best case occurs when there is no dominance between solutions, or each solution is dominated only by another. In that case, the calculation of the intersection $s.ds \cap$ ods has a complexity ($O(1)$) and the best case of Algorithm 4 is $O(N \log N)$. Please, note that Algorithm 4 also performs the early detection of nondominance, in order to minimize the calculations as much as possible.
4) *Algorithm 5:* It consists of two nested loops, so its worst computational complexity is $O(N^2)$. The best case of computational complexity occurs, once again, when each solution is dominated by another solution. In that case, the complexity is $O(N)$.

Therefore, the worst complexity of MNDS is the sum of $O(N \log N)$, $O(N^2)$, $O((M-2)(N \log N + N^2))$, and $O(N^2)$ which is equal to $O(MN^2)$. The best case happens when there is no dominance among the solutions. In this case, at the end of Algorithm 3 all the dominance sets of the solutions are empty. In this case, MNDS ends and its complexity is $O(N \log N)$.

It is important to emphasize the difference between our proposal and the state-of-the-art algorithms. In our case, when the number of fronts decreases the algorithm tends to approximate the behavior of the best case. The spatial complexity is determined by the size of the *dominance sets* of each solution ($|s_i.ds| = |P|$) which corresponds to $O(N^2)$.

IV. Experimental Work

A. Implementation Details

To validate the performance of MNDS,[4] we compare the computational time of MNDS with six state-of-the-art algorithms: 1) BOS [3]; 2) HNDS [13]; 3) ENS-SS [9]; 4) ENS-BS [9]; 5) ENS-NDT [10]; and 6) DDA-NS [12]. To do so, we use the BOS implementation provided by the authors.[5]

[4] MNDS is integrated into the jMetal framework: `https://github.com/jMetal/jMetal.`

[5] `https://github.com/Proteek/Best-Order-Sort`

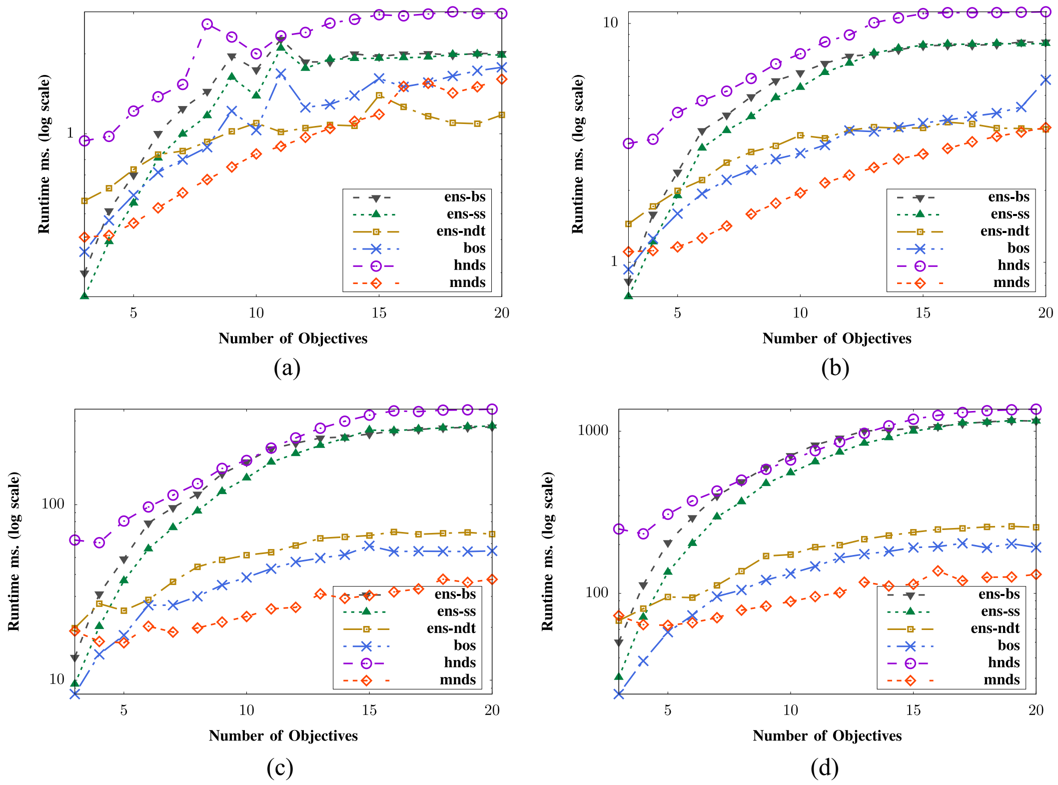


Fig. 2. Experiment 1—Results with a fixed number of solutions, and increasing the number of objectives, using the BOS dataset. (a) 500 solutions. (b) 1000 solutions. (c) 5000 solutions. (d) 10 000 solutions.

For the ENS-SS and ENS-BS algorithms, the implementations provided by Buzdalov[6] were used with minimal modifications. ENS-NDT was implemented in Java from the C# source code provided by the authors. In addition, we implemented the HNDS and DDA-NS algorithms from scratch. In addition to computational time, in some experiments, we also count the number of comparisons of objective function values, as it has been done in similar studies. We have to point out that most of the computing time in MNDS is not invested in comparing the objective function values of the solutions, so a good performance of our algorithm is expected.

The implementation of all the algorithms was done in Java without using multithreading nor specific CPU/GPU features as SIMD[7] or similar.

B. Experimental Settings

In order to compare the algorithms previously discussed, four types of experiments were carried out.

1) Varying the number of objectives for a fixed number of solutions, using the BOS dataset.[5]
2) Varying the population size for a fixed number of objectives, using again the BOS dataset.
3) Varying the number of objectives for a fixed population size, using datasets generated by NSGA-II. In this case, we have additionally obtained the number of comparisons made by each algorithm.
4) Executing the algorithms within the NSGA-II algorithm. To do so, we have replaced the original FNDS algorithm in NSGA-II for each of the evaluated algorithms.

The original BOS dataset contains 10 000 solutions with up to ten objectives; we extended it to 20 objectives, generating the new values randomly.

In experiment 1), the algorithms were executed varying the number of objectives between 3 and 20, with population sizes of 500, 1000, and 5000. In experiment 2), the size of the population ranged between 500 and 10 000 with an increase of 1000 for 5, 10, 15, and 20 objectives. For experiment 3), the NSGA-II [1] implementation from jMetal [15] was used to generate 16 datasets obtained after 500 generations for the DTLZ1 [16], DTLZ2 [16], WFG1 [17], and WFG2 [17] problems with 5, 10, 15, and 20 objectives. The population size used was 1000 solutions. All the algorithms were executed 5000 times under the same conditions using the execution time as a performance measure. The final execution time was calculated averaging those 5000 executions.[8] It is worth noting that the NSGA-II was used with the same problems and applying the same configuration as the one defined in this article describing BOS. In this way, when comparing MNDS against BOS, we are also comparing MNDS, indirectly, with the other algorithms that were also compared with the BOS algorithm, that is, FNDS, deductive sort, corner sort, and

[6]https://github.com/mbuzdalov/nondominated-sorting

[7]Single instruction, multiple data.

[8]This number of executions mitigates possible differences in runtime values due to the behavior of the just-in-time compiler and the garbage collector of Java.

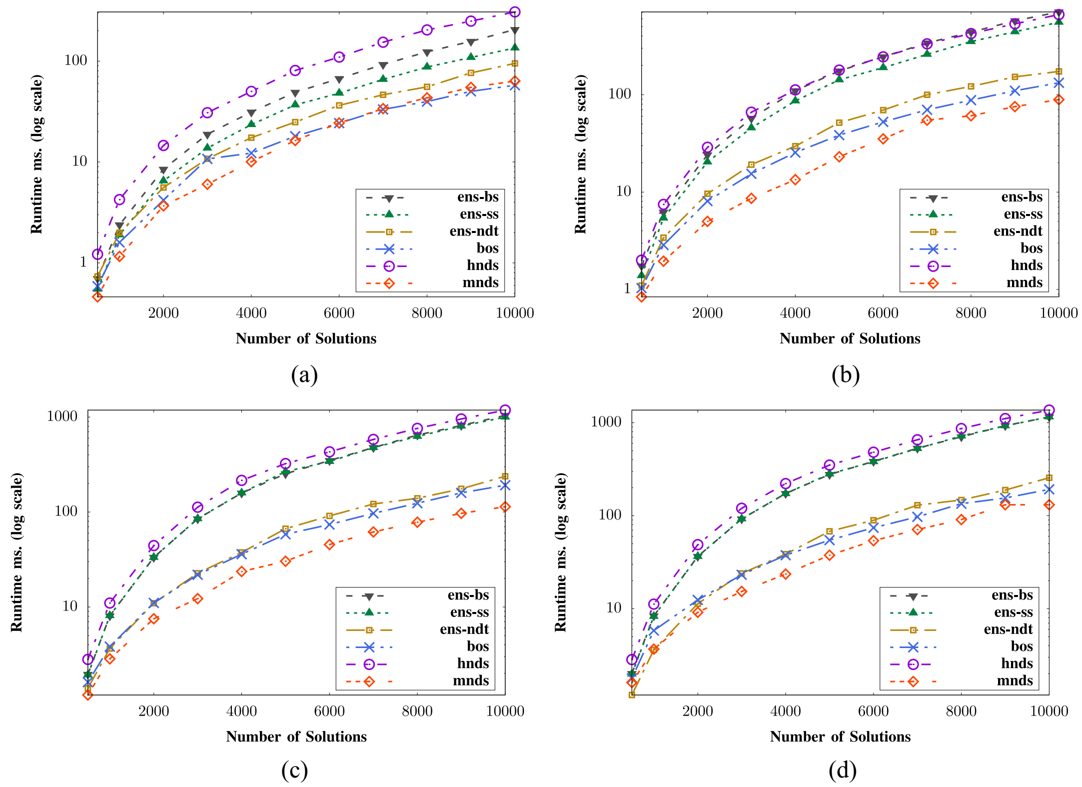


Fig. 3. Experiment 2—Results with a fixed number of objectives, and increasing the number of solutions, using the BOS dataset. (a) 5 objectives. (b) 10 objectives. (c) 15 objectives. (d) 20 objectives.

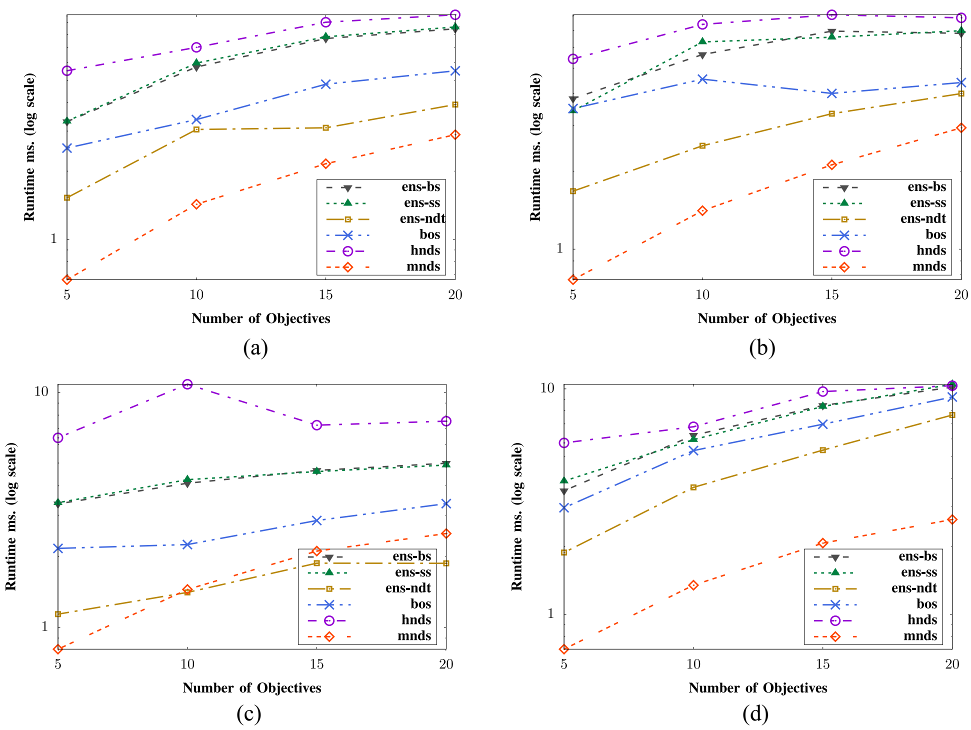


Fig. 4. Experiment 3—Results with the dataset generated by NSGA-II with a population of 1000 solutions after 500 generations (a) DTLZ1. (b) DTLZ2. (c) WFG1. (d) WFG2.

divide-and-conquer sort. Finally, in experiment 4), NSGA-II was configured to run for 2000 generations with a population size of 1000 solutions. The crossover operator used was the simulated binary crossover, with a distribution index value of 20 and a crossing probability of 90%. As a mutation operator, we used the polynomial mutation. All the executions used the

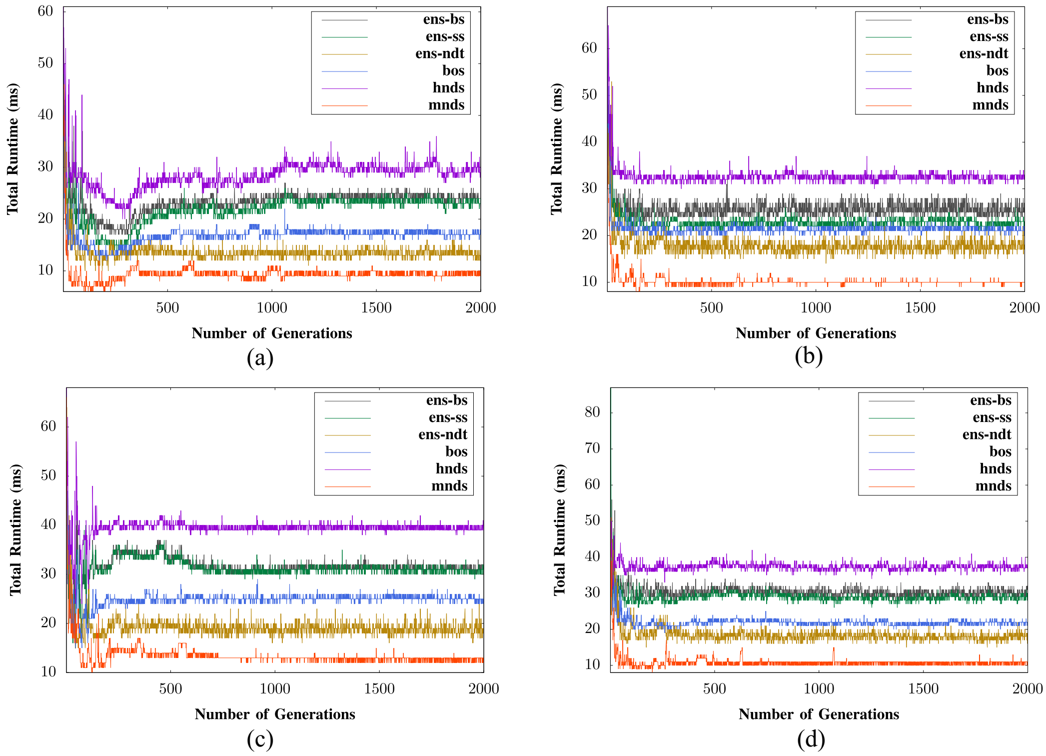


Fig. 5. Experiment 4—Results executing the algorithms within NSGA-II for 2000 generations with a population of 1000 solutions. (a) DTLZ1. (b) DTLZ2. (c) WFG1. (d) WFG2.

same random seed, and therefore, with the same initial population. The DTLZ and WFG algorithms were configured for five objectives.

The computer and software versions used have the following features.

1) Debian GNU/Linux 9.0. 64-bits architecture.
2) 4 x Intel Core i5 CPU M-460. 2.53 GHz.
3) 8 GB of RAM memory.
4) *Java Version:* 1.8.0-121, 64 bits.

C. Results

The execution times obtained in the experiments, by all the algorithms, but DDA-NS, are shown in Figs. 2–5. We have excluded DDA-NS from the figures due to the large differences in performance with the rest of the algorithms. The MNDS algorithm was designed to work efficiently with large population sizes as well as with a large number of objectives. As a result, the algorithm maintains a very high performance even if we increase the number of objectives or the size of the population. With few objectives ($\leq$5) and/or a small number of solutions, the behavior of the compared techniques is similar, except for HNDS and DDA-NS which perform worse than the rest. However, it can be observed in the figures that: 1) as the number of objectives increases or the size of the population increases, the rest of the algorithms suffer a performance degradation and 2) only BOS and ENS-NDT algorithms present a performance close to MNDS when using the BOS dataset.

TABLE X
EXPERIMENT 3—NUMBER OF COMPARISONS MADE BY THE ALGORITHMS WITH THE DATASET GENERATED BY NSGA-II WITH A POPULATION OF 1000 SOLUTIONS AFTER 500 GENERATIONS

	Obj	BOS	ENS-SS	ENS-BS	ENS-NDT	HNDS	DDA-NS	MNDS
DTLZ1	5	5.37e+08	2.40e+06	2.40e+06	4.20e+08	1.97e+06	5.79e+07	**6.84e+04**
	10	5.08e+08	3.83e+06	3.83e+06	1.68e+09	2.68e+06	1.20e+08	**1.34e+05**
	15	7.10e+08	2.40e+06	2.40e+06	6.53e+08	1.97e+06	5.83e+07	**6.88e+04**
	20	6.13e+08	6.78e+06	6.78e+06	2.34e+09	4.15e+06	2.41e+08	**2.66e+05**
DTLZ2	5	6.65e+08	2.50e+06	2.50e+06	6.65e+08	2.01e+06	5.96e+07	**6.75e+04**
	10	5.31e+08	3.70e+06	3.70e+06	9.71e+08	2.61e+06	1.21e+08	**1.34e+05**
	15	4.88e+08	4.26e+06	4.26e+06	1.35e+09	2.89e+06	1.81e+08	**2.00e+05**
	20	5.31e+08	5.39e+06	5.39e+06	2.10e+09	3.46e+06	2.41e+08	**2.67e+05**
WFG1	5	5.05e+08	2.26e+06	2.26e+06	7.60e+08	1.94e+06	5.46e+07	**7.04e+04**
	10	3.56e+08	2.86e+06	2.86e+06	6.87e+08	2.19e+06	1.18e+08	**1.35e+05**
	15	3.50e+08	3.39e+06	3.39e+06	6.51e+08	2.46e+06	1.77e+08	**2.02e+05**
	20	3.75e+08	2.88e+06	2.88e+06	5.30e+08	2.20e+06	2.38e+08	**2.57e+05**
WFG2	5	4.97e+08	2.12e+06	2.12e+06	3.21e+08	1.82e+06	6.00e+07	**6.82e+04**
	10	7.45e+08	3.80e+06	3.80e+06	1.50e+09	2.66e+06	1.20e+08	**1.35e+05**
	15	9.20e+08	5.48e+06	5.48e+06	2.77e+09	3.50e+06	1.80e+08	**2.01e+05**
	20	1.18e+09	9.07e+06	9.07e+06	5.28e+09	5.30e+06	2.40e+08	**2.54e+05**

The computing times obtained with the datasets generated by NSGA-II in experiment 3), indicate that the differences with the other algorithms are noticeable. The results of experiment 3) can be observed in Fig. 4. As stated in Section III-C, MNDS tends to approximate the behavior of the best case when the number of fronts decreases. Such behavior can be observed in Fig. 5 where MNDS outperforms the rest of the algorithms when they are running within NSGA-II. We have to note that in this experiment we are reporting the execution time of each NSGA-II iteration as performance measure; we made ten independent runs but the differences in the running times were negligible (the standard deviations were very small) to be observed in the graphs, so we decided to plot a single value.

The number of comparisons made by each algorithm in experiment 3) is shown in Table X, where we can observe that MNDS requires a number of comparisons that is at least one order of magnitude lower than the best of the other algorithms compared. This result confirms the expectations we claimed in Section IV-A.

V. Conclusion

In this article, we have presented a new and efficient algorithm for computing the nondominated sorting called MNDS based on the merge sort algorithm. The experimental work showed that MNDS strongly outperforms the current state-of-the-art algorithms in terms of running time and the number of comparisons carried out. Therefore, MOEAs based on Pareto ranking can strongly benefit from significant time reductions, particularly in the case of using large populations and solving many-objective problems.

As future work, we plan to enhance our approach in several ways. Particularly, we think that the algorithm used to calculate the ranking of each solution from the domination sets could be improved by the use of different search methods and data structures. For example, we could use the Timsort algorithm instead of merge sort, and the sequential search used in finding the ranking of each solution could be replaced by a binary search or a k-d tree. We will also explore issues, such as that the performance of the MOEAs using MNDS will increase as the number of nondominated solutions also increases. This circumstance occurs in all MOEA algorithms, that is, as the generations advance, the number of nondominated solutions also increases. Finally, we will also consider to adapt our algorithm to be used in steady-state EAs.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[2] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Comput. Eng. Netw. Lab., Swiss Federal Inst. Technol., Zürich, Switzerland, Rep. 103, 2001.

[3] P. Roy, M. M. Islam, and K. Deb, *Best Order Sort: A New Algorithm to Non-Dominated Sorting for Evolutionary Multi-Objective Optimization*, Assoc. Comput. Mach. Inc., New York, NY, USA, Jul. 2016, pp. 1113–1120.

[4] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1162/evco.1994.2.3.221

[5] H. T. Kung, F. Luccio, and F. P. Preparata, "On finding the maxima of a set of vectors," *J. ACM*, vol. 22, no. 4, pp. 469–476, Oct. 1975. [Online]. Available: http://doi.acm.org/10.1145/321906.321910

[6] H. Fang, Q. Wang, Y.-C. Tu, and M. F. Horstemeyer, "An efficient non-dominated sorting method for evolutionary algorithms," *Evol. Comput.*, vol. 16, no. 3, pp. 355–384, Nov. 2008. [Online]. Available: http://dx.doi.org/10.1162/evco.2008.16.3.355

[7] K. McClymont and E. Keedwell, "Deductive sort and climbing sort: New methods for non-dominated sorting," *Evol. Comput.*, vol. 20, no. 1, pp. 1–26, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00041

[8] H. Wang and X. Yao, "Corner sort for Pareto-based many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92–102, Jan. 2014.

[9] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to non-dominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, Apr. 2015.

[10] P. Gustavsson and A. Syberfeldt, "A new algorithm using the non-dominated tree to improve non-dominated sorting," *Evol. Comput.*, vol. 26, no. 1, pp. 89–116, Mar. 2018. [Online]. Available: https://doi.org/10.1162/evco_a_00204

[11] M. Drozdík, Y. Akimoto, H. Aguirre, and K. Tanaka, "Computational cost reduction of nondominated sorting using the M-front," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 659–678, Oct. 2015.

[12] Y. Zhou, Z. Chen, and J. Zhang, "Ranking vectors by means of the dominance degree matrix," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 34–51, Feb. 2017.

[13] C. Bao, L. Xu, E. D. Goodman, and L. Cao, "A novel non-dominated sorting algorithm for evolutionary multi-objective optimization," *J. Comput. Sci.*, vol. 23, pp. 31–43, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877750317310530

[14] K. Li, K. Deb, Q. Zhang, and Q. Zhang, "Efficient nondomination level update method for steady-state evolutionary multiobjective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2838–2849, Sep. 2017.

[15] J. J. Durillo and A. J. Nebro, "jMetal: A java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.advengsoft.2011.05.014

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 1, May 2002, pp. 825–830.

[17] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

Bibliography

[1] K. Miettinen, *Nonlinear multiobjective optimization.* Boston, USA: Kluwer, 1999.

[2] M. Harman and B. F. Jones, "Search-based software engineering," *Information and Software Technology*, vol. 43, no. 14, pp. 833–839, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/B6V0B-44D4196-4/2/fa86eaf9f0f0d2d05bcf64c38b160eed

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE-EC*, vol. 6, pp. 182–197, Apr. 2002.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, Tech. Rep. 103, May 2001.

[5] K. Deb, K. Miettinen, and S. Chaudhuri, "Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches," *Evolutionary Computation, IEEE Transactions on*, vol. 14, pp. 821 – 841, 01 2011.

[6] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research*, vol. 13, pp. 533–549, 1986.

[7] ——, "Tabu search–Part I," *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[9] J. Holland, *Adaptation in Natural and Artificial Systems.* University of Michigan Press, 1975.

[10] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. of the IEEE Int. Conf. on Neural Networks.* Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948.

[11] M. Dorigo, "Optimization, learning and natural algorithms," Ph.D. dissertation, Politecnico di Milano, Italy, 1992, (In Italian).

[12] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *J. Global Optimization*, vol. 39, no. 3, pp. 459–471, 2007. [Online]. Available: http://dx.doi.org/10.1007/s10898-007-9149-x

[13] X. S. Yang, *Nature-Inspired Methaeuristic Algorithms.* Luniver Press, 2008.

[14] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci*, vol. 179, no. 13, pp. 2232–2248, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2009.03.004

[15] C. A. Coello Coello, G. Toscano Pulido, and M. Salazar Lechuga, "Handling Multiple Objectives With Particle Swarm Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[16] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Stochastic Algorithms: Foundations and Applications, 5th International Symposium, SAGA 2009, Sapporo, Japan, October 26-28, 2009. Proceedings*, ser. Lecture Notes in Computer Science, O. Watanabe and T. Zeugmann, Eds., vol. 5792. Springer, 2009, pp. 169–178. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-04944-6

[17] H. R. Hassanzadeh and M. Rouhani, "A multi-objective gravitational search algorithm," in *Second International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2010, Liverpool, UK, 28-30 July, 2010*, D. Al-Dabass, A. A. Pantelous, A. Orsoni, and A. Abraham, Eds. IEEE, 2010, pp. 7–12. [Online]. Available: http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5613973

[18] A. Mishra, M. Das, and T. Panda, "Multi-objective artificial bee colony (moabc) algorithm to improve content-based image retrieval performance," *Journal of Theoretical and Applied Information Technology*, vol. 59, pp. 745–758, 01 2014.

[19] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "MOCell: A cellular genetic algorithm for multiobjective optimization," *Int. J. Intell. Syst*, vol. 24, no. 7, pp. 726–746, 2009. [Online]. Available: http://dx.doi.org/10.1002/int.20358

[20] J. D. Knowles and D. W. Corne, "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

[21] M. Li and X. Yao, "Quality evaluation of solution sets in multiobjective optimisation: A survey," *ACM Comput. Surv.*, vol. 52, no. 2, Mar. 2019. [Online]. Available: https://doi.org/10.1145/3300148

[22] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms–A comparative case study," in *Parallel Problem Solving from Nature – PPSN V (5th*

PPSN'98), ser. Lecture Notes in Computer Science (LNCS), A. E. Eiben, T. Back, M. Schoenauer, and H.-P. Schwefel, Eds. Amsterdam, The Netherlands: Springer-Verlag (New York), Sep. 1998, vol. 1498, pp. 292–304.

[23] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1999.

[24] E. Zitzler, "Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications," Ph.D. dissertation, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Nov. 1999.

[25] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a pareto front," in *Late Breaking Papers at the Genetic Programming 1998 Conference*, J. R. Koza, Ed. University of Wisconsin, Madison, Wisconsin, USA: Stanford University Bookstore, 22-25 Jul. 1998, pp. 221–228. [Online]. Available: http://www.lania.mx/~ccoello/EMOO/vanvel2.ps.gz

[26] C. A. Coello Coello and M. Reyes Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *MICAI 2004: Advances in Artificial Intelligence*, R. Monroy, G. Arroyo-Figueroa, L. E. Sucar, and H. Sossa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 688–697.

[27] Yiu-Wing Leung and Yuping Wang, "U-measure: a quality measure for multiobjective programming," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 33, no. 3, pp. 337–343, 2003.

[28] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[29] M. P. Fanti, G. Iacobellis, W. Ukovich, V. Boschian, G. Georgoulas, and C. Stylios, "A simulation based decision support system for logistics management," *Journal of Computational Science*, vol. 10, pp. 86–96, 2015.

[30] D. Schmaranzer, R. Braune, and K. F. Doerner, "A discrete event simulation model of the viennese subway system for decision support and strategic planning," in *Winter Simulation Conference (WSC)*, 2016, pp. 2406–2417.

[31] M. A. Zaffar, R. L. Kumar, and K. Zhao, "Diffusion dynamics of open source software: An agent-based computational economics (ACE) approach," *Decision Support Systems*, vol. 51, no. 3, pp. 597–608, 2011.

[32] T. Chesney, S. Gold, and A. Trautrims, "Agent based modelling as a decision support system for shadow accounting," *Decision Support Systems*, vol. 95, pp. 110–116, 2017.

[33] X. Fu, M. Dong, S. Liu, and G. Han, "Trust based decisions in supply chains with an agent," *Decision Support Systems*, vol. 82, pp. 35–46, 2016.

[34] T. Ng, M. Khirudeen, T. Halim, and S. Chia, "System dynamics simulation and optimization with fuzzy logic," in *IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2009)*, December 2009, pp. 2114–2118.

[35] T. Kremmel, J. Kubalík, and S. Biffl, "Software project portfolio optimization with advanced multiobjective evolutionary algorithms," *Applied Soft Computing*, vol. 11, pp. 1416–1426, January 2011. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2010.04.013

[36] M. C. Fu, *Handbook of Simulation Optimization.* Springer Publishing Company, Incorporated, 2014.

[37] E. Orta, M. Ruiz, N. Hurtado, and D. Gawn, "Decision-making in {IT} service management: a simulation based approach," *Decision Support Systems*, vol. 66, no. 0, pp. 36–51, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167923614001766

[38] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[39] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Tech. Rep., 2001.

[40] J. J. Durillo and A. J. Nebro, "jMetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1016/j.advengsoft.2011.05.014

[41] I. González, J. Gómez, A. Tayebi, and M.F. Cátedra, "Optimization of a dual-band helical antenna for ttc applications at s band," *IEEE Antennas and Propagation Magazine*, vol. 54, no. 4, pp. 63–77, 2012.

[42] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to nondominated sorting for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 201–213, April 2015.

[43] P. Gustavsson and A. Syberfeldt, "A new algorithm using the non-dominated tree to improve non-dominated sorting," *Evol. Comput.*, vol. 26, no. 1, pp. 89–116, Mar. 2018. [Online]. Available: https://doi.org/10.1162/evco_a_00204

[44] P. Roy, M. Islam, and K. Deb, *Best order sort: A new algorithm to non-dominated sorting for evolutionary multi-objective optimization.* Association for Computing Machinery, Inc, 7 2016, pp. 1113–1120.

[45] H. Fang, Q. Wang, Y.-C. Tu, and M. F. Horstemeyer, "An efficient non-dominated sorting method for evolutionary algorithms," *Evol. Comput.*, vol. 16, no. 3, pp. 355–384, 9 2008. [Online]. Available: http://dx.doi.org/10.1162/evco.2008.16.3.355

[46] K. McClymont and E. Keedwell, "Deductive sort and climbing sort: New methods for non-dominated sorting," *Evol. Comput.*, vol. 20, no. 1, pp. 1–26, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00041

[47] H. Wang and X. Yao, "Corner sort for pareto-based many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92–102, Jan 2014.

[48] M. Drozdík, Y. Akimoto, H. Aguirre, and K. Tanaka, "Computational cost reduction of nondominated sorting using the m-front," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 659–678, Oct 2015.

[49] Y. Zhou, Z. Chen, and J. Zhang, "Ranking vectors by means of the dominance degree matrix," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 1, pp. 34–51, Feb 2017.

[50] C. Bao, L. Xu, E. D. Goodman, and L. Cao, "A novel non-dominated sorting algorithm for evolutionary multi-objective optimization," *Journal of Computational Science*, vol. 23, pp. 31 – 43, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877750317310530

[51] N. Auger, V. Jugé, C. Nicaud, and C. Pivoteau, "On the worst-case complexity of timsort," 2018.