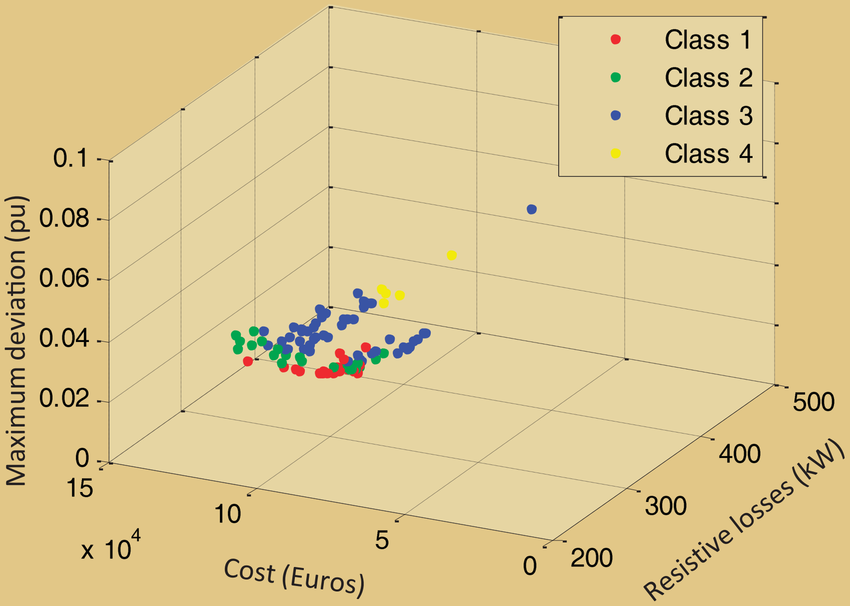


Eunice Sandra Gomes de Oliveira

INCORPORATION OF PREFERENCES, ADAPTIVE OPERATORS AND HYBRIDIZATION IN MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Dissertação de Doutoramento na área científica de Engenharia Eletrotécnica, orientada pelo Senhor Professor Doutor Carlos Alberto Henggeler de Carvalho Antunes e pelo Senhor Professor Doutor Álvaro Filipe Peixoto Cardoso de Oliveira Gomes, e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

dezembro de 2012

· U C ·

Universidade de Coimbra

UNIVERSIDADE DE COIMBRA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

INCORPORATION OF PREFERENCES, ADAPTIVE OPERATORS AND HYBRIDIZATION IN MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Eunice Sandra Gomes de Oliveira

Dissertação submetida à Universidade de Coimbra para obtenção do grau de Doutor em Engenharia Eletrotécnica, especialidade em Optimização e Teoria de Sistemas

COIMBRA
dezembro de 2012

Tese realizada sob orientação do

Professor Doutor Carlos Alberto Henggeler de Carvalho Antunes

Professor Catedrático do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

e do

Professor Doutor Álvaro Filipe Peixoto Cardoso de Oliveira Gomes

Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra

Aos meus pais

Ao Jorge

“Devemos ter uma margem de ilusão para seguir com a vida para a frente.”

AGRADECIMENTOS

As primeiras palavras de agradecimento são para os meus orientadores, Professor Carlos Henggeler Antunes e Professor Álvaro Gomes, pelo incondicional apoio prestado ao longo destes anos, tanto em termos científicos como a nível pessoal. Não posso deixar de salientar a competência científica e o rigor sempre colocado no trabalho realizado. Agradeço ainda toda a motivação que me transmitiram, o acompanhamento constante e as agradáveis reuniões de trabalho.

Agradeço ainda aos meus colegas de laboratório, em especial ao Paulo Lima e ao Jorge Martinho que trabalharam diretamente comigo na elaboração deste trabalho, nomeadamente na implementação dos modelos dos estudos de caso. O meu obrigada por toda a ajuda, pelas dúvidas esclarecidas, mas principalmente por todo o companheirismo e pelo bom ambiente de trabalho.

Ao IT (Instituto de Telecomunicações de Coimbra) agradeço o facto de ter possibilitado o uso do cluster para a realização de simulações computacionais. Agradeço ainda ao Professor Luis Cruz toda a disponibilidade e ajuda prestada.

À Xana o meu agradecimento pela força e apoio nesta reta final. Por vezes não é preciso dizer muito, mas falar no momento certo com as palavras certas. Xana, não me esqueço da “red line”!

À Joana, à Rita, ao Paulo e ao Bruno pela nossa amizade.

Aos meus pais por todo o esforço que fizeram para que eu pudesse estudar, por todo o apoio e toda a ajuda que me têm dado, por todos os valores que me transmitiram ao longo da vida. Aos meus pais, que tantas vezes os privei da minha companhia e da minha ajuda, que tantas vezes tentei diminuir a distância e minimizar as saudades com os nossos telefonemas.

As últimas palavras vão para o Jorge, para o “meu” Jorge, não por serem as menos importantes, mas por serem aquelas que a emoção não quer deixar escrever. Ao Jorge, o meu companheiro, o meu ombro amigo, o meu porto de abrigo, só posso agradecer por TUDO! Pela partilha dos sorrisos e das alegrias nos momentos bons, por toda a ajuda e força nos momentos mais difíceis, por todo o carinho, pela preocupação constante, por todos os ensinamentos e principalmente por estar ao “meu lado” ao longo de já 17 anos! Isso mesmo, por estar ao meu lado neste e noutros projectos.

Acknowledgments

This thesis has been partially supported by Energy and Mobility for Sustainable Regions Project CENTRO-07-0224-FEDER-002004, and FCT under projects MIT/SET/0018/2009 and PEst-C/EEI/UI0308/2011.

RESUMO

A resolução de um problema de optimização multiobjectivo envolve, em geral, não apenas uma fase de pesquisa, capaz de fornecer um conjunto representativo da frente óptima de Pareto, mas também uma fase de decisão, consistindo na escolha da solução (ou conjunto de soluções) aceitável como recomendação final tendo em vista a sua aplicação prática. Neste sentido, a incorporação de preferências durante o processo evolutivo permite focar a pesquisa evitando a exploração de soluções irrelevantes (minimizando assim o tempo de computação) e facilita a integração de conhecimento do decisor no processo de pesquisa (minimizando o esforço cognitivo). Estes aspectos são particularmente importantes quando o número de funções objectivo é grande e/ou a sua natureza é conflituante, uma vez que a dimensão do espaço de pesquisa assim como o número de soluções não-dominadas admissíveis tende a ser elevado.

A proposta de uma abordagem evolutiva, designada por EvABOR (Evolutionary Algorithm Based on an Outranking Relation), apresentada neste trabalho incorpora as preferências de um decisor de modo a guiar a pesquisa para regiões do espaço mais de acordo com as preferências explicitadas. Estas são captadas e tornadas operacionais recorrendo aos parâmetros e princípios do método ELECTRE TRI. A relação de prevalência (*outranking*), na qual o ELECTRE TRI se baseia, é usada para substituir/complementar a relação de não dominância nos habituais operadores do algoritmo evolutivo (cruzamento, mutação e selecção).

Dado que a qualidade das soluções iniciais pode influenciar o desempenho de um algoritmo evolutivo, e existindo conhecimento sobre o problema em causa, nomeadamente ao lidar com problemas reais, propõe-se uma metodologia de construção de soluções iniciais baseada no GRASP (Greedy Randomized Adaptive Search Procedure) permitindo também a incorporação de preferências. Adicionalmente, a necessidade de explorar as regiões do espaço de pesquisa de forma mais eficiente, levou à implementação de um procedimento de pesquisa local, baseado no Simulated Annealing, onde as preferências explicitadas pelo decisor são tidas em conta, sendo também incorporadas numa versão multiobjectivo do Simulated Annealing. Este trabalho teve como resultado um novo algoritmo onde se explora a hibridização do GRASP com o Simulated Annealing, incorporando as preferências tanto na fase de construção como na fase de pesquisa local.

Os algoritmos propostos foram aplicados na resolução de dois problemas recorrendo a dados reais: um problema de compensação de energia reactiva em redes de distribuição de energia eléctrica, no caso do EvABOR, e um problema de controlo remoto de cargas, no caso do algoritmo híbrido.

ABSTRACT

The resolution of a multi-objective optimization problem involves, in general, not only a search phase adequate to provide a representative set of the Pareto-optimal front, but also a decision phase consisting in the identification of a solution (or a set of solutions) acceptable as a final recommendation having in mind practical implementation. The incorporation of preferences during the evolutionary process allows focusing the search according to the preference information elicited from the decision maker, avoiding the exploration of irrelevant solutions (thus minimizing the computational time) and facilitating the integration of knowledge in the search process (minimizing the cognitive effort). These aspects are particularly important in combinatorial problems, when the number of objective functions is large and/or their nature is conflicting, since the size of the search space as well as the number of non-dominated solutions tends to be very high.

The evolutionary approach, called EvABOR (Evolutionary Algorithm Based on an outranking Relation), presented in this work incorporates the decision maker's preferences to guide the search for regions of the space more in accordance with the elicited preferences. These are captured and made operational using the principles and parameters of the ELECTRE TRI method. The outranking relation in the ELECTRE TRI method is used to replace/complement the non-dominance relation in the usual evolutionary algorithm operators (crossover, mutation and selection).

Since the quality of the initial solutions may influence the performance of an evolutionary algorithm a methodology based on GRASP (Greedy Randomized Adaptive Search Procedure) is proposed for the construction of initial solutions. This procedure is particularly relevant when knowledge about the problem at hand exists, which happens, in general, in real-world problems. Additionally, the need to exploit regions of the search space more efficiently led to the implementation of a local search procedure based on Simulated Annealing, in which the preferences elicited from a decision maker are taken into account. This motivated the development of a new approach for multi-objective optimization problems in which GRASP and Simulated Annealing are hybridized, incorporating preferences in the construction phase or/and the local search phase.

The proposed algorithms are applied to provide decision support in the resolution of two real-world problems: a reactive power compensation problem in electrical distribution

networks, using the EvABOR algorithm, and a direct load control problem, using the hybrid algorithm.

INDEX

ACRONYMOUS LIST

AC	Air Conditioner
ACO	Ant Colony Optimization
DM	Decision Maker
DSM	Demand-side Management
DLC	Direct Load Control
EA	Evolutionay Algorithm
ELECTRE	*ELimination Et Choix Traduisant la REalité*
EvABOR	Evolutionary Algorithm Based on an Outranking Relation
GA	Genetic Algorithm
GRASP	Greedy Randomized Adaptive Search Procedure
HESA	Hybrid Evolutionary Simulated Annealing
HMH	Hybrid Metaheuristic
ITS	Iterated Local Search
NCL	No Controlled Load
NSGA	Nondominated Sorting Genetic Algorithm
MOGA	Multi-Objective Genetic Algorithm
MOGLS	Multi-Objective Genetic Local Search
MOO	Multi-Objective Optimization
MOEA	Multi-Objective Evolutionary Algorithm
MOEA/D	Multi-Objective Evolutionary Algorithm based on Decomposition
MOOP	Multi-Objective Optimization Problem
MORGA	Multi-Objective Random Greedy search Algorithm
MOSA	Multi-Objective Simulated Annealing
PAES	Pareto Archived Evolution Strategy
M-PAES	Memetic Pareto Archived Evolution Strategy
PSA	Pareto Simulated Annealing
PSO	Particle Swarm Optimization
PT	Power Transformer
RCL	Restrict Candidate List
SA	Simulated Annealing
SPEA	Strength Pareto Evolutionary Algorithm

SS	Sub-station
TS	Tabu Search
VEGA	Vector Evaluated Genetic Algorithm
VNS	Variable Neighborhood Search

FIGURE LIST

TABLE LIST

CHAPTER 1

INTRODUCTION[1]

The adequate modeling of real-world optimization problems generally requires the consideration of distinct axes of evaluation for assessing the merits of potential solutions. Namely in engineering problems, aspects of operational, economical, environmental, and quality of service nature are at stake. Therefore, mathematical models must explicitly address these multiple, incommensurate and often conflicting aspects of evaluation as objective functions to be optimized. Multi-objective optimization (MOO) models enable to grasp the trade-offs between the objective functions that are relevant for decision purposes in order to reach a satisfactory compromise solution that can be accepted as the final outcome of the process. The essential concept in MOO is the one of non-dominated (efficient, Pareto-optimal) solutions, that is feasible solutions for which no improvement in all objective functions is possible simultaneously, since in order to improve an objective function it is necessary to accept worsening at least another objective function value. Then in MOO the main goal is to obtain the non-dominated set of solutions, named non-dominated front in the objective function space. However, it must be noticed that, in many real-world problems, the non-dominated front is unknown, being the goal of the optimization process to achieve an approximated front, as close as possible of the true one.

The use of Evolutionary Algorithms (EAs) to deal with MOO models has gained an increasing relevance due to their ability to work with a population of individuals (solutions) that hopefully converges to the true non-dominated front [Deb (2001), Coello et al. (2002)]. EAs are particularly suited for tackling hard combinatorial and/or non-linear models, as they are less susceptible to the shape or continuity of the non-dominated front than classical (mathematical programming) optimization methods. EAs can incorporate techniques to

[1] This chapter is partially based on [Oliveira E, Antunes C H, Gomes A. A comparative study of different approaches using an outranking relation in a multi-objective evolutionary algorithm. Computers & Operations Research 2013; 40 (6): 1602–1615].

preserve the diversity of solutions for a comprehensive depiction of non-dominated front thus unveiling the trade-offs in different regions of the search space. These techniques possess advantages compared with the use of “scalarizing” functions, in which a surrogate scalar function aggregating the multiple objectives is optimized, as in traditional mathematical programming approaches.

Although the non-dominance is the essential concept in MOO, it is a poor one, in the sense that it lacks discriminative power for decision recommendation purposes. Non-dominated solutions are not comparable between them, so no solution arises as the “final” one [Branke (2008), Branke et al. (2010)]. The rationalization of the comparison between non-dominated solutions requires taking into account the expression of the decision maker’s preferences that somehow “enrich” the non-dominance relation [Rachmawati and Srinivasan (2006)]. These preferences represent a set of opinions, values, convictions and perspectives of reality, which configure a personal model of the reality under study, which the decision maker (DM) leans on to evaluate the different potential solutions [Coello (2000), Branke and Deb (2004), Corne and Knowles (2007)].

Recent studies have shown that EAs based only on the non-dominance relation are insufficient to deal with MOO models, namely whenever the number of objective functions is large [Corne and Knowles (2007), Farina and Amato (2002), Knowles and Corne (2007)]. In these situations, the non-dominance relation may become inefficient in the selection of individuals for the next generation and lead to a weak selective pressure [di Pierro et al. (2007), Garza-Fabre et al. (2009)]. As it is referred to in [Deb et al. (2010)], in these cases the progress of the population tends to slow down and, the time consumed in the search process to find at least, a good approximation to the non-dominated front may become prohibitive. In addition to the problems associated with the selection procedure and the time consumed in the search process, a major difficulty arises at the end of the optimization process when it is necessary to choose a solution (or a small set of solutions for further screening) having in mind its practical implementation. In fact, in a real-world multi-objective optimization problem (MOOP), the number of solutions in the non-dominated front is generally very large due to the conflicting nature of the objective functions, possibly its number, and the frequent combinatorial nature of the problem [Branke et al. (2010)].

The preference information supplied by the DM is of paramount importance to guide the search to the regions where solutions more in accordance with his/her preferences are located, thus narrowing the scope of the search to the regions of interest and reducing the computational effort [Branke and Deb (2004), Branke et al. (2001), Rachmawati and

Srinivasan (2009)]. The convergence to these regions is improved by incorporating mechanisms for preference expression into the evolutionary process. Therefore, techniques aimed at meaningfully capturing and incorporating the DM's preferences into the evolutionary process should play a key role in real-world decision processes based on complex (namely combinatorial) MOO models.

Besides EAs, other metaheuristics, such as Tabu Search (TS), Simulated Annealing (SA), Greedy Randomized Adaptive Search Procedure (GRASP), Iterated Local Search (ITS), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Variable Neighborhood Search (VNS), among others, have been applied to real-world optimization problems displaying a good performance [Osman and Laporte (1996), Osman and Kelly (1996), Dréo et al. (2006), Talbi (2009)]. Metaheuristics may be characterized as solution search methods for complex problems, namely of combinatorial and/or strong nonlinear characteristics for which the resolution using (exact) mathematical programming algorithms is impossible or computationally unacceptable in a reasonable time. The term metaheuristic was introduced by Glover in 1977 [Glover (1977)]. Several definitions for this term can be found in the literature. For Voß et al. (1999) a metaheuristic "is an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions at each iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method". Metaheuristics balance exploration (diversification) and exploitation (intensification) procedures to effectively scan the search space. A wide overview of metaheuristics may be found in [Glover and Kochenberger (2003), Coello et al. (2002), Blum and Roli (2003), Talbi (2009)]. In this thesis, the main concepts of SA and GRASP metaheuristics, as well as their main approaches in the context of MOO will be presented in Chapter 4.

The different characteristics of each metaheuristic make some of them more suited for certain types of problems. Consequently, a new research trend has been developed along this perspective, focusing mainly on the problem rather than the algorithm [Blum et al. 2010]. With this perspective, the combination of some components from different metaheuristics is made to improve the performance of the overall approach thus leading to a new class of algorithms, called hybrid metaheuristics (HMH). The benefit of HMH is to include in the same algorithm the advantages of different metaheuristics working as a symbiosis with the same aim, which reveals to be a more efficient method to solve the problem at hand.

In the context of MOOP, having in mind the resolution of real-world problems, the issues referred to above and the recent developments in this area, three main algorithms are proposed in this work: an EA with the incorporation of preferences, named EvABOR (Evolutionary Algorithm Based on an Outranking Relation), an EA improved with a local search phase, named HESA (Hybrid Evolutionary Simulated Annealing), and a hybrid meta-heuristic combining characteristics of GRASP and SA with incorporation of preferences. Two real-world problems have been dealt with using each one of these algorithms: a reactive power compensation problem and a direct load control problem.

In this chapter the main motivations for the development of the proposed algorithms have been presented. In Chapter 2 the main concepts about multi-objective optimization are presented as well as the state-of-the-art in incorporation of preferences in EAs and also in hybrid metaheuristics with the focus on the hybridization of GRASP and SA.

Chapter 3 is devoted to the development of the EvABOR algorithms. Preferences are incorporated in those algorithms using the parameters and principles of the ELECTRE TRI method which is devoted to the sorting problem, i.e., assigning alternatives (solutions) to ordered categories of merit according to multiple evaluation aspects. Consequently the first section of this chapter is devoted to explain the main concepts of this method. Three versions of the EvABOR algorithm are presented and used to obtain a set of non-dominated solutions to the reactive power compensation problem in accordance with the preferences elicited from a DM. The main difference between the three versions of EvABOR consists in the priority given to the non-dominance relation and the outranking relation (used in the ELECTRE TRI method) in the algorithms. Experiments using the reactive power compensation problem allow us to conclude about the best way to combine these two relations to obtain a non-dominated set of solutions according to preferences elicited from a DM.

In Chapter 4 a general review about SA and GRASP metaheuristics is done, followed by the presentation of the proposed hybrid metaheuristics (the HESA and the GRASP+SA algorithms). The HESA algorithm is an extended approach of EvABOR with local search to improve the convergence of the initial algorithm to regions more in accordance with preferences. Considering the importance of the quality of initial solutions, a construction phase is developed in the spirit of the GRASP algorithm. The results obtained from the application of these algorithms to a direct load control problem are analyzed in the last section of this chapter. Finally, the most relevant conclusions about this work are drawn in the last chapter, as well as some directions in future research are pointed out.

CHAPTER 2

MULTI-OBJECTIVE OPTIMIZATION:

INCORPORATION OF PREFERENCES INTO EAS AND HMH[2]

2.1. INTRODUCTION

A multi-objective optimization problem (MOOP) is characterized by the maximization or minimization of several objective functions possibly subject to a set of constraints. In this work, without loss of generality, the general concepts are presented for the minimization problem (if a maximization problem occurs it is explicitly referred). Considering this assumption a MOOP may be defined, in a general form, as follows:

$$\begin{aligned} &\text{Minimize } f_m(X),\ m = 1,\ldots, M;\\ &\text{subject to } g_j(X)\geq 0,\ j = 1,\ldots, J;\\ &h_k(X)=0,\ k = 1,\ldots, K;\\ &x_i^{(L)} \leq x_i \leq x_i^{(U)},\ i = 1,\ldots, n, \end{aligned} \tag{2.1}$$

where f_m are the objective functions to minimize, g_j and h_k are constraints imposed by the particular characteristics of the MOOP; X is a vector of n decision variables x_1, x_2, …, x_n and, $x_i^{(L)}$ and $x_i^{(U)}$ define the limits of the decision variable space. A vector X belonging to the decision variable space and satisfying all constraints is a feasible solution to the MOOP.

In MOOP it does not exist, in general, a single feasible solution optimizing all objective functions. Due to their conflicting nature, there are solutions for which an improvement in one objective can only be obtained by decreasing the performance in other objective functions. Consequently, the concept of optimal solution, considered in single-objective

[2] This chapter is partially based on [Oliveira E, Antunes C H, Gomes A. A comparative study of different approaches using an outranking relation in a multi-objective evolutionary algorithm. Computers & Operations Research 2013; 40 (6): 1602–1615] and on [Oliveira E, Antunes C H, Gomes A. Incorporation of preferences in an evolutionary algorithm using an outranking relation - the EvABOR approach, International Journal of Natural Computing Research 2011; 2 (1), 63–85].

optimization problems, is replaced by the concept of non-dominated (efficient or Pareto-optimal) solutions. A solution X_1 dominates another solution X_2 when X_1 is better than X_2 in at least one objective, and not worse in the other objectives. A solution is non-dominated if there is no other feasible solution that dominates it. The concept of efficient or non-inferior solution generally refers to the decision variable space whereas the concept of non-dominance or Pareto optimality generally refers to its image on the objective function space.

The complexity and the dimension of MOOPs, particularly the combinatorial nature and the characteristics of the search space, require suitable methodological and computational tools. In recent years, the use of metaheuristics in the resolution of MOOP has increased significantly due to their ability to find a set of good quality solutions, although with no guarantee of Pareto-optimality, involving a reasonable computational effort without imposing too exigent requirements to the mathematical models. These techniques are particularly used in problems for which there are no suitable mathematical programming algorithms and/or the computational time is prohibitive.

Several multi-objective metaheuristic approaches adapt the procedures designed for single-objective problems by resorting to weighted-sum scalar functions aggregating by means of weighting coefficients the multiple objective functions explicitly considered in the mathematical model. On one hand, this may be very limited because although recognizing the problem as a multi-objective one, the method to tackle it is then just a single objective optimization process without considering the true nature of the multi-objective problem. On the other hand, even by defining a strategy for changing the weights in a well-distributed manner it is not guaranteed that this results in a well-spread and diverse non-dominated front. Additionally, in combinatorial problems with binary and integer decision variables the optimization of weighted-sum scalar functions is not able to reach unsupported solutions (that is, those for which no supporting hyper-plane exists). Last but not least, interpreting these weights as coefficients of importance associated with the objective functions is generally not correct due to their interdependence with the measurement scales [Das and Dennis (1997), Deb (2001)].

EAs are one of the most popular metaheuristics approaches to deal with MOOPs. EAs are based on Darwin's evolutionary theory, and they mimic the evolutionary principles of the nature in the context of search and optimization problems. In contrast to the classical methods, EAs deal with a population of individuals (solutions) at each iteration (generation) instead of a single solution, which allows finding multiple Pareto-optimal solutions in a single run. EAs are indifferent to the convexity and form of the Pareto-optimal front which is an additional advantage. Using a biological analogy, a solution can be viewed as a chromosome whose elements are the genes, and new solutions are generated using operators that replicate the recombination and the mutation of chromosomes. In general, the recombination (crossover) operator works on two individuals (parents) and generates two or more individuals (offspring), which are the combination of their parent's genes. The natural selection (the "survival of the fittest") is also replicated in the EAs: the performance of each individual is evaluated and the ones with best performances have a higher probability to pass their characteristics to the next generation. In general, the evolutionary process begins with a population generated randomly and this evolves in the direction of the Pareto-optimal front in successive iterations using the crossover, mutation and selection operators.

The first EA, called Vector Evaluated Genetic Algorithm (VEGA), is proposed by Schaffer in [Schaffer (1985)]. The main drawback of this approach is its lack of promoting diversity. If VEGA is applied for a large number of iterations the population tends to converge to individual optimal solutions. In [Goldberg (1989)] a multi-objective evolutionary algorithm (MOEA) using the concept of non-dominance is proposed and the use of a sharing mechanism to preserve the diversity of the non-dominated set. Several EAs devoted to MOOP have been developed; some of the most popular are Multi-Objective Genetic Algorithm (MOGA) [Fonseca and Fleming (1993)], Strength Pareto Evolutionary Algorithm (SPEA) [Zitzler and Thiele (1999)], Pareto Archived Evolution Strategy (PAES) [Knowles and Corne (2000)] and Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [Deb et al. (2002)]. Details about EAs can be found in the vast literature [Fonseca and Fleming (1995), Coello (1999), Veldhuizen and Lamont (2000), Deb (2001), Coello et al. (2002), Zhou et al. (2011)].

Despite the recognized success of EAs in the resolution of MOOPs, some issues have been considered, particularly in problems where the dimension of the search space is huge, which generally arises in real-world optimization problems. Consequently some mechanisms have been proposed to improve the efficiency and the efficacy of search procedures. Two of these approaches are the incorporation of preferences into EAs, to use additional information to guide the search process both reducing the computational effort and the cognitive effort in processing the results having in mind a final choice, and the hybridization of metaheuristics, as a means to use different techniques in different search phases or particular regions of the search space.

2.2. INCORPORATION OF PREFERENCES INTO EAS

As in multi-objective mathematical programming algorithms, the incorporation of preferences into an EA can be done using one of the three main approaches as *a priori*, *a posteriori* and progressively (interactive) [Steuer (1986)].

In the *a priori* approach the preferences are elicited from the DM before the EA starts. A value (or utility) function is usually considered to transform the MOOP into a scalar optimization problem, in which the single objective function embodies the preference expression parameters [Fonseca and Fleming (1993), Deb (1999)]. A disadvantage usually pointed out to this approach lies on the fact that it is necessary to elicit all the preference information from the DM without knowledge of the possible alternatives, particularly in complex MOO mathematical models.

The *a posteriori* approach is the most used in evolutionary MOO. In this approach the non-dominated front is evaluated exhaustively with the aim of obtaining the whole Pareto-optimal front or at least the best approximation to this front. In *a posteriori* approaches a significant computational effort is generally devoted to the search of solutions that may be uninteresting from a practical point of view and convey no value-added for decision support purposes. The well-known algorithms NSGA-II [Deb et al. (2002)], SPEA2 [Zitzler et al. (2002)] and PAES [Knowles and Corne (1999)] are examples of *a posteriori* methods, which are aimed at characterizing thoroughly the Pareto-optimal front.

In the progressive (interactive) approach the preferences elicited from the DM are used to guide the search during the evolutionary process [Thiele et al. (2009), Chaudhuri and Deb (2010), Branke et al. (2010), Deb et al. (2010)]. It is assumed that those preferences may change over time as more knowledge is gathered, not just about the solution space and the trade-offs at stake between the objective functions, but also about the shaping of a DM's preference structure. That is, the solutions provided by the EA contribute to a preference refinement process that in turn leads to focusing the search onto the regions in which solutions more in accordance with the preferences expressed by the DM are located. This enables a learning process of the trade-offs at stake between the competing objectives in different regions of the search space. Interactive approaches may require *a priori* specification of a few preference information parameters, while other parameters may be provided during the evolutionary process. In some interactive algorithms the preferences are elicited based on a set of solutions which are presented to the DM during the evolutionary process. This is the approach used in [Branke et al. (2010)] and

[Deb et al. (2010)] where, after a certain number of generations, a small set of alternatives is presented to the DM for choosing her/his preferred one or assessing intensities of preference between pairs of alternatives. Using this information the algorithm determines one [Deb et al. (2010)], or more [Branke et al. (2010)], value functions and the EA searches for non-dominated solutions that optimize these functions.

The previous classification is done according to the point in the search process at which the incorporation of preferences occurs. Preference information may be incorporated into an EA using distinct sets of technical parameters, which may also be representative of different DM's attitudes. Different processes can be referred to, such as goal attainment [Fonseca and Fleming (1993)] (in which the main idea is to be as close as possible to goals the DM would like to attain in each objective function), the specification of acceptable trade-off between objectives [Branke et al. (2001)] (thus using the notion of marginal rates of substitution) and the relative importance between objectives [Cvetkovic and Parmee (2002), Jin and Sendhoff (2002)] (assigning importance weights to the objectives). In other cases the concept of non-dominance is modified and/or replaced by other constructs [Fernández et al. (2010), Said et al. (2010)]. Some recent works use an outranking relation (between a pair of alternatives) for preference incorporation into an EA. Rekiek et al. (2000) combines the PROMETHEE II method [Brans and Mareschal (1986)] with an EA to rank each population during the evolutionary process based on preferences elicited *a priori*. Also, Coelho et al. (2003) use PROMETHEE II during the EA, in a method called PAMUC, to deal with preferences and constraints in an *a priori* approach. An *a posteriori* approach based on concepts of ELECTRE III and PROMETHEE methods is presented in [Fonteix et al. (2004)]. In [Parreiras et al. (2006), Parreiras and Vasconcelos (2007)] the PROMETHEE II method, or a modified PROMETHEE II, is also applied *a posteriori* for further analyzing the solutions in the Pareto-optimal front.

In the algorithms proposed in this thesis, an outranking relation is used to enrich the non-dominance relation. The preferences are captured and made operational by using the parameters of the ELECTRE TRI method. The choice of this method relies on the fact that it is devoted to the sorting problem, allowing a comparison of each solution with predefined standards rather than using comparisons between solutions as in methods devoted to the choice and ranking problems as, for example, ELECTRE I and ELECTRE II/III. Another important advantage of ELECTRE TRI is the possibility of using a veto threshold to account for non-compensatory aspects. Although the non-dominance relation is the essential one in MOOP, it is not sufficient to include further non-controversial elements of the DM's preferences in order to discriminate between non-dominated solutions. As happens with other methods, ELECTRE TRI also allows to consider the relative importance of each objective function using a set of weights. However, in ELECTRE TRI, these values are scale independent and are truly coefficients of importance assigned to the objective functions, and they are not used to build a value function. Also, ELECTRE TRI enables the preference relation to be established in a gradual manner using indifference and preference thresholds and it is possible to express the exigency of the sorting defining a cutting-level.

2.3. HYBRIDIZATION OF METAHEURISTICS

Despite the success of metaheuristics and the applicability of these algorithms to a diverse set of problems in several areas (aerodynamics, fluid dynamics, telecommunications, data mining, finance, scheduling and production problems, among others), it has been recognized that there is no algorithm considered as the best approach to solve all type of problems [Raidl (2006), Talbi (2009), Blum et al. (2011)]. This assertion is also stated in [Wolpert and Macready (1997)] supported by the "no free lunch" theorem meaning that if an algorithm performs well on one set of problems then it will perform poorly (worse than random search) on others [Montgomery (2002)]. Consequently, in the last years a new paradigm of algorithms combining components of two or more metaheuristics has arisen. These algorithms, named hybrid metaheuristics (HMH), attempt to combine advantageous characteristics of each metaheuristic to obtain a more efficient and effective algorithm for the resolution of the problem at hand. As referred above, in HMH the focus is on the characteristics of the problem, and consequently the choice of metaheuristics to hybridize is extremely dependent on the nature of the problem.

Several classifications of HMH have appeared in the literature, either due to the proposal of new HMHs or the different perspectives concerning how to combine the metaheuristics operational components [Talbi (2002), El-Abd and Kamel (2005), Raidl (2006), Talbi (2009), Blum et al. (2010)]. In the classification proposed by Raidl (2006) different categories are considered, taking into account the type of algorithms being hybridized, the level at which the different algorithms are combined (whether the identity of an algorithm is preserved or not), order of execution (if it is sequential, interleaved or parallel), and the control strategy (integrative or collaborative). This classification is based on the taxonomy proposed by Talbi (2002) also integrating the points-of-view of Cotta (1998) and Blum et al. (2005). Concerning the parallelism of HMH, this classification relies on the one proposed by El-Abd and Kamel (2005) and Cotta et al. (2005). The subcategories considered with respect to the hybridization of metaheuristics with exact optimization techniques are adapted from Puchinger and Raidl (2005). Figure 2.1 shows Raidl's classification.

Talbi (2009) proposes other classification for metaheuristics, dividing the metaheuristics in four main groups: metaheuristics hybridized with metaheuristics, exact methods, constraint programming approaches, and machine learning and data mining techniques. For each group, Talbi classifies the metaheuristics as flat (homogeneous/heterogeneous, global/partial, general/specialist) or hierarchical (low level versus high level, and relay versus teamwork).

More recently, Blum et al. (2010) classify the HMH in five categories. This classification is more restricted than the one proposed by Raidl and it is only based on the type of the combined algorithms. The authors consider the hybridization of metaheuristics with (meta-) heuristics, constraint programming, tree search methods, problem relaxations, and dynamic programming.

2.4. Hybridization of Algorithms with SA or GRASP Components

One of the most popular hybrid algorithms is the combination of a local search method with other metaheuristics to intensify the exploitation of high-quality solutions in a specific local area and improve the convergence of the algorithm. The genetic local search (also called memetic algorithms) is an example of this type of hybridization, in which a local search phase is incorporated into an EA [Moscato (1989), Moscato et al. (2004), Hart et al. (2005), Neri et al.(2011)]. In MOOPs, the aim is to obtain a set of non-dominated solutions as near as possible to the Pareto-optimal front preserving the diversity of the population. However, the use of local search can introduce further complications for achieving diversity in the population, and therefore a convenient balance between local search and the evolutionary process must be achieved [Knowles and Corne (2005), Krasnogor and Smith (2005), Nguyen et al. (2007)]. A particular study about the balance between genetic search and local search in memetic algorithms applied to multi-objective permutation flowshop scheduling is done in [Ishibuchi et al. (2003)]. Despite the large number of works using metaheuristics in MOOP, few multi-objective memetic algorithms are described in the literature until 2005: A multi-objective genetic local search (MOGLS) is proposed in [Ishibuchi and Murata (1996, 1998)]; in Knowles and Corne (2000) the memetic Pareto archived evolution strategy (M-PAES) is presented and [Jaszkiewicz (2002, 2002a, 2004)] proposes a random directions MOGLS and the Pareto memetic algorithm. A review about these and other works is done in

[Knowles and Corne (2005)]. Ishibuchi and Yamamoto (2004) present a fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining.

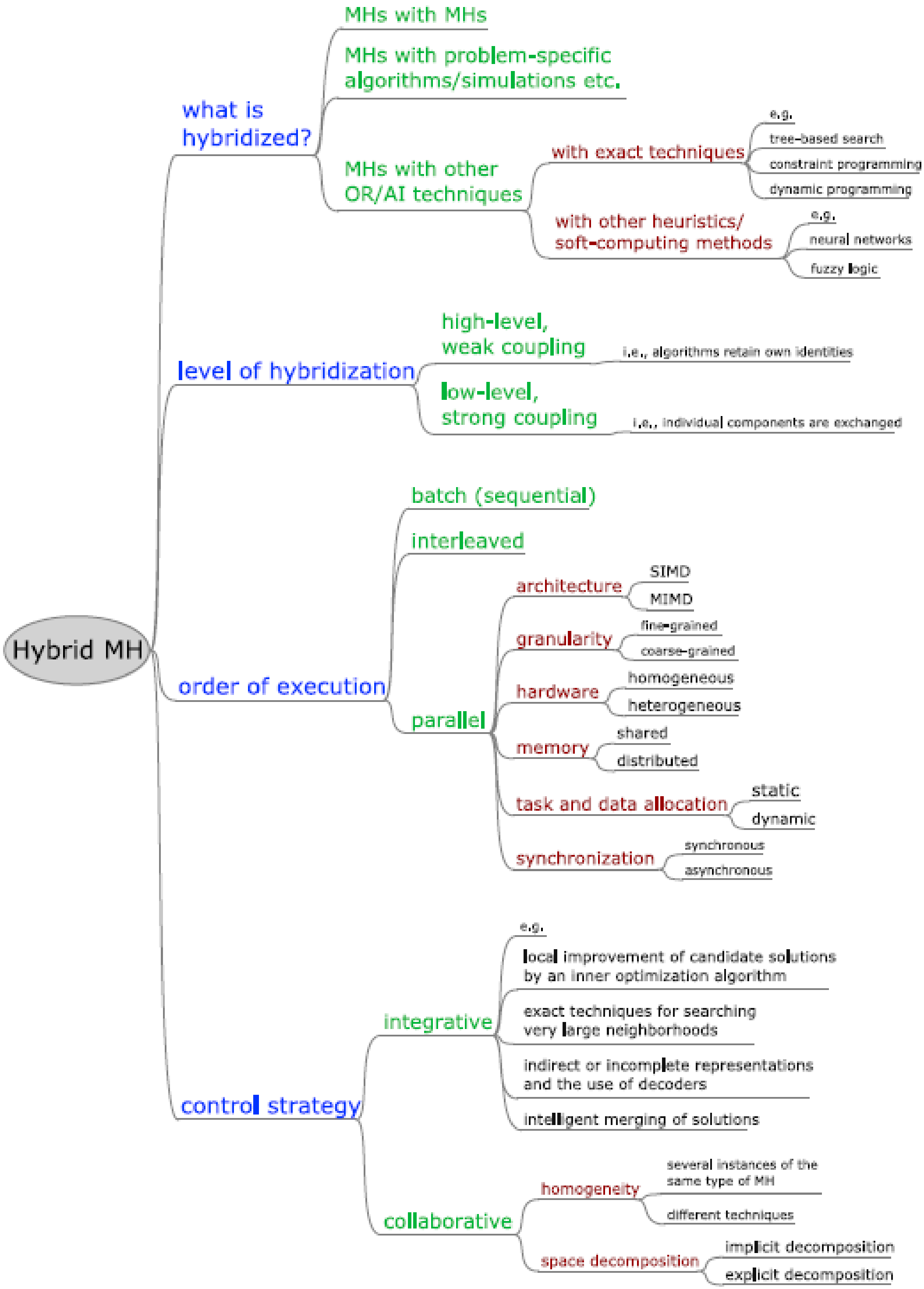


Figure 2.1– Classification proposed by Raidl [Raidl (2006)].

More recently memetic algorithms to deal with MOOPs have been proposed and the research in this area has been growing significantly: Arroyo and Armentano (2005), Liu et al. (2007), Wang and Tai (2007), Wanner et al. (2008), Caponio and Neri (2009), Jaszkiewicz and Zielniewicz (2009), Yoshida and Mori (2009), Ishibuchi et al. (2010), Lara et al. (2010), Nguyen et al. (2012). Updated surveys are available in [Chen et al. (2011), Zhou et al. (2011), Neri and Cotta (2012), Jaszkiewicz et al. (2012)].

In this thesis, the focus is on the hybridization of SA, as a local search method, with other metaheuristics. SA is one of the most successful metaheuristics used in the intensification phase in hybrid population-based metaheuristics [Anghinolfi and Paolucci (2008)]. However, this hybridization in the context of MOOP is not as vast as in single-objective optimization problems. Some recent works are referred below, most of them combining SA with a GA or an EA. Yogeswaran et al. (2007, 2009) propose a HMH using GA and SA to solve machine loading problems in flexible manufacturing systems, in which the system unbalance must be minimized and the system throughput must be maximized. Hui (2010) joins the advantages of SA and GA in a unified algorithm, called ASAGA, to optimize the key component sizes in a hydraulic hybrid vehicle. The optimization problem is formulated considering the fuel consumption, the braking energy regenerative ability, the driving performance and the added price off the hydraulic hybrid vehicle. Despite the conflicting character of these objectives, Hui uses a fitness function grouping them for solution evaluation. In the EMOSA algorithm an adaptive evolutionary multi-objective approach is combined with SA [Li and Landa-Silva (2011)]. EMOSA is an improved version of MOEA/D (multi-objective evolutionary algorithm based on decomposition) which employs SA for the optimization of each subproblem and adapts the search directions (weighting vectors) to increase the diversity of non-dominated solutions. Yannibelli and Amandia (2012) combine a multi-objective SA algorithm and a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. In this problem two conflicting objective functions are considered: to minimize the project makespan and to assign the most effective set of human resources to each project activity. The multi-objective SA algorithm is integrated into the multi-objective evolutionary algorithm to improve the performance of the evolutionary search. Martinez-Martin et al. (2012) use and compare three hybrid multi-objective SA algorithms to the design of reinforced concrete bridge piers. The different approaches differ in the initial temperature and the probability acceptance function to accept worse solutions. The neighborhood move considered in both algorithms is based on the mutation operator of GAs. The concept of non-dominance is used in the evaluation of solutions. In [Zhang et al.

(2013)] a SA technique based multi-objective cultural differential evolutionary algorithm is presented to solve a daily hydrothermal operation scheduling with economic emission problem. The SA algorithm is incorporated into the cultural computational model.

Another hybridization of SA usually found in MOOPs consists in its combination with components incorporating memory to prevent revisiting solutions. This is achieved using TS or some mechanism based on tabu lists. In [Burke et al. (2001)] a HMH is presented for solving a space allocation problem with two conflicting criteria: the misuse of the space and the penalty for the violation of soft constraints. The HMH is based on hill climbing, SA, tabu lists and a mutation operator. A machine loading problem in flexible manufacturing system is solved in [Swarnkar and Tiwari (2004)] using a hybrid algorithm based on TS and SA. In this approach a short-term memory provided by the tabu list can be used to avoid revisiting the solution while preserving the stochastic nature of the SA method. Baños et al. (2007) develop another hybrid algorithm, named MOSATS, that combines SA and TS in a population based context. MOSATS uses a crowding criterion in the probability acceptance function to improve the diversity of the population. This hybrid algorithm is applied for solving a graph partitioning problem. In [Cakir et al. (2011)] a MOO of a single-model stochastic assembly line balancing problem with parallel stations is presented. Two objective functions are considered: the minimization of the smoothness index and the minimization of the design cost. The proposed algorithm based on SA, named m_SAA, uses a multinomial probability mass function approach to decide about the acceptance of worse solutions and also includes a tabu list to prevent revisiting recently searched solutions.

The SA has been combined also with PSO in [Xia and Wu (2005)] for solving a multi-objective flexible job-shop scheduling problem. Three objective functions are considered to minimize: the makespan or maximal completion time of machines, the total workload of the machines, which represents the total working time of all machines, and the critical machine workload, that is the machine with the biggest workload. A weighted sum of these objective functions is used in the evaluation of solution.

Even though, in general, the initial population is generated randomly in EAs, it is known that a good quality of initial solutions can improve the convergence of the algorithm. This has motivated different approaches to the generation of initial solutions. Some of them are based on GRASP construction phase, where solutions are generated in accordance to the problem at hand by exploiting its characteristics. This aspect may be particularly important in real-world problems where the information about the problem may be used in the initial solutions generation. The GRASP algorithm is a multi-start local search approach consisting in two distinct phases that are successively repeated: greedy randomized construction of starting solutions and a local search procedure (details about this metaheuristic will be presented in Chapter 4). In MOOPs, GRASP is often hybridized with path-relinking to introduce a memory structure. GRASP with hybrid path relinking for bi-objective winner determination in combinatorial transportation auctions is presented in [Buer and Pankratz (2009)]. In this work, a Pareto-based GRASP is introduced with a post-optimization procedure that hybridizes truncated path relinking with exact branch-and-bound. Another hybrid algorithm combining GRASP with path relinking is proposed in [Alpay (2009)]. In this case, the hybrid algorithm is used to address a production sequencing problem for mixed-model assembly line in a just-in-time production system. Two objective functions are considered: the minimization of setups and the maximization of stability of material usage rates. Despite the good performance of GRASP, Alpay refers that the GRASP performs poorly with regard to CPU time. Ishida et al. (2009) present the hybridization of GRASP with path relinking to create rules that together have good performance for classification. Marti et al. (2011) propose different hybridizations of GRASP and path-relinking for multi-objective optimization: a bi-objective orienteering problem and a bi-objective path dissimilarity problem. In [Kafafy et al. (2011)] a hybrid evolutionary metaheuristics applied on 0/1 multi-objective knapsack problems is proposed. GRASP combined with data mining techniques is used to obtain an initial set of high quality solutions dispersed along the Pareto-optimal front and then a greedy randomized path-relinking with local search or reproduction operators are applied to improve the quality and to guide the search to explore new regions in the search space. An overview about hybridizations of GRASP with path-relinking is performed in [Festa and Resende (2013)].

GRASP has been also hybridized with other metaheuristics in the resolution of single-objective optimization problems [Resende (2008), Festa and Resende (2009c), Resende and Ribeiro (2010)]. However, few works with GRASP hybridization in MOOPs have been published. Lourenço et al. (2001) propose two hybrid algorithms to solve a bus driver scheduling problem: one based on the TS and the other based on genetic algorithms. In both approaches, GRASP is used as a procedure within these multi-objective algorithms. In the genetic algorithm, GRASP is used to define a new crossover operator called perfect offspring. In [Hanoun and Nahavandi (2012)] two objective functions are considered in the resolution of a flowshop scheduling problem. A greedy heuristic and SA are used, but a hierarchical optimization approach is followed: GRASP is used in the minimization of material waste and then SA is used in the minimization of the total tardiness time.

Considering that a HMH can also be viewed as an algorithm that not only combines components of different metaheuristics but also incorporates some of their characteristics, the mGRASP/MH algorithm presented in [Li and Landa-Silva (2009)] can be considered as a hybrid algorithm. Li and Landa-Silva present a GRASP approach including elitism, cooperation between solutions and weight-vector adaptations to accelerate convergence and diversify the search. In the construction phase of GRASP, the algorithm uses not only problem-specific greedy information but also the elite solution found in the previous local search.

The hybridized approaches proposed in this thesis consist in a Hybrid Evolutionary Simulated Annealing (HESA) algorithm that has been developed in the spirit of a memetic approach, combining EvABOR-III (an Evolutionary Algorithm Based on Outranking Relation) with a local search method. SA has been used as the local search method to intensify the search of solutions belonging to the current best class of merit. The second hybrid approach combines GRASP with SA. The underlying idea is to generate better solutions than the ones created randomly and explore these solutions using a local search method. The incorporation of preferences is also included in these two hybrid algorithms in the local search phase.

CHAPTER 3

EVOLUTIONARY ALGORITHMS BASED ON AN OUTRANKING RELATION FOR PREFERENCE INCORPORATION[3]

In real-world MOOP, the dimension of the search space is usually very large and irregular due to the number of objective functions to be evaluated and the non-linear and/or combinatorial characteristics of the mathematical model. This may lead to a prohibitive computational effort for characterizing the non-dominated front or even obtaining a well spread non-dominated solution set. Besides, when the number of conflicting objectives to be dealt with increases, the number of non-dominated solutions also increases significantly. Despite the success of EAs in dealing with these issues [Coello et al. (2002), Deb (2001)], whenever a high number of non-dominated solutions exists the selection operator is usually less effective and the selection of solutions to the next generations becomes practically random thus slowing the evolutionary process [di Pierro et al. (2007), Garza-Fabre et al. (2009), Deb et al. (2010)]. This further complicates the practical exploitation of results in real-world problems when a solution (or a small set of solutions for further screening) must be chosen, due to the large number of solutions in the non-dominated front that generally occurs.

The difficult characteristics of most real-world MOOPs and the associated issues mentioned above require methodological tools to improve the efficiency and the efficacy of the solution search methods. The incorporation of preferences is one of the most used approaches to improve these aspects. The incorporation of preferences in EAs aimed at providing decision

[3] This chapter is partially based on [Oliveira E, Antunes C H, Gomes A. A comparative study of different approaches using an outranking relation in a multi-objective evolutionary algorithm. Computers & Operations Research 2013; 40 (6): 1602–1615] and [Oliveira E, Antunes C H, Gomes A. Incorporation of preferences in an evolutionary algorithm using an outranking relation - the EvABOR approach, International Journal of Natural Computing Research 2011; 2 (1), 63–85].

support in real-world problems presents two main advantages: it contributes to reducing the computational effort by focusing the search on regions of the search space that appear more interesting according to the preferences elicited from a DM, and it reduces the cognitive effort imposed on the DM by offering him/her solutions more in accordance with those expressed preferences and therefore displaying, in principle, more satisfactory trade-offs between the competing objectives [Branke and Deb (2004), Coello (2000)]. As a result, the overall efficiency of the algorithm is increased, as well as the effectiveness of the decision support process, since the search process has been guided towards a final non-dominated solution (or a reduced set of the most preferred solutions) according to meaningful preference expression mechanisms having in mind a practical implementation.

An EA, which incorporates and makes the preferences elicited from a DM operational during the search process, by means of the technical parameters of the ELECTRE TRI method, has been developed.

The introduction of the preference expression parameters used in the ELECTRE TRI method has revealed to be suitable both from the point of view of meaningfulness of preference elicitation and its use in the operational framework of an EA. The different versions of the EA developed (called EvABOR, Evolutionary Algorithm Based on an Outranking Relation) include features of the ELECTRE TRI method to guide the search according to the preference information expressed and use an outranking relation and the concept of classes of merit to generate the population for the next generation. Preferences are herein represented by means of technical parameters: weights, indifference, preference and veto thresholds, a set of references profiles and a cutting level (which may be perceived as the level of exigency of the classification). The weights reflect the true importance of each objective function (its "voting power") and are not scaling coefficients to achieve some aggregate value. The veto threshold enables to preclude situations often arising in real-world problems in which full compensation between the objective function values is undesirable or even unacceptable. The indifference and preference thresholds enable to introduce a gradual preference relation. The reference profiles define the classes of merit in which the solutions are classified, as explained in the next section, and the aim of the EvABOR approaches is to obtain solutions belonging to the best class of merit as much as possible.

3.1. THE ELECTRE TRI METHOD

The ELECTRE (*ELimination Et Choix Traduisant la REalité*) family of multi-criteria methods developed by Roy and his co-workers [Roy (1996)] is based on the construction and the exploitation of an outranking relation. The term outranking in this context means "is at least as good as" or "not worse than" [Roy (1991)]. The ELECTRE methods may be classified according to the type of the problem each one deals with: choice, sorting and ranking. The choice problem refers to the identification of the best alternative (solution) or a limited subset of the best alternatives (since incomparability is allowed). The ranking problem deals with the establishment of a partial or complete pre-order of the alternatives from the best to the worst one. The sorting problem consists in assigning each alternative to predefined ordered categories (classes of merit). The major difference between these formulations concerns the judgment of the alternatives [Mousseau et al. (2000), Zopounidis and Doumpos (2002)]. The choice and ranking approaches are based on relative judgments and consequently the evaluation depends entirely on the set of alternatives considered. The sorting approach considers an absolute judgment, in the sense that the pair-wise comparisons are made between the alternative to sort and a set of alternatives defined by the DM named reference profiles. This presents two important advantages. Firstly, since the number of reference profiles is in general much lower than the number of alternatives, significantly fewer comparisons must be done. The second issue is concerned with the quality of the alternatives. In the ranking problem the set of alternatives is partially or completely ordered; however, the quality of all alternatives may not be good enough according to the preferences elicited. A similar situation may arise in the choice problem if the set of alternatives does not comply with the DM's preferences; the "best" alternative will be chosen, but it could not be a good enough alternative according to the preferences expressed.

To guarantee the quality of solutions and reduce the number of pair-wise comparisons, the ELECTRE TRI method has been chosen to incorporate the preferences in the evolutionary process. Since ELECTRE TRI deals with the sorting problem, the comparisons are made between the alternatives and the reference profiles. Consequently a measure of the quality of the solutions is provided, which is given by the class of merit each solution is assigned to (a solution belonging to the best class of merit is preferred). Each category, C^i, i = 1,…, n, is limited by two reference alternatives (profiles), b^{i-1} and b^i, i = 1,…, n, defined for each criterion (objective function in the EA context) g_j, j = 1,…, m (Figure 3.1). We assume, without any loss of generality, that C^n is the best category and C^1 the worst category. An alternative a_1 that verifies $b^{i-1} < g_j(a_1) < b^i$ for all criteria g_j is assigned to the category C^i. However, in real-world problems the conflicting nature of the criteria leads generally to alternatives with a good performance in some criteria and a bad performance in others. Consequently the situation mentioned for alternative a_1 is rarely obtained and the other ELECTRE TRI parameters (thresholds, weights and a cutting-level) provide additional information about the preferences and the exigency of the classification. If an alternative a_2 verifies, for example, $b^{i-1} < g_j(a_2) < b^i$, for all criteria g_j except one, for instance g_1, the class to which it is going to be assigned depends on the relative importance (weight) of each criterion g_j, the difference between $g_1(a_2)$ and the reference profiles that bound the class C^i and the value of the cutting-level. Some examples about the assignment of a solution to a class of merit depending on the values of the ELECTRE TRI parameters can be found in Appendix A.

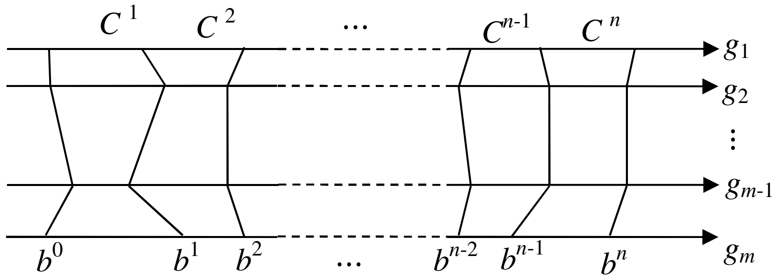


Figure 3.1 - Definition of the classes of merit in ELECTRE TRI.

For each reference profile defined for each criterion, a set of indifference, preference and veto thresholds is required. The aim of indifference q_j and preference p_j thresholds is to introduce the acceptance of some imprecision when comparing two alternatives by considering them as indifferent if their individual performances in each criterion g_j differ less than q_j. The transition from indifference to preference changes gradually in a linear manner from q_j to p_j (Figure 3.2). This acceptance is translated by the criterion concordance indexes c_j and is evaluated for each criterion (Equation 3.1). The veto thresholds v_j prevent an alternative having a good performance in one or more criteria but having a very bad performance in another criterion to be assigned to the best category, or they force this alternative to be assigned to a low preference category independently of having very good performance in all other criteria, thus allowing for the introduction of some non-compensatory aspects in the decision. The transition from p_j to v_j changes also in a linear manner.

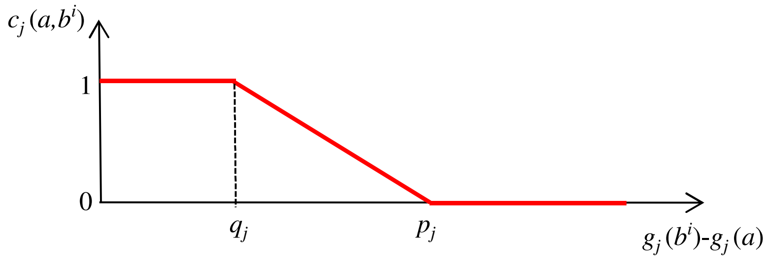


Figure 3.2 - Computation of criterion concordance index c_j.

$$\boldsymbol{c_j(a, b^i) = min\left\{1, max\left(0, \frac{g_j(a)-g_j(b^i)+p_j}{p_j-q_j}\right)\right\}} \tag{3.1}$$

Other parameter of ELECTRE TRI is the weight w_j associated to each criterion. The weights in ELECTRE TRI are scale-independent; that is, they are not linked to the scales in which each criterion is measured. In this framework the weights are not technical devices for translating the performances in the criteria into a common value measure (as in value-based approaches) but weights play the role of true importance coefficients (the voting power) of each criterion.

The level of exigency (the majority requirement of criterion "coalitions") to enforce the assignment of a given alternative to a category is stated by a real value, the cutting-level λ, defined in the interval [0.5,1].

In ELECTRE TRI the evaluation is done in two main phases: the building of a fuzzy outranking relation *S* and the exploitation of this relation with the purpose of assigning the solutions to the classes of merit. The outranking relation *S* validates or invalidates the assertion aSb^i meaning "a is at least as good as b^i" as referred before. Four main stages may be distinguished in establishing the outranking relation *S*, which are associated with the evaluation of:

1. Criterion concordance indexes $c_j(a, b^i)$;
2. Global concordance indexes $C\left(a, b^i\right)$;
3. Criterion discordance indexes $d_j(a, b^i)$;
4. Credibility degree $\sigma(a, b^i)$;

The assertion aSb^i is stated based on the concordance and the discordance principles. The concordance principle determines if there are sufficiently strong reasons to confirm this assertion and the discordance principle states if there are no impeditive reasons to contradict it. The concordance principle is evaluated using the criterion concordance indexes (Equation 3.1) indicating how much each criterion agrees with the assertion aSb^i and the global concordance indexes (Equation 3.2) quantifying the relative importance of the coalitions of criteria that are in favor of the previous assertion. The discordance principle is evaluated based on the criterion discordance indexes (Equation 3.3).

$$\boldsymbol{C(a, b^i)} = \frac{\sum_{j=1}^{m} w_j . c_j(a, b^i)}{\sum_{j=1}^{m} w_j} \tag{3.2}$$

$$\boldsymbol{d_j(a, b^i) = min\left\{1, max\left(0, \frac{g_j(b^i) - g_j(a) - p_j}{v_j - p_j}\right)\right\}} \tag{3.3}$$

The last step of the ELECTRE TRI method consists in the assignment of each alternative to a class of merit. An assertion aSb^i is considered to be valid if the value of the credibility degree σ(a,b^i) is greater than the cutting-level λ. The credibility degree is given by

$$\boldsymbol{\sigma(a, b^i) = C(a, b^i) \prod_{j \in F} \frac{1 - d_j(a, b^i)}{1 - C(a, b^i)}} \tag{3.4}$$

where *F* is the subset of the criteria for which the discordance index (Equation 3.3) is larger than the global concordance index $C\left(a, b^i\right)$.

Two assignment procedures (optimistic and pessimistic) may be adopted in the ELECTRE TRI method. In approaches proposed in this thesis the pessimistic procedure is applied so an alternative is assigned to the highest class of merit C^i for which the alternative outranks the lower bound reference profile of this class.

The ELECTRE TRI method requires a significant set of parameters that embodies the DM's preferences. These parameters must be elicited from DMs preferably via an analyst with expertise in this methodology. In the context of a real-world problem, some parameters may be predetermined according to the experience associated with previous studies. For instance, indifference and preference thresholds can be fixed as percentages (say 2% and 10%, respectively) of the value ranges in each class.

For further operational details about the ELECTRE TRI method see Mousseau et al. (2000).

3.2. The EvABOR Approaches

In the **Ev**olutionary **A**lgorithm **B**ased on an **O**utranking **R**elation (EvABOR) approaches the DM's preferences are captured and incorporated into the EA using the outranking relation in the same spirit as in the ELECTRE TRI method. The main structure of EvABOR is the usual structure of an EA, including the crossover and the mutation operators followed by the selection of the individuals of the next generation (Figure 3.3). The DM's preferences captured by the ELECTRE TRI parameters are incorporated in all the genetic operators (crossover, mutation and selection) in order to guide the search to a region of the space in which solutions more in accordance with those preferences are located.

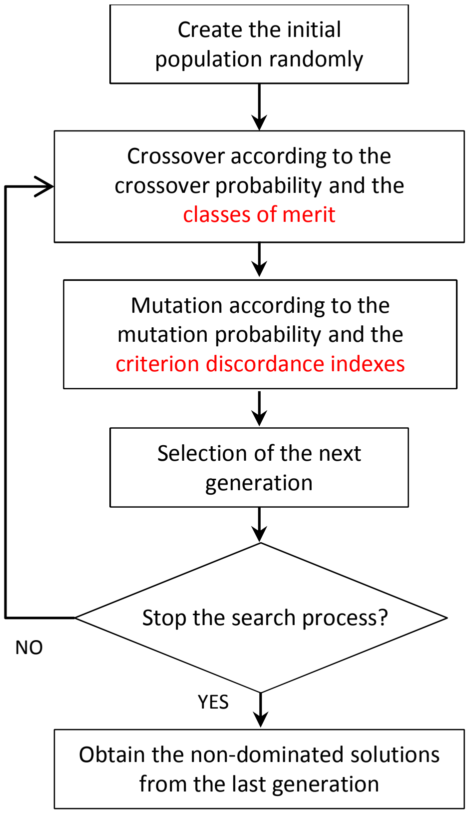


Figure 3.3 – Flowchart of the EvABOR approaches.

Three distinct EvABOR approaches have been developed. The main differences between these approaches lie on the selection of the individuals (from the progenitors and offspring set) that go to the next generation. EvABOR-I uses the outranking relation only, and consequently the ordered classes of merit, to select the solutions (dominated and non-dominated) to be carried to the next generation. In the other two approaches a symbiosis between the outranking and the non-dominance relations is used. EvABOR-II uses the ordered classes of merit considering only the non-dominated solutions in each class, whereas EvABOR-III uses the ordered classes of merit after a non-dominance test is applied to the whole population. This means that in EvABOR-II the priority is given to the outranking relation and in EvABOR-III it is given to the non-dominance relation in the selection of the next generation.

In all approaches it is possible to define a level of intra-class elitism, in the sense of favoring each objective function, by means of the specification of a real parameter $\beta \in]0,1]$. When there are more individuals in a class than the necessary to complete the next generation, a percentage of the best solutions for each objective function is chosen. This percentage is proportional to the weight (the importance) of each objective function. A value of $\beta=0$

means that no intra-class elitism is applied. At the end of the evolutionary process, if it is necessary, the non-dominance relation is used to obtain the set of non-dominated solutions in the region of interest.

3.2.1. Crossover Operator

In EvABOR approaches, the idea underlying the crossover operator is that parents that better fulfill the DM's preferences are more likely to be chosen to produce offspring with characteristics similar to parents. Consequently, a solution belonging to the best class of merit has a higher probability of being selected as a parent. Parent individuals are selected using a binary tournament procedure similar to the one in NGSA-II [Deb et al. (2002)] but the outranking relation replaces the non-dominance relation. The binary tournament has the advantage that all solutions participate in the competition to obtain the parents. When two solutions compete to be one of the parents, the one that belongs to the best class of merit is chosen. If both solutions belong to the same class of merit then the one having the best performance in most objective functions is selected (Figure 3.4). The crossover operator is applied according to the crossover probability and a 2-point crossover has been implemented to obtain the offspring set.

```
p_1 = permutation of the population
p_2 = other permutation of the population
dim_pop = dimension of the population
for i =1 to dim_pop
    if class of  (p_1(i)) > class of  (p_2(i))
       parent_1 = p_1(i)
    elseif class of  (p_1(i)) < class of  (p_2(i))
       parent_1 = p_2(i)
    else choose the solution that has better values for more
          objective functions
    end
end
```

Figure 3.4 – Pseudo-code to choose one of the parents for the crossover operator in EvABOR approaches.

3.2.2. Mutation Operator

The mutation operator is applied, according to the mutation probability, to favor the diversity of the population. To achieve this aim more effectively, several mutation operators are implemented being this mutation operator portfolio strongly dependent on the physical characteristics of the problem to be tackled. The choice of one of these operators is based on the criterion discordance indexes (Equation 3.3). These values provide information about which objective function has the worst performance according to the DM's preferences, and consequently it is possible to select a more adequate mutation operator attempting to improve the objective function with the worst performance. That is, mutation operators that are likely to improve this objective function have a greater probability of being chosen (although the process remains essentially a random one). As criterion discordance indexes are evaluated using the reference profiles as well as the preference and veto thresholds elicited from the DM, the choice of the mutation operator is indeed guided by the DM's preferences. It is important to emphasize that this approach is essentially useful when dealing with real-world problems in which the mutation operators can be assigned a physical meaning.

3.2.3. Selection of the Next Generation

The main differences between the three versions of the EvABOR algorithm are in the selection of the individuals that pass to the next generation. In EvABOR-I only the outranking relation is used to select those individuals. In EvABOR-II and III a symbiosis between the non-dominance relation and the outranking relation is considered (Figure 3.5).

In EvABOR-I parents and offspring are sorted into classes of merit using the ELECTRE TRI method. The solutions belonging to the best class of merit pass to the next generation. If this number of solutions is not sufficient to complete the next generation, the solutions belonging to the next class of merit pass to the next generation. This process is repeated until the next generation is completed or a class has more solutions than is necessary to complete the next generation. When this situation occurs, the necessary number of solutions is randomly selected from the respective class or using an elitist process depending of the value of an EvABOR parameter $\beta \in [0,1]$, which regulates the degree of intra-class elitism.

The zero value means that the solutions are chosen randomly from the respective class. If $\beta = 1$ an intra-class elitist process based on the weight of the objective functions is applied. The number of solutions chosen is proportional to the weight of each objective function. For example, for three objective functions f_1, f_2 and f_3 with weights 0.45, 0.30 and 0.25, respectively, EvABOR chooses 45% of the best solutions according to objective function f_1, 30% of the best solutions according to objective function f_2, and 25% of the best solutions according to objective function f_3. The aim is to favor the objective function with higher weight (importance) according to the preferences elicited. A value of $\beta \in]0,1[$ means that $(1-\beta).NS$ solutions are randomly chosen and $\beta.NS$ are chosen from the ones that have the best values for each objective function (always proportionally to the weight of each objective function). *NS* denotes the necessary number of solutions to complete the next generation.

In EvABOR-II, the selection of the next generation is similar to the one described for EvABOR-I, but only the non-dominated solutions of each class of merit are selected to pass to the next generation. The non-dominated solutions are selected beginning from the best class of merit until the next generation is completed. If the last class used has more non-dominated solutions than necessary, the solutions are picked randomly or in an elitist manner depending of the value of β, as described before.

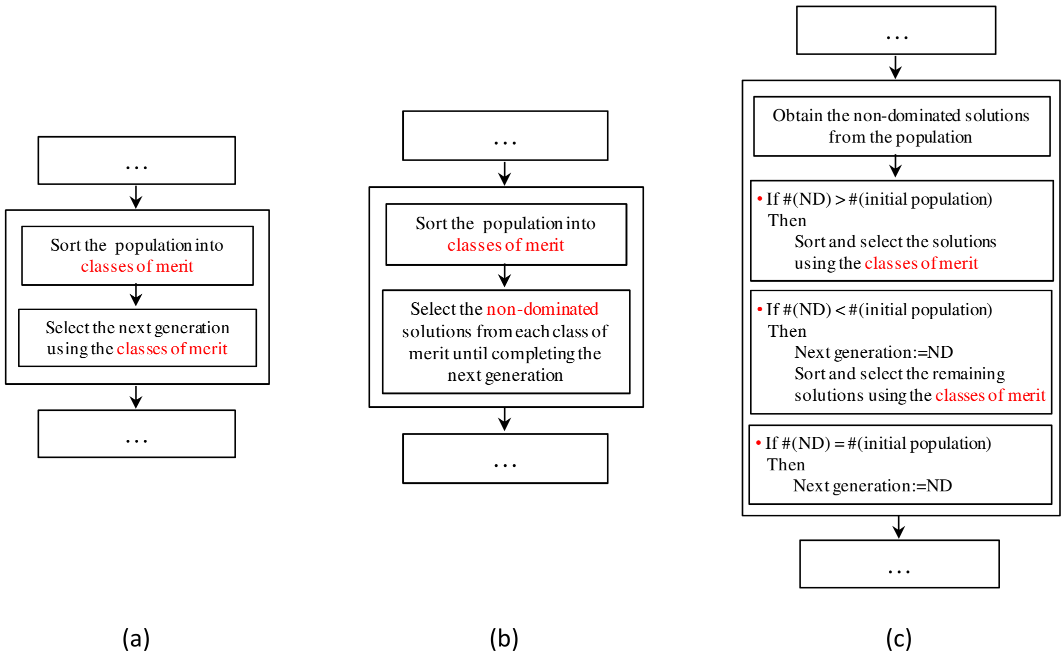


Figure 3.5 – Selection of the next generation in: (a) EvABOR-I; (b) EvABOR-II and (c) EvABOR-III.

In this second approach, dominated solutions can be included in the next generation because the non-dominated solutions picked from different class of merit may display dominance between them. In EvABOR-II the dimension of the population can be variable if the total number of non-dominated solutions in all classes is inferior to the initial dimension of the population.

Contrarily to the other two approaches in which priority is given to the outranking relation, the selection in EvABOR-III gives priority to the non-dominance relation but enriches it with the outranking relation. Initially, the set of non-dominated solutions is selected from the population (parents and offspring). The next procedure depends on the number of non-dominated solutions in the population:

- If the number of non-dominated solutions is greater than the dimension of the population then the non-dominated solutions are sorted into classes of merit using the outranking relation, and the process to select the solutions passing to the next generation is the same as described above in EvABOR-I (but non-dominated solutions only are considered). A scheme illustrating this approach is presented in Figure 3.6.
- If the number of non-dominated solutions is inferior to the dimension of the population then all non-dominated solutions pass to the next generation and the remaining solutions (dominated solutions) in the population are sorted into classes of merit using the outranking relation. The next generation is completed by selecting the dominated solutions from the best classes of merit using the process described for the other versions of EvABOR.
- Finally, if the number of non-dominated solutions is equal to the dimension of the population then the next generation consists of these solutions.

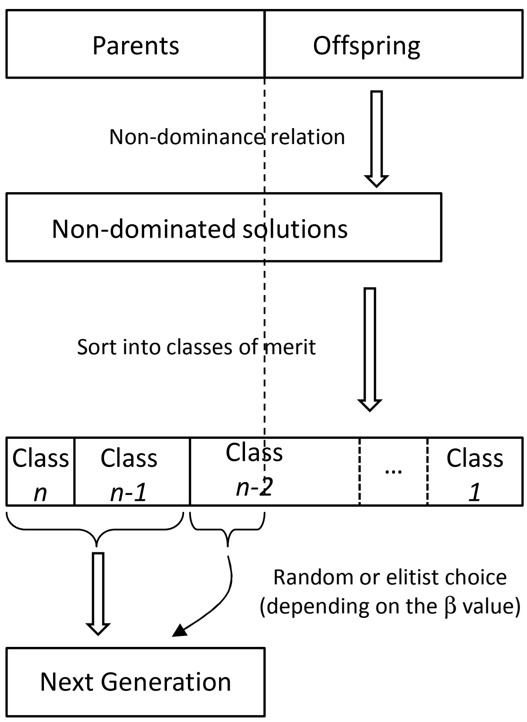


Figure 3.6 – Scheme of selection of the next generation from non-dominated solutions in EvABOR-III.

The structure of EvABOR approaches is modular, which means that the algorithms are prepared to use other type of multi-criteria approaches to replace (or to complement) the method used in this work. In these circumstances, operators that use ELECTRE TRI may be replaced. For instance, the function that selects the parents to the crossover, the function that identifies the mutation operator within the portfolio and the one that determines the individuals that pass to the next generation may be replaced.

In each generation the outranking relation is assessed using the performances of each individual according to each objective function and the corresponding values of each reference profile. Consequently, for a population of *m* individuals, *r* reference profiles and *k* objective functions the computational effort of the outranking relation involves *m.r.k* comparison operations. It is important to note that *r* and *k* values are, in general, low. One advantage of ELECTRE TRI over other techniques is the lower number of pairwise comparisons required, since these are made between each individual and the reference profiles rather than between all individuals.

3.3. A Reactive Power Compensation Problem

3.3.1. The Case Study

Reactive power compensation is a relevant problem in electrical distribution systems to guarantee an efficient delivery of active power to loads also contributing to releasing system capacity, reducing system losses and improving system power factor and bus voltage profile. A reactive power compensation problem involves determining the size and location of capacitors (local sources of reactive power) in nodes of the electrical distribution network [Antunes et al. (2009)].

Mathematical programming techniques have been proposed to deal with this problem generally requiring some less practical assumptions about the network characteristics in order to facilitate computational manageability due to the intrinsic combinatorial nature of the reactive power compensation problem [Iba et al. (1988), Delfanti et al. (2000)]. Since different types of capacitors can be installed in different network nodes, a very large number of potential solutions must be evaluated in general. This led to the use of metaheuristic approaches to compute solutions in the Pareto-optimal front. Tabu Search, Simulated Annealing, Ant Colony Optimization, Particle Swarm Optimization, and Evolutionary/Genetic Algorithms have been used to tackle this problem, considering single and multi-objective models [Zhang et al. (2007)]. In this thesis, the results obtained with EvABOR approaches are presented and analyzed. Having in mind a practical implementation, the incorporation of preferences elicited from a DM in the EA may be an attractive approach to guide the search to regions more in accordance with preferences elicited.

The merit of potential solutions to this problem must be assessed using operational, economical and quality of service aspects. Therefore, multiple objective programming models have been used to provide decision support in this problem considering incommensurate and conflicting objective functions to be optimized [Zhang et al. (2007)]. In this work the minimization of resistive losses, capacitor installation cost and maximum deviation to the nominal voltage at each network node have been considered. The integer-valued decision (control) variables indicate the type of capacitor, characterized by a capacity value and a cost, to be installed in each network node. The real-valued decision variables refer to the active and reactive power as well as voltage at each network node. Constraints are related to requirements of acceptable node voltage profile (quality of service), power flow (physical laws in electric networks), and impossibility of capacitor location at certain nodes (technical restrictions). For further technical details on the multi-objective mathematical model and the power flow algorithm see Antunes et al. (2009) and Pires et al. (2012) where a mathematical model considering only two objective functions (cost and resistive power losses) has been developed.

The network used as a case study is an actual Portuguese radial electrical distribution network with 94 nodes (Figure 3.7). The node zero is the sub-station and other nodes indicate the load demand points or derivations for lateral buses, in which capacitors may be installed. This network has some challenging features due to its extension in a rural region and a poor voltage profile. The study is done for peak load conditions, in which the active power losses are 320.44 kW and the number of nodes not respecting the voltage lower bounds is 66 in 94 nodes (the node’s voltage magnitude must be within ±10% of the nominal value). That is, the network is not working according to regulations in peak load conditions and therefore capacitors must be installed for reactive power compensation and voltage profile improving purposes. This means that the zero cost solution is not feasible. The solutions obtained must be practicable, so the number of capacitors to be installed has been limited (in this case, the maximum number of capacitors that can be installed is 20).

Eight types of capacitors are considered (Table 3.1) for possible installation with capacities in the range of 50 to 400 kVAr and cost in the range of 2035€ to 9395€ (taken from a catalog of an electric equipment supplier).

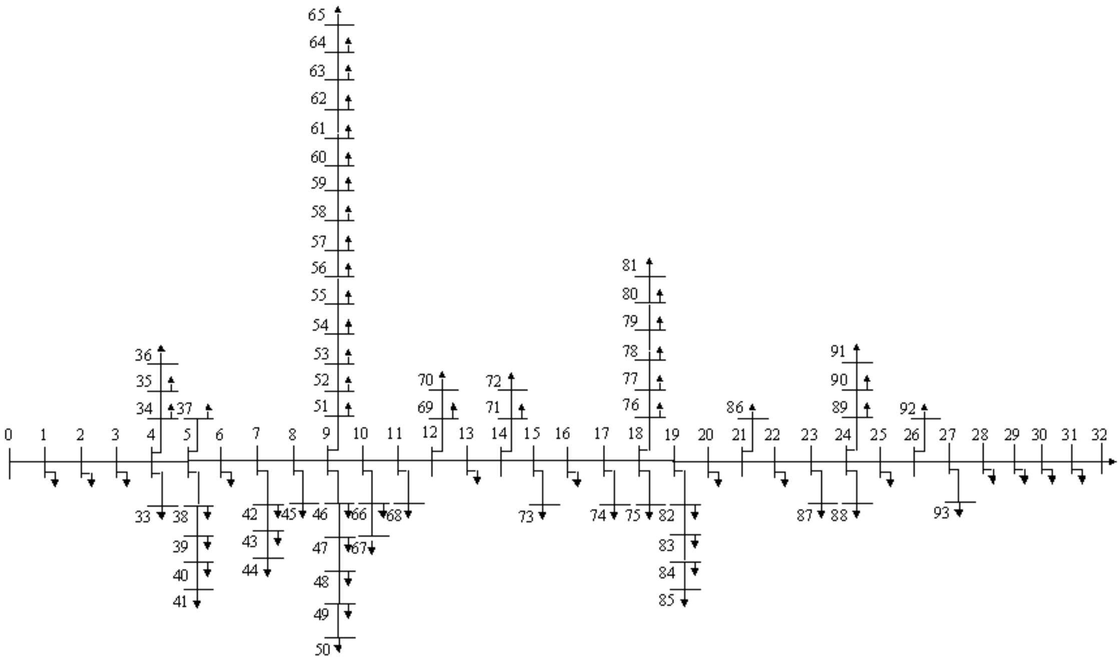


Figure 3.7 – Actual electrical distribution network used.

	Maximum Capacity (kVAr)	Cost (Euro)
C1	50	2035
C2	100	2903
C3	140	4545
C4	200	4875
C5	240	5716
C6	300	6578
C7	360	7337
C8	400	9395

Table 3.1 – Capacitor dimension and acquisition cost.

3.3.2. IMPLEMENTATION DETAILS

A solution (compensation scheme) to this problem is encoded as an array of integers with each element associated with a network node (Figure 3.8). A “0” means that no capacitor is installed in the corresponding node and an integer value refers to the type of the capacitors therein installed (8 different types of capacitors in this case).

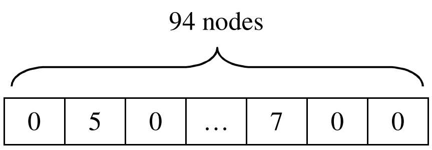


Figure 3.8 – Solution encoding.

Initially a random population is generated retaining just the feasible individuals (solutions that satisfy the voltage lower bound requirements at the nodes). During the evolutionary process a power flow algorithm computes active and reactive power, as well as the voltage, at each network node resulting from a given compensation solution associated with a location and size of capacitors.

At each iteration parents are selected using a tournament technique. Other selection techniques, such as roulette wheel, were tested, but with worse overall results. A 2-point crossover operator (Figure 3.9) is applied.

Solution A

0	0	2	7	0	5	...	0	3	0	1	0	0

Solution B

0	0	3	4	5	7	...	0	0	1	0	3	0

Offspring

0	0	2	7	5	7	...	0	0	1	1	0	0

Figure 3.9 – Example of the crossover operator.

Six types of mutation operators have been implemented specifically for this problem (in the sense that a physical meaning in the electrical network can be associated with each of them, thus also conveying information about the actual problem):

1. Reducing the capacity of a capacitor previously installed in a given node to the immediately lower size (Figure 3.10 (a));

2. Increasing the capacity of capacitor previously installed in a given node to the immediately upper size (Figure 3.10 (b));

3. Removing the capacitor previously installed in a given node (Figure 3.10 (c));

4. Installing a new capacitor in an uncompensated node (Figure 3.10 (d));

5. Relocating the capacitor previously installed in a given node to an adjacent node (Figure 3.10 (e) and (f)).

As referred to in Section 3.2.2, one of these mutation operators is chosen according to the values of the criterion discordance indexes. The operators that have more probability of improving the worst objective function according to the preferences elicited have more probability of being chosen. However, it is important to note that the process remains mostly random because some of these operators determine the sense of variation of some objective functions but other operators do not have the same behavior. For instance, when the size of a capacitor is increased or the number of capacitors in the network is increased the cost objective function will be degraded, but it is not possible to guarantee that the losses objective function improve (because this depends on several characteristics of the network and on the capacitors already installed).

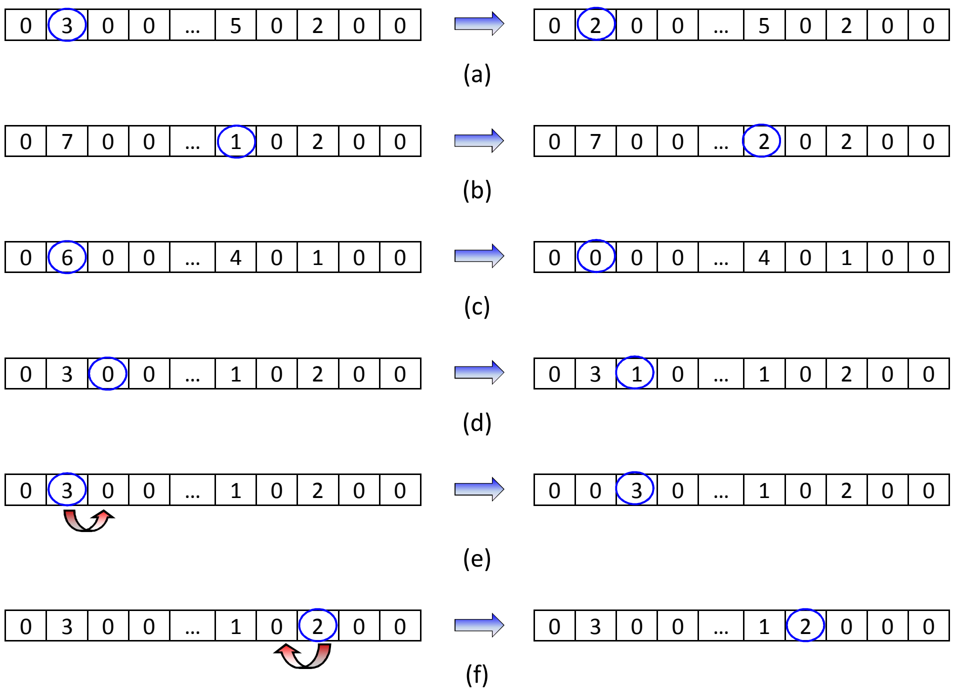


Figure 3.10 - Examples of the mutation operators.

After the crossover and the mutation operators are applied, the selection of the solutions for the next generation is done according to the version of EvABOR to be applied.

The unfeasible solutions that may arise during the search process are discarded from the population. Repairing procedures have been tested but they revealed too expensive computationally.

3.3.3. Analysis of Results

Solutions to the reactive power compensation problem described in the previous section have been computed using the EvABOR approaches as well as an EA with the ELECTRE TRI method applied *a posteriori*.

A comparison between the three EvABOR approaches has been done with the aim of discovering which is more effective to find a set of solutions more in accordance with the preferences elicited from a DM. These solutions are also compared with a set of non-dominated solutions obtained using the EA with preferences elicited *a posteriori*.

The behavior of EvABOR approaches using different values of the intra-class elitist parameter β is analyzed as well as some results related to the crossover operator. Some illustrative examples are also presented to clarify how ELECTRE TRI parameters affect the evolutionary process, and their influence on the classification of the obtained solutions.

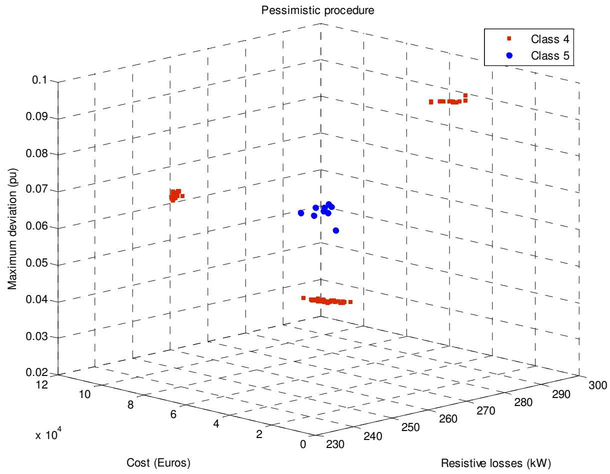
The algorithms have been tested using different parameters for the EA. After a preliminary tuning phase, the crossover probability is set equal to 1 and the mutation probability equal to 0.2.

The Intra-Class Elitist Behavior on EvABOR Approaches

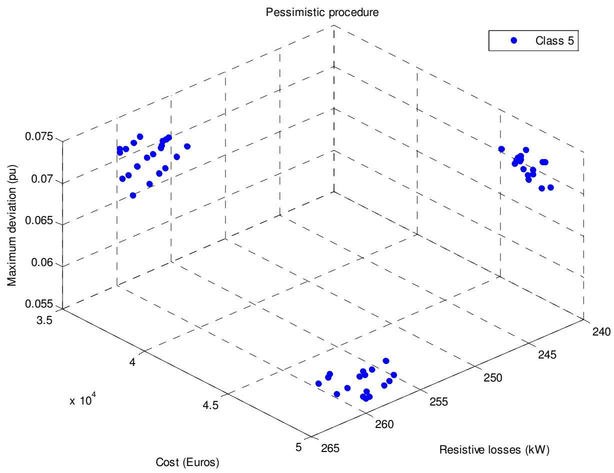
The β parameter controls the level of intra-class elitism of EvABOR approaches when solutions are chosen within the same class of merit. If the value of β is 0 this means that solutions are randomly picked from the different classes. The increase of this value also increases the intra-class elitism of EvABOR approaches. From the computational experiments carried out it is possible to infer that values of β near or equal to 1 lead to a sort of speciation process in which niches of solutions are created, located near the areas of the search space where the objective functions attain their individual optima, slowing the evolutionary process.

In EvABOR-I this effect is visible observing the evolution of the front along the evolutionary process (Figure 3.11 (a)-(c)). This excessive intra-class elitist pressure, β=1, causes a loss of solution diversity as well as premature convergence. The niche effect also arises in the other two EvABOR approaches when considering β=1. However, for the same maximum number of iterations this effect tends to disappear in EvABOR-II and III (Figure 3.12).

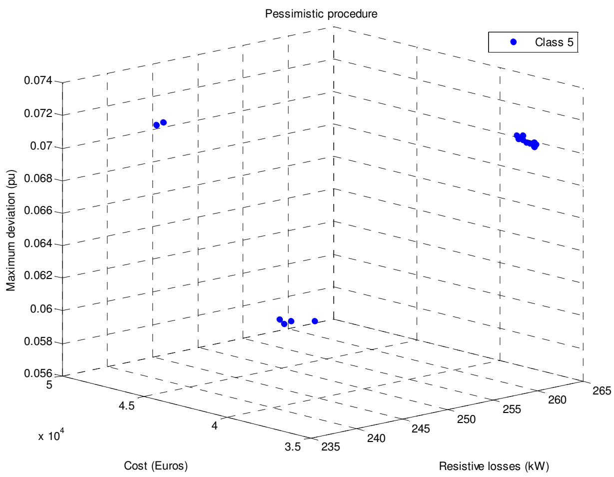
To understand the different behavior of EvABOR approaches concerning the intra-class elitist pressure, an analysis characterizing the population during the evolutionary process has been done. The number of non-dominated solutions and the number of remaining solutions in the population from one generation to the next during the evolutionary process are analyzed for different values of β.



(a)



(b)



(c)

Figure 3.11 – Examples of non-dominated solutions obtained with β=1 using EvABOR-I.

(a) Iteration 50; (b) Iteration 100; (c) Iteration 200.

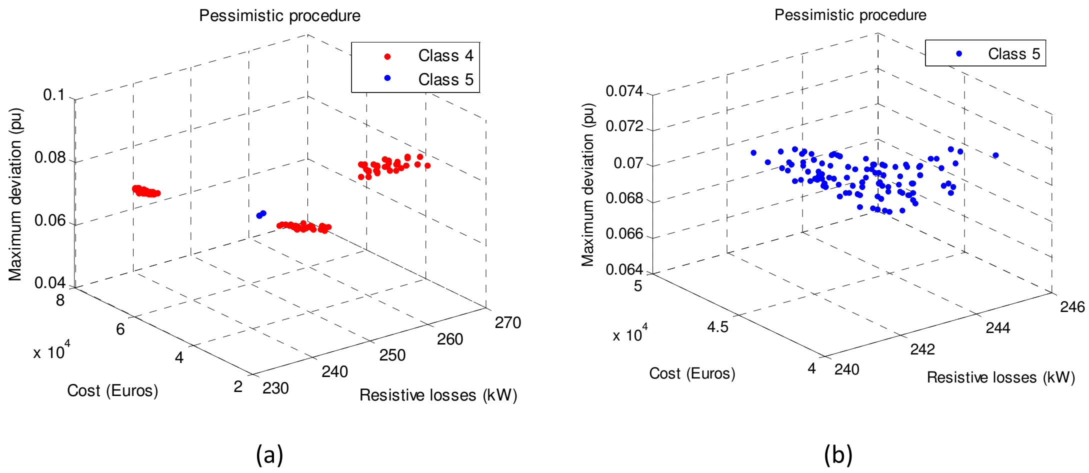


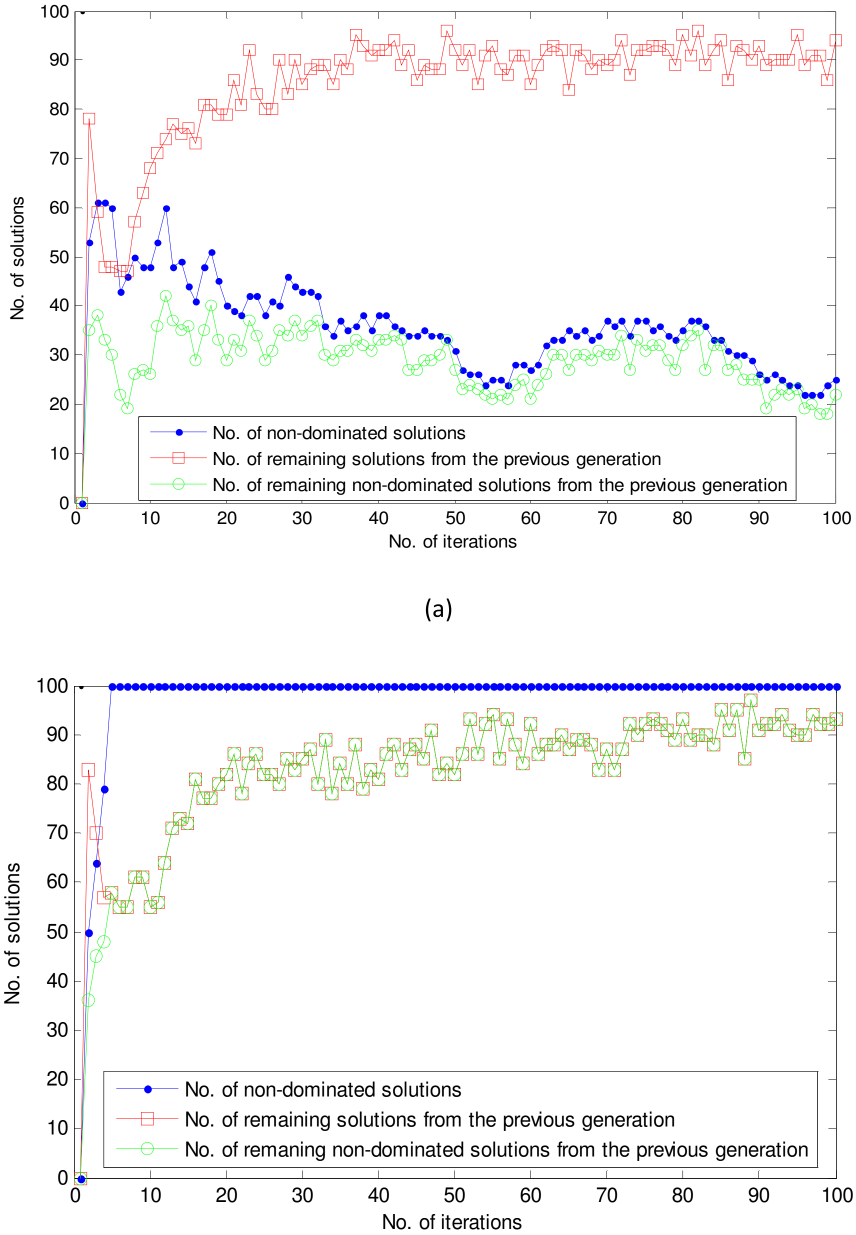
Figure 3.12 – Results using EvABOR-III when β=1: (a) 100 iterations; (b) 200 iterations.

From this analysis it is possible to conclude that when β=1 a high number of solutions remains in the population from one generation to the next and the algorithm has more difficulty in producing new solutions to improve the convergence to obtain solutions belonging to the best class of merit (Figure 3.13). For β=0 that number is lower when compared with β=1, which indicates the algorithm's capability to generate more new solutions increasing the convergence to the region of the search space more in accordance with the DM's preferences (Figure 3.14). To illustrate and compare these situations, the evolution of the population during the evolutionary process is presented for EvABOR-I and III in Figure 3.13 and Figure 3.14.

In conclusion, values of β equal or near 1 do not favor the population diversity and the convergence of the algorithm due to the excessive intra-class elitist pressure. Some more examples are presented in [Oliveira and Antunes (2010a)] and also in a section below where a detailed study of EvABOR-III is done.

Stop Conditions

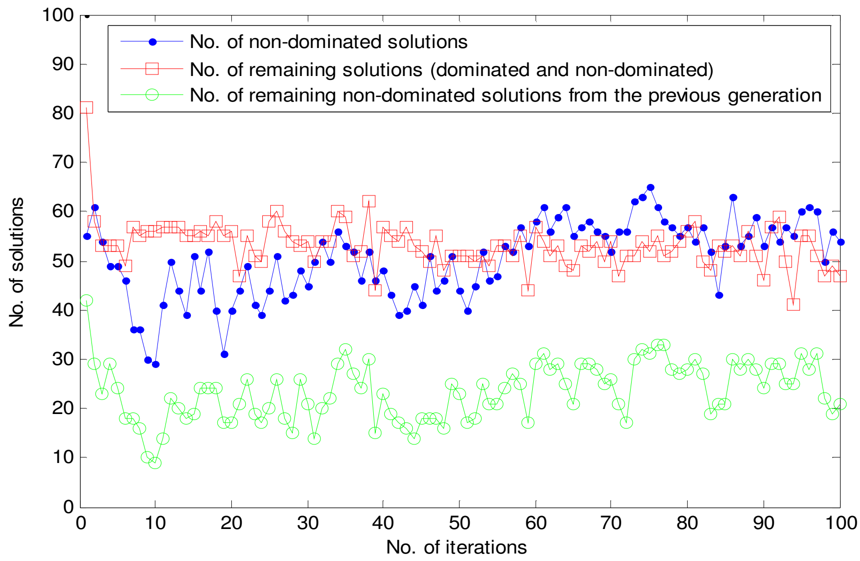
Another interesting aspect unveiled is the fact that the number of solutions remaining in the population increases with the increase of the number of solutions belonging to the best class of merit. This aspect is particularly clear in EvABOR-III when β=0 since the number of solutions remaining in the population is not very high and the evolution of solutions in the classes of merit occurs faster than in the other approaches. The comparison of Figure 3.14 (b) and Figure 3.15 enables confirming this effect. Note the increasing number of solutions that remain in the population after iteration 70, when more solutions belong to the best class of merit.



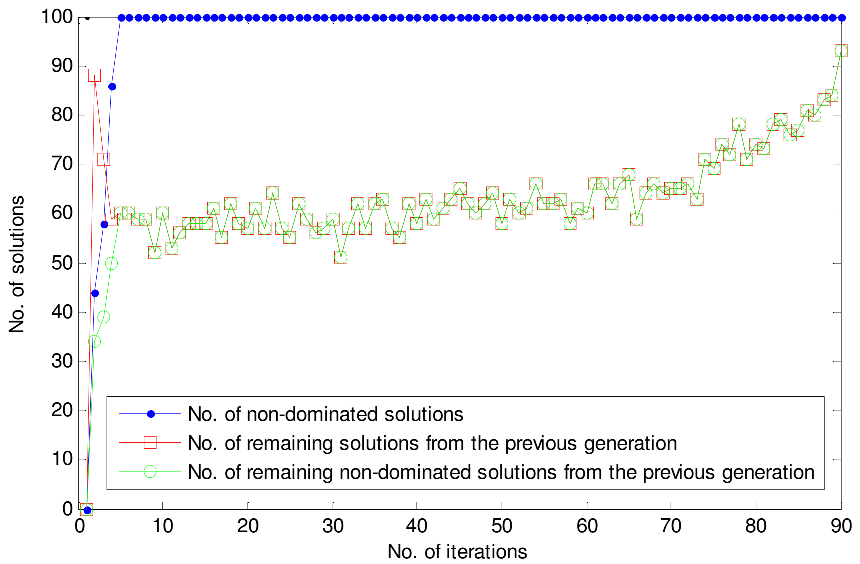
(a)

(b)

Figure 3.13 – Evolution of the population during the evolutionary process with β=1 using: (a) EvABOR-I; (b) EvABOR-III.



(a)



(b)

Figure 3.14 – Evolution of the population during the evolutionary process with β=0 using: (a) EvABOR-I; (b) EvABOR-III.

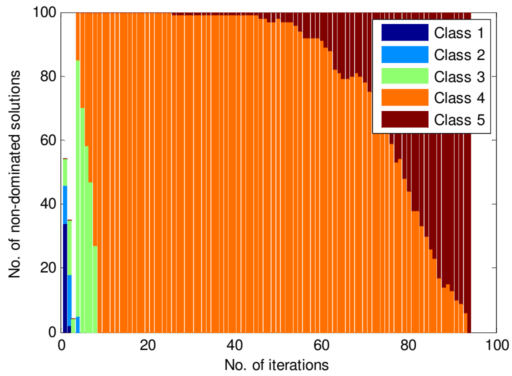


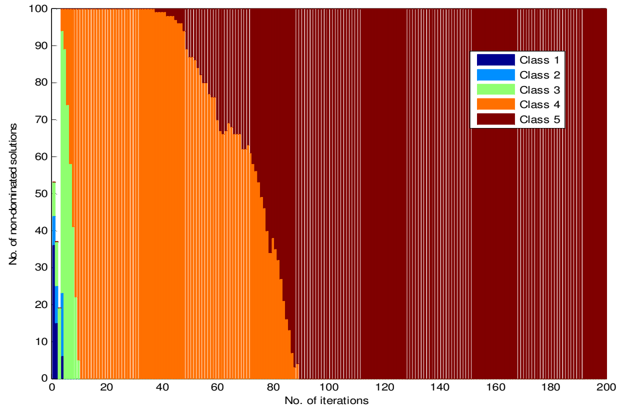
Figure 3.15 – Evolution of the population by classes of merit (β=0) using EvABOR-III.

Expanding the previous comparison to 200 iterations it is possible to conclude that when the entire population belongs to the best class of merit about 90% of the solutions remain in the population in consecutive generations (Figure 3.16). This fact shows some stability in the population. Consequently, the EvABOR algorithms stop when the maximum number of iterations is attained or all solutions in the population belong to the best class of merit.

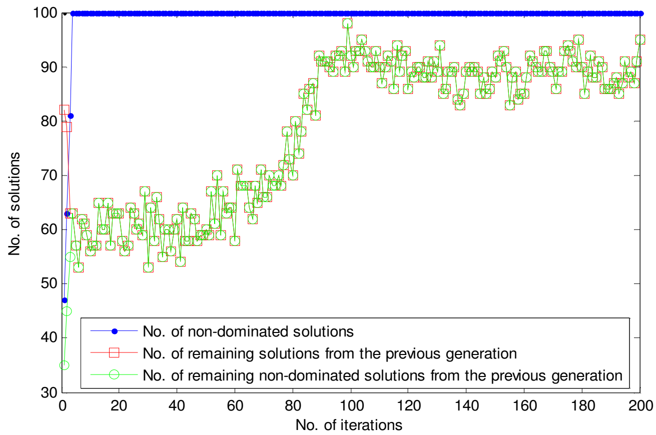
Comparison of EvABOR Approches

The comparison between the different EvABOR approaches has been done with the aim of discovering which one is more effective to find a set of solutions more in accordance with the preferences elicited from a DM (solutions belonging to the best class of merit). To compare the three EvABOR approaches some distinct instances of the reactive power compensation problem were created for testing the algorithms. Each one may be interpreted as representing a different level of exigency imposed by the DM. Three of these instances are presented in this section and used to compare the versions of the EvABOR algorithm. The three different instances will be also useful to introduce the influence of some ELECTRE TRI parameters in the EA.

The dimension of the initial population is 100 for all the results herein presented, but other values have been considered with similar results concerning the comparison of the EvABOR approaches. As referred previously, the algorithms stop when the maximum number of iterations is attained or all solutions in the population belong to the best class of merit.



(a)



(b)

Figure 3.16 - Evolution of the classes (a) and type of solutions (b) during the evolutionary process.

As seen before, the parameter β controls the intra-class elitism of EvABOR approaches. With the exception of values of β near or equal to 1, in which the selective pressure is high, the effect of this parameter in the different EvABOR approaches is similar. In this section, to illustrate the performance of each EvABOR version the results presented have been obtained considering $\beta=0$.

The different approaches have been compared according to the number of solutions in the non-dominated front obtained, the classes of merit in this front, the percentage of non-dominated solutions belonging to each class, the number of iterations necessary to achieve all solutions belonging to the best class of merit, and the percentage of runs in which this latter situation occurs. The results presented are an average of 30 runs of each EvABOR approach.

In the first instance the ELECTRE TRI parameters used are the ones presented in Table 3.2. This instance is not very exigent in terms of technical parameters; however, the values considered herein allow us to analyze and compare the performance of the three EvABOR versions.

Table 3.3 shows the results obtained with this set of parameters for a maximum of 100 and 200 iterations. For both cases EvABOR-III has found the largest non-dominated front. This is justified by the challenging characteristics of this problem, namely related with the very large number of non-dominated solutions in the search space (due to the interplay between the size and location of the capacitors and its influence on cost, losses and maximum voltage deviation objective functions) and the way the EvABOR-III is able to deal with them. As EvABOR-III filters the non-dominated solutions followed by the intervention of the outranking relation, the number of non-dominated solutions tends to increase quickly and, in general, from early iterations all the solutions in the population are non-dominated. This can be verified in Figure 3.17, where the number of non-dominated solutions during the EA is presented as well as the number of solutions (dominated and non-dominated) that are preserved in the next generation (as a way to assess the performance of the algorithm to generate new solutions).

		Resistive Losses (kW)	**Cost (€)**	**Maximum Voltage Deviation (p.u.)**
Reference Profiles		240 260 290 320	38000 60000 85000 100000	0.01 0.03 0.065 0.09
Thresholds	Indifference	5	8000	0.005
	Preference	10	15000	0.01
	Veto	30	40000	0.08
Weights		100/3	100/3	100/3
λ		0.5		

Table 3.2 - ELECTRE TRI parameters considered in the first instance.

	Maximum Number of Iterations = 100			Maximum Number of Iterations = 200		
	EvABOR-I	EvABOR-II	EvABOR-III	EvABOR-I	EvABOR-II	EvABOR-III
Dimension of the non-dominated front	53.4	98	100	59.8	99.8	100
Classes in the non-dominated front	4 and 5	4 and 5	4 and 5	4 and 5	4 and 5	5
% of non-dominated solutions in the best class	2.99%	31.63%	85.7%	0.17%	80.06%	100%
% of runs with the best class only in the front	5%	0%	60%	0%	60%	100%
Number of iterations executed	99.95	100	93.4	200	164	94.1

Table 3.3 – Average values from 30 runs for the first instance.

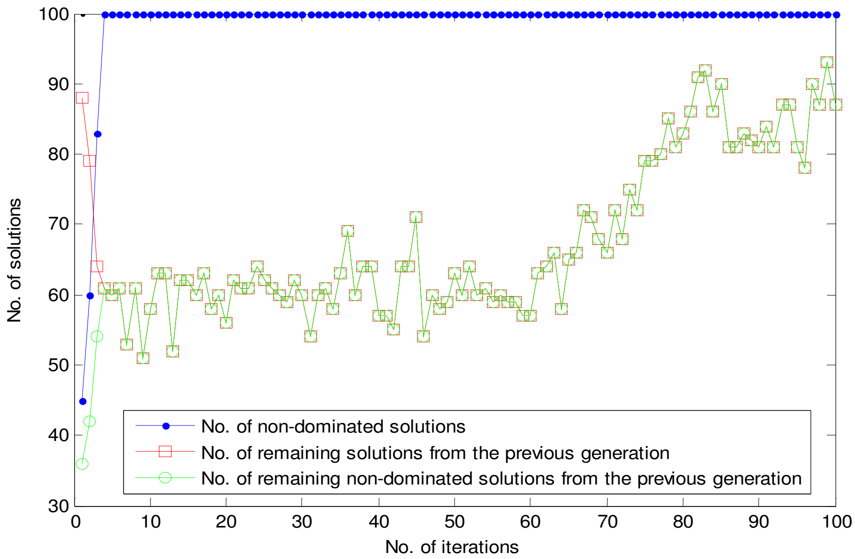


Figure 3.17 – Analysis of the type of solutions during the evolutionary process in EvABOR-III.

A similar analysis has been done to EvABOR-II and EvABOR-III to characterize the population during the evolutionary process. As shown in Figure 3.18 the number of non-dominated solutions presented in each generation of EvABOR-I is much lower than in EvABOR-III (in which from early generations all the solutions in the population are non-dominated) and in EvABOR-II, then the final number of non-dominated solutions is lower than in the other two approaches (Figure 3.17 and Figure 3.19). This fact is due to the priority given to the outranking relation in EvABOR-I and to the non-dominance relation being applied only at the end of the evolutionary process. Consequently, dominated and non-dominated solutions belonging to the best classes of merit pass from one generation to the next one. At the end of the evolutionary process the dimension of the final population is reduced since the dominated solutions are excluded from the population. The corresponding values for

EvABOR-II are presented in Figure 3.19, as well as the dimension of the population at each iteration. Note that, in this case, in some generations some solutions obtained may not be non-dominated. This is due to the fact that the dominance relation to be applied to each class of merit may lead to a situation in which there are dominated solutions obtained from different classes of merit. In EvABOR-II the dimension of the population may vary, but this situation occurs only at the beginning of the evolutionary process when the number of dominated solutions in each class is very large.

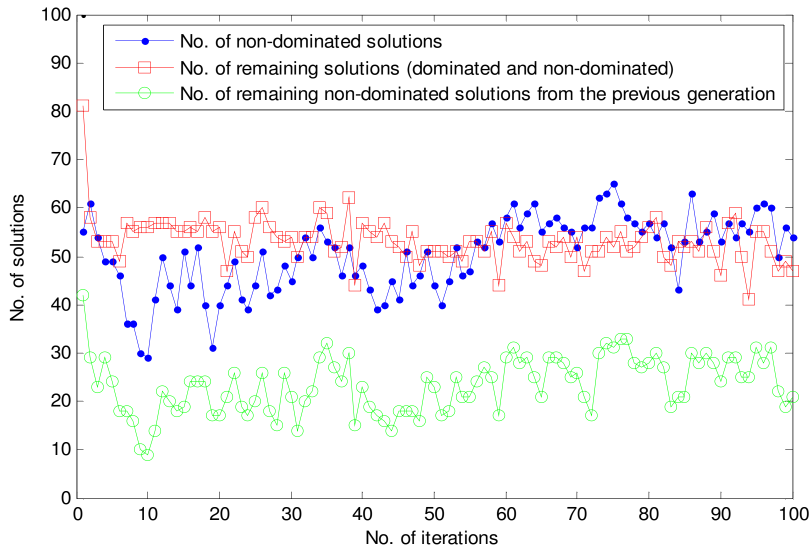


Figure 3.18 – Analysis of the type of solutions during the evolutionary process in EvABOR-I.

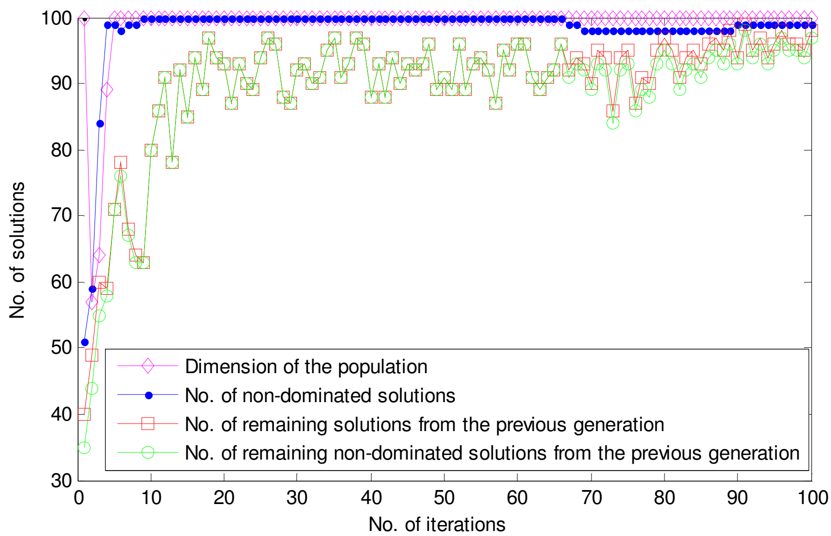
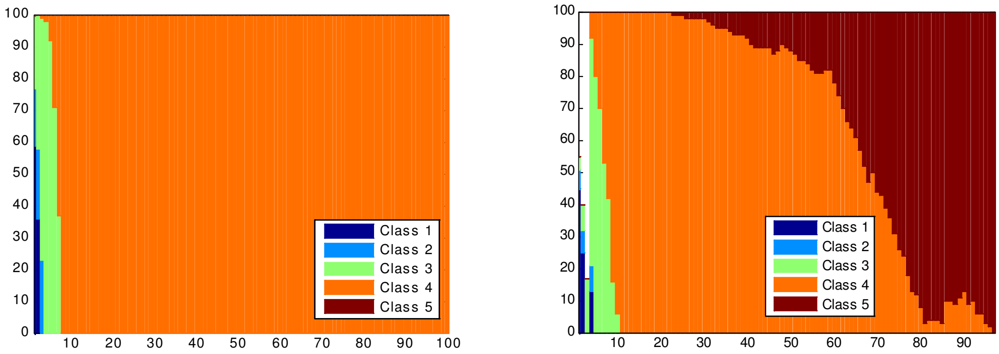


Figure 3.19 – Analysis of the type of solutions during the evolutionary process in EvABOR-II.

Since in EvABOR approaches the aim is to find a set of solutions more in accordance with the preferences elicited from a DM, the number of solutions in the final non-dominated set is not the most important issue. In this context, it is more important to compare the quality of the solutions with respect to the preferences expressed by the DM. This analysis can be performed by comparing the classes of the solutions depicted in the non-dominated front and the number of solutions belonging to each class of merit, which act as a categorical measure of the quality of the solution. As far as this aspect is concerned, EvABOR-III presents the best performance again, since the percentage of solutions belonging to the best class of merit in the non-dominated front is always higher in this approach (Table 3.3). This is further reinforced when the maximum number of iterations is increased to 200. In this case all the solutions in the final non-dominated front belong to the best class of merit (class 5 in this illustrative example) in all runs, while in the other approaches the non-dominated front is composed by solutions in classes 4 and 5. Another important conclusion is that EvABOR-III is the algorithm that obtains the whole non-dominated set belonging to the best class of merit with a smaller number of iterations and more often than the other two approaches. For example, for a maximum of 100 iterations EvABOR-III obtains a non-dominated front composed by solutions in class 5 in 60% of the runs while the other approaches need much more iterations to obtain solutions belonging to this class or even do not achieve this type of solutions.

In order to study deeply the behavior and the effectiveness of the EvABOR-I and EvABOR-II approaches, a maximum number of iterations larger than 100 is preferable. With 100 iterations it seems that EvABOR-II displays a best performance than EvABOR-I. Considering a maximum of 200 iterations this conviction is confirmed: EvABOR-I does not achieve a non-dominated set with all solutions in the best class while EvABOR-II does. In fact, in 60% of the runs EvABOR-II obtains a non-dominated set formed only by solutions in class 5 and 80.06% of the solutions in the non-dominated front belong to this class of merit.

Analyzing the behavior of EvABOR-I in more detail to understand why the progress of the solutions towards the best classes of merit in this approach is slower when compared to the other approaches (Figure 3.20), it is possible to verify that in EvABOR-I the number of dominated solutions in several generations is very high (Figure 3.21). In fact, approximately half of the population consists of dominated solutions, being the reason why the search progress in EvABOR-I is slower.



(a) (b)

Figure 3.20 – Evolution of the classes during: (a) EvABOR-I; (b) EvABOR-III.

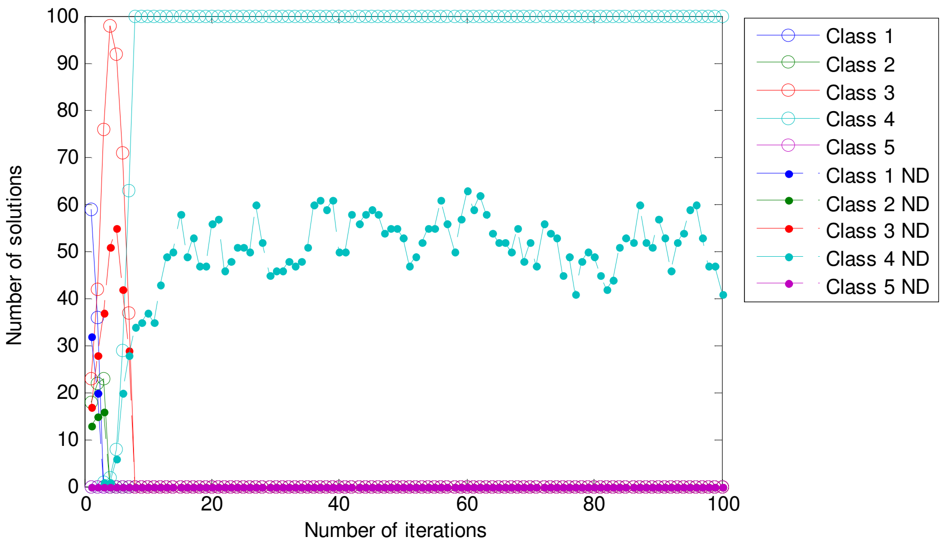
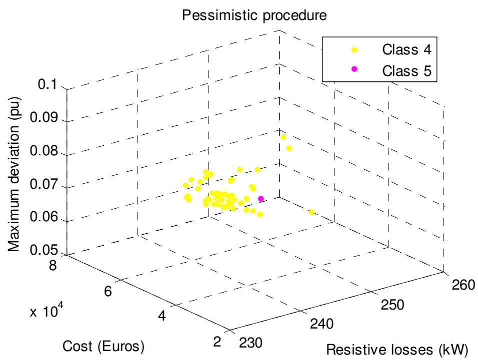


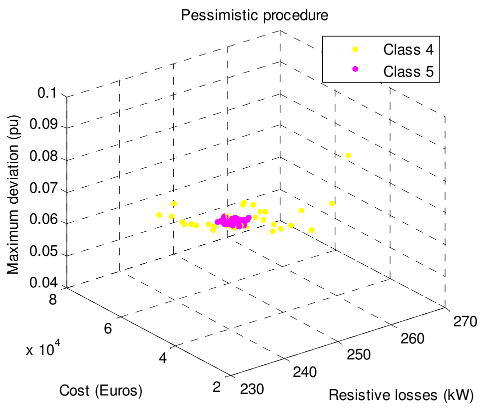
Figure 3.21 – Evolution of the non-dominated solutions in each class during EvABOR-I.

Figure 3.22 displays representative examples of the non-dominated front obtained with each EvABOR approach, where is possible to verify that the number of solutions belonging to the best class of merit is larger for EvABOR-III in comparison with EvABOR-I and EvABOR-II.

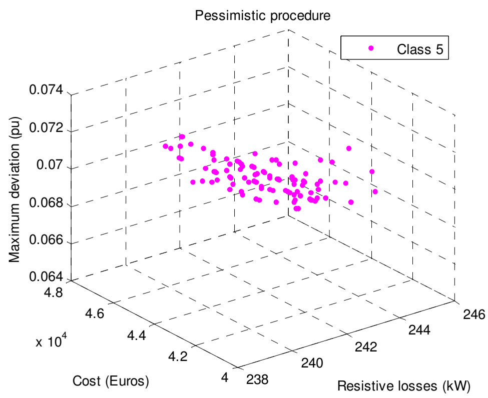
The previous analysis gives strong indications that EvABOR-III is the version of the algorithm with best performance. In order to gather further elements for reinforcing this conclusion other instances have been considered. In Table 3.4 a more exigent set of preference parameters is defined by decreasing the threshold values associated with the cost objective function. The difference between the cost of a solution and the corresponding reference profiles must be lower than in the first instance in order to consider the solution as indifferent or preferred with respect to the reference profiles. Also, the probability to have a solution not classified in the best class of merit increases since the veto threshold is lower.



(a)



(b)



(c)

Figure 3.22 – Example of the final generation found by: (a) EvABOR-I; (b) EvABOR-II; (c) EvABOR-III.

<table>
<tr><td colspan="2"></td><td>Resistive Losses (kW)</td><td>Cost (€)</td><td>Maximum Voltage Deviation (p.u.)</td></tr>
<tr><td colspan="2">Reference Profiles</td><td>240
260
290
320</td><td>38000
60000
85000
100000</td><td>0.01
0.03
0.065
0.09</td></tr>
<tr><td>Thresholds</td><td>Indifference
Preference
Veto</td><td>5
10
30</td><td>5000
10000
30000</td><td>0.005
0.01
0.08</td></tr>
<tr><td colspan="2">Weights</td><td>100/3</td><td>100/3</td><td>100/3</td></tr>
<tr><td colspan="2">λ</td><td colspan="3">0.5</td></tr>
</table>

Table 3.4 – ELECTRE TRI parameters considered in the second instance.

	Maximum Number of Iterations = 100			Maximum Number of Iterations = 200		
	EvABOR-I	EvABOR-II	EvABOR-III	EvABOR-I	EvABOR-II	EvABOR-III
Dimension of the non-dominated front	57.53	99.83	100	55.63	99.30	100
Classes in the non-dominated front	4	4 and 5	4 and 5	4	4 and 5	4 and 5
% of non-dominated solutions in the best class	0%	0.81%	7.43%	0%	3.5%	30.20%
% of runs with the best class only in the front	0%	0%	0%	0%	0%	0%
Number of iterations executed	100	100	100	200	200	200

Table 3.5 – Average values from 30 runs with more exigent thresholds for the cost function.

In Table 3.5 the average values from 30 runs for each EvABOR version, with a maximum of 100 and 200 iterations, are presented considering the preference parameters defined in Table 3.4. These values allow us to reinforce the first conclusions about the performance of the three versions of the EvABOR algorithm. Despite the more exigent conditions imposed to the cost objective function, the EvABOR-III algorithm is again the one with the best performance when compared with the other two versions. The average dimension of the non-dominated front is larger using EvABOR-III, but the most important fact is that the percentage of solutions belonging to the best class of merit is larger when this version of EvABOR is used. This demonstrates its capability to attain solutions according to the preferences elicited even with more exigent sets of parameters.

The comparisons of the values in Table 3.3 and Table 3.5 also confirm that the new set of thresholds is more exigent leading to a larger computational effort (a higher number of iterations) to achieve solutions belonging to the best class of merit. Also the quantity of these solutions is lower than in the first instance. These effects are similar for the three EvABOR approaches. As the values of the cost thresholds are inferior with respect to the first instance, solutions having a higher cost may not be assigned to the best class of merit since the difference between their cost and the respective reference profile may be larger than the current indifference or preference thresholds. Additionally, the veto threshold can also prevent a solution with a bad performance in the cost objective function to be classified in the best class of merit even if the performance in the other objective functions is good. In this second instance this effect is more evident than in the previous one since the value of the veto threshold is lower. Figure 3.23 presents the comparison between the non-dominated set obtained in the first and in the second instances with EvABOR-III for a maximum of 200 iterations. Figure 3.23 (c) is a zoom of Figure 3.23 (b) to make clearer the lower number of solutions in class 5 obtained with the second instance and also the inferior value of the cost function for these solutions. It is important to note the inferior value of the cost function for these solutions. As this set of parameters is more exigent regarding the cost objective function, for a solution be classified in the best class of merit its cost must be lower due to the stronger exigency imposed to this function by the new set of thresholds.

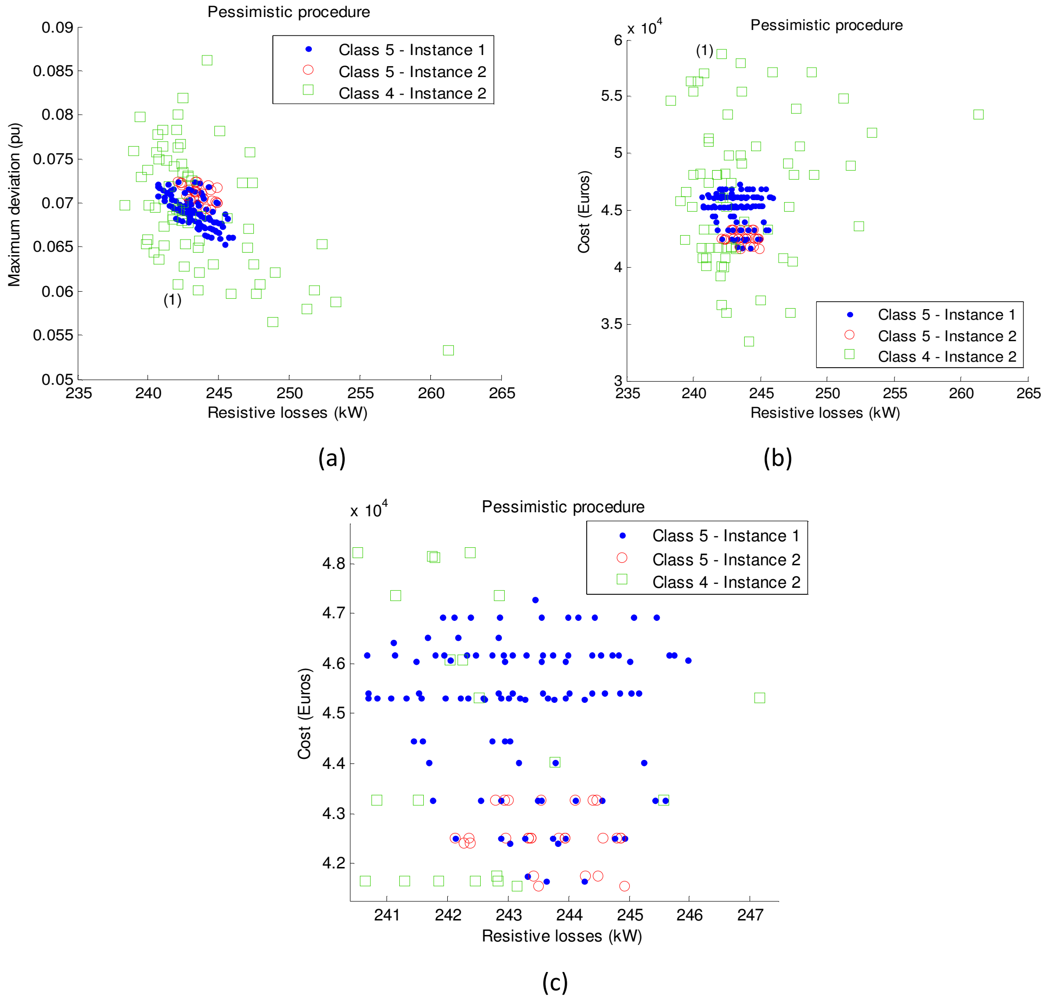


Figure 3.23 – Comparison of non-dominated sets obtained with different values of thresholds using EvABOR-III.

In a third instance the preference parameters related to the maximum voltage deviation are slightly more exigent: the reference profile of this objective function that bounds inferiorly the best class of merit is decreased from 0.01 to 0.006. All the other parameters defined in Table 3.2 remain the same. As we consider the maximum of all voltage deviations in all nodes, it is difficult to find solutions with better values for this objective function. This is the technical reason that leads to a low number of solutions belonging to the best class of merit in all runs (Table 3.6). However, as it is shown in this table, EvABOR-III is still the version of the algorithm with the best performance. EvABOR-II obtains solutions in the best class of merit in just 1 of the 30 runs only when considering a maximum of 200 iterations. Although having some difficulty to obtain solutions in the best class of merit with this preference parameter set, EvABOR-III achieves this aim even for a maximum of 100 iterations.

	Maximum Number of Iterations = 100			Maximum Number of Iterations = 200		
	EvABOR-I	EvABOR-II	EvABOR-III	EvABOR-I	EvABOR-II	EvABOR-III
Dimension of the non-dominated front	57.87	100	100	56.5	99.97	100
Classes in the non-dominated front	4	4	4 and 5	4	4 and 5	4 and 5
% of non-dominated solutions in the best class	0%	0%	1.5%	0%	0.1%	9.93%
% of runs with the best class only in the front	0%	0%	0%	0%	0%	0%
Number of iterations executed	100	100	100	200	200	200

Table 3.6 – Average values from 30 runs with different reference profiles to the maximum voltage deviation.

The comparison of data in Table 3.3 and Table 3.6 confirms that this third instance is more exigent than the first one. With this new set of parameters EvABOR-I does not obtain any solution belonging to the best class of merit contrarily to what happens in the first instance. EvABOR-II needs much more iterations to obtain solutions in the best class and this percentage is smaller than in the previous instance. Despite EvABOR-III presents the best performance in comparison with the other two versions, the number of solutions in class 5 is much fewer than in the first instance.

An illustrative example is shown in Figure 3.24, where the differences between two non-dominated sets obtained with EvABOR-III, considering the first and the third instances, can be compared. Figure 3.24 (c) and (d) are a zoom of the region containing the solutions in the best class of merit, respectively. In these figures only solutions belonging to class 5 are displayed to facilitate analysis. In Figure 3.24 (b) and (d) it is possible to note that the value of the maximum voltage deviation of solutions belonging to the best class of merit is lower with respect to the case of solutions obtained in the third instance. This fact is due to the more exigent reference profile defined for the maximum voltage deviation objective function. Solutions (1) and (2) signaled in Figure 3.24 (c) and (d) have a good performance in losses and cost objective functions, respectively, but they are not classified in the best class of merit in the case of third instance, since the value for the maximum voltage deviation is too high (0.0723 p.u. for solution (1) and 0.07198 p.u. for solution (2)). Despite the good performance of solution (3) in the maximum voltage deviation objective function (0.06582 p.u.), this solution is not assigned to class 5 due to the high cost and losses.

In all instances tested, EvABOR-III is always the version of the algorithm with best performance. Therefore, we may conclude that priority must be given to the non-dominance relation in the sense that this relation must be applied before the outranking relation to filter the non-dominated solutions. Otherwise, the existence of dominated solutions in the population may contribute to delay the progress towards achieving better classes of merit (as in EvABOR-I). Also, there is evidence leading us to affirm that the algorithm guides the search to the region of interest, according to the preferences elicited, more efficiently if the non-dominance relation is applied to the entire population (as in EvABOR-III) and not individually to each class of merit (as happens in EvABOR-II).

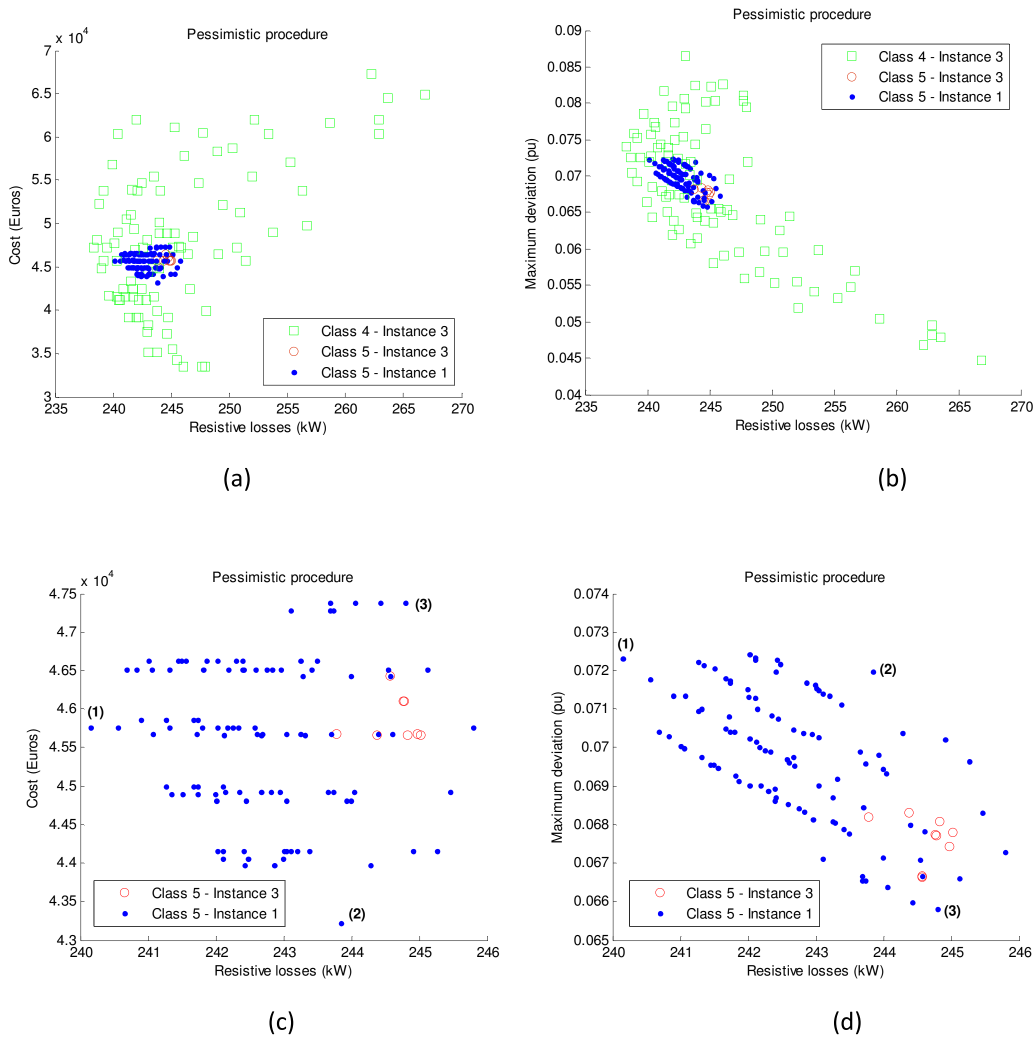


Figure 3.24 – Comparison of two non-dominated sets obtained with EvABOR-III.

THE INFLUENCE OF THE β PARAMETER INTO EVABOR-III

In the previous section a comparison of the three EvABOR approaches has been presented leading us to conclude that EvABOR-III is the most effective approach to find a set of solutions in accordance with the preferences elicited. In face of the superiority of EvABOR-III in the incorporation of preferences into the evolutionary process a more detailed analysis about the influence of the β parameter is done in this section.

Dimension of the population	50			100		
β	0	0.5	1	0	0.5	1
Dimension of the non-dominated front	50	50	50	100	100	100
No. of iterations executed	**99.47**	100	100	**95.63**	99.07	99.47
No. of non-dominated solutions in class 5	**11.33**	2.40	1.23	**85.43**	18.60	44.53
No. of non-dominated solutions in class 4	**38.67**	47.60	48.77	**14.57**	81.40	55.47

Table 3.7– Average results from 30 runs with a maximum of 100 iterations.

Dimension of the population	50			100		
β	0	0.5	1	0	0.5	1
Dimension of the non-dominated front	50	50	50	100	100	100
No. of iterations executed	**136.56**	192.06	196.23	**112.53**	147.86	174.36
No. of non-dominated solutions in class 5	**44.6**	22.03	11.9	**98.67**	98.63	83.7
No. of non-dominated solutions in class 4	**5.4**	27.97	38.1	**1.33**	1.37	16.3

Table 3.8 – Average results from 30 runs with a maximum of 200 iterations.

The results of experiments carried out with the set of preferences defined in Table 3.2 for 100 and 200 iterations, with a population size of 50 and 100 individuals, and different values for the β parameter are summarized in Tables 3.7 to 3.8. The main conclusions are similar independently of the maximum number of iterations. With fewer iterations, EvABOR-III provides better results, obtaining a higher number of solutions in the best classes of merit when β=0. The increase of the β parameter leads to a sort of speciation process in which niches of solutions are created, located near the areas of the search space where the objective functions attain their optimal values, slowing the evolutionary process. In this case the algorithm needs some more time to achieve the region more in accordance with the preferences elicited from the DM. This niche effect tends to disappear with the increase of the number of iterations. Despite β=0 allows to obtain in average better results, the best values obtained for each objective function are achieved considering β=1 (Table 3.9). This result is expected due to the fact that solutions that pass to the next generation are chosen from the classes of merit in an elitist manner.

Dimension of the population		50			100		
β		0	0.5	1	0	0.5	1
Losses	Minimum	240.85	237.03	**236.48**	240.29	240.31	**238.77**
	Maximum	246.41	262.72	264.78	246.29	246.48	253.97
	Average	243.26	247.39	251.23	243.12	243.27	245.02
	Standard deviation	1.41	7.47	10.91	1.46	1.57	4.29
Cost	Minimum	42253	27265	**24738**	41662	41193	**33854**
	Maximum	49688	66893	68705	47510	48004	56652
	Average	46244	49810	51883	45151	45353	46643
	Standard deviation	1746.2	11464.9	17565.4	1391.1	1611.9	5976.0
Maximum Voltage Deviation	Minimum	0.0654	0.0488	**0.0472**	0.0655	0.0650	**0.0580**
	Maximum	0.0737	0.0886	0.0902	0.0728	0.0731	0.0803
	Average	0.0698	0.0696	0.0700	0.0695	0.0693	0.0694
	Standard deviation	0.0020	0.0101	0.0151	0.0018	0.0020	0.0054

Table 3.9 – Average values of the objective functions from 30 runs with a maximum of 200 iterations.

THE INFLUENCE OF ELECTRE TRI PARAMETERS INTO THE SEARCH PROCESS

In a previous section the comparison of EvABOR approaches is presented and the influence of the thresholds and the reference profiles on the search process has been also illustrated. In this section, an analysis of the effect of other ELECTRE TRI parameters on the evolutionary process is presented. Due to the superiority of EvABOR-III in the incorporation of preferences into the evolutionary process, the results presented in the current and in remaining sections are the ones obtained using this version of EvABOR. However, it is important to stress that the effect of these parameters into the other two EvABOR approaches is similar concerning the aspects here addressed (some other examples can be found in [Oliveira and Antunes (2010b)].

To analyze the influence of weights in EvABOR approaches, three different instances have been considered, in which:

- all the objective functions have the same weight (weighting vector=[100/3, 100/3, 100/3]);
- the losses function has the largest weight (weighting vector=[60, 20, 20]);
- the cost function has the largest weight (weighting vector=[20, 60, 20]).

All the other ELECTRE TRI parameters are the ones considered in the first instance defined in Table 3.2. The value of β is 0 because this is the value with which EvABOR-III achieved the best performance (as referred in previous section).

The weights in ELECTRE TRI reflect the true importance of each objective function and are not used as technical parameters for aggregating the objective functions in a scalar function and computing a common value measure as in other approaches. Comparing the several results obtained with the different versions of EvABOR it can be seen that the solutions computed are predominantly located in regions of the search space in agreement with the preference information conveyed by the different sets of weights. Figure 3.25 shows 3D and 2D plots of the non-dominated front obtained from runs using EvABOR-III in the three instances with different set of weights. Comparing the regions highlighted with a green dashed rectangle in Figure 3.25 ((b), (d) and (f)) the influence of the weights is clear. For instance, when resistive losses is the objective function with higher weight there is a predominance of solutions with lower losses (Figure 3.25 (d)) and there are no solutions in the region with higher loss values (marked by a blue rectangle). However, if the cost is considered as the most important objective function, more solutions are obtained in the region of the search space with minimum cost (as may be observed within the ellipse). A similar conclusion can be reached by analyzing the representation of the statistical data in Figure 3.26.

More information from 30 runs (average values) is provided in Table 3.10 where the components of the weighting vector refer to the resistive losses, cost, and voltage deviation objective functions, respectively. It is possible to verify that the minimum values for each objective function are obtained when the corresponding weight is the highest. It is easier for the EvABOR-III algorithm to obtain solutions in the best class of merit when the objective function with a larger weight is the resistive losses (see Table 3.10). In this case the number of iterations to obtain all the solutions in the non-dominated front belonging to the best class of merit is lower than in the other cases.

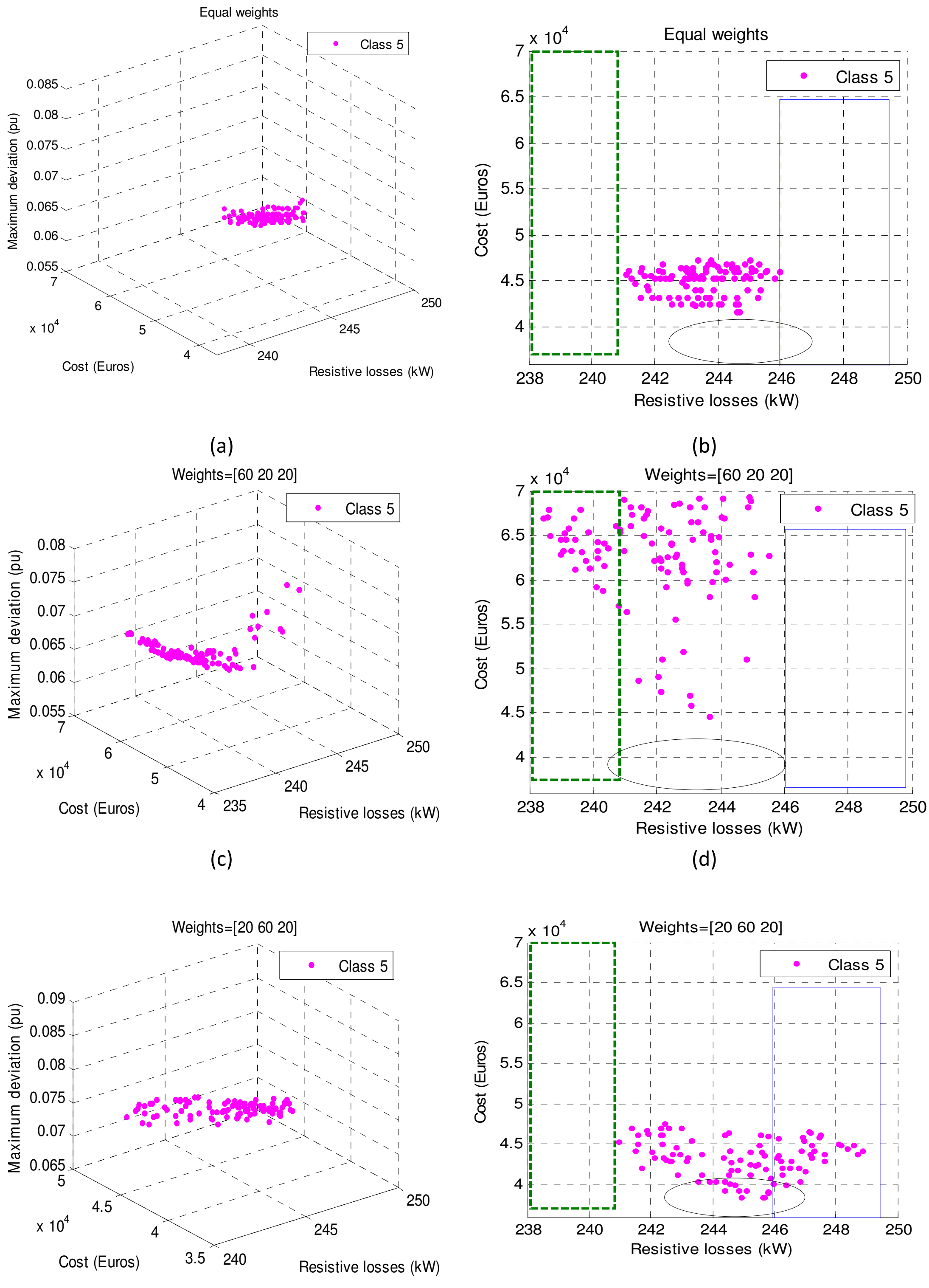
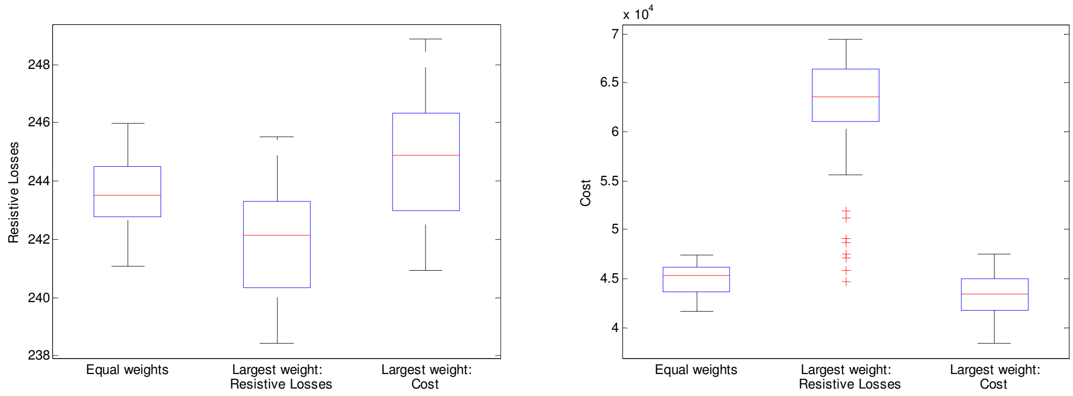


Figure 3.25 – Impacts of different weights in the EvaBOR-III results.



(a) (b)

Figure 3.26 – Statistical data representation for resistive losses (a) and cost (b).

Weight vector=[Losses Cost Deviation]		**[100/3 100/3 100/3]**	**[60 20 20]**	**[20 60 20]**
Maximum no. of iterations		200	200	200
No. of iterations executed		112.53	**21**	30.86
Dimension of the population		100	100	100
Dimension of the non-dominated front		100	100	100
No. of solutions in class 5		98.67	100	100
No. of solutions in class 4		1.33	0	0
No. of solutions in class 3		0	0	0
No. of solutions in class 2		0	0	0
No. of solutions in class 1		0	0	0
Losses	Minimum	240.29	**239.31**	239.89
	Maximum	246.29	245.50	253.90
	Average	243.12	242.60	243.87
	Standard deviation	1.46	1.56	2.57
Cost	Minimum	41662.37	41611.5	**37316.5**
	Maximum	47509.63	69232.03	47829.6
	Average	45151.14	58896.6	43322.5
	Standard deviation	1391.1	7297.6	2654.1
Maximum Voltage Deviation	Minimum	0.06547	0.05848	0.06566
	Maximum	0.07281	0.08027	0.08097
	Average	0.06953	0.06586	0.07436
	Standard deviation	0.0018	0.0054	0.0036

Table 3.10 – Average values from 30 runs and different weight values using EvABOR-III.

The veto threshold is another important parameter in ELECTRE TRI and consequently in the evolutionary process of EvABOR approaches. This parameter precludes a solution having a bad performance according to the preferences elicited in a given objective function from being classified in the best class of merit even if its performance is very good in the other objective functions. The EvABOR approaches have been tested with distinct veto thresholds for each objective function. For illustrative purposes, results obtained using EvABOR-III considering different values for the veto threshold (0.08, 0.075 and 0.04) for the maximum voltage deviation objective are presented below. All other parameters are the same defined in Table 3.2. The three values of the veto threshold have distinct impacts on the results. While the less demanding value practically does not influence the sorting of solutions into categories, a slightly more demanding value (0.075) already prevents most solutions from being classified in the best class of merit. A strongly more demanding value (0.04) leads to a situation where no solution is classified in the best class of merit. Figure 3.27 shows comparative examples for different veto thresholds, in which two similar solutions are classified in different classes of merit (class 5 in (a) and class 4 in (b)) due to the different values of the veto threshold. When the veto threshold is 0.08 the solution is classified in class 5. However, when this threshold is a little more demanding (0.075) a similar solution is classified in class 4 (Figure 3.27 (c)). For a more demanding value for this threshold (0.04) all solutions in the non-dominated front belong to class 4. Table 3.11 summarizes some results from 30 runs obtained using EvABOR-III with the referred different veto thresholds for the maximum voltage deviation objective function. When the veto threshold is 0.08 a non-dominated set with almost all solutions belong to class 5 is obtained.

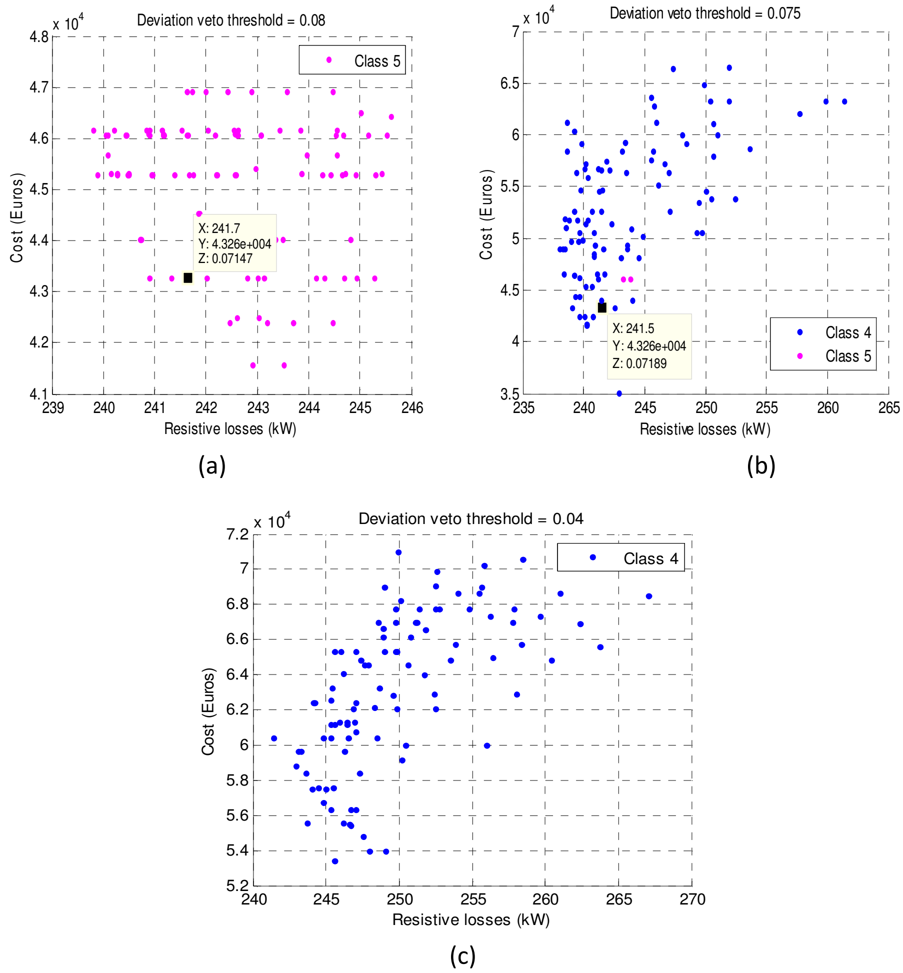


Figure 3.27 – Illustrative example of the veto thresholds influence using EvABOR-III.

Deviation veto threshold	0.08	0.075	0.04
Maximum no. of iterations	200	200	200
No. of iterations executed	**112.53**	200	200
Dimension of the population	100	100	100
Dimension of the non-dominated front	100	100	100
No. of non-dominated solutions in class 5	**98.67**	14.3	**0**
No. of non-dominated solutions in class 4	**1.33**	85.7	**100**

Table 3.11 – Average values from 30 runs with different deviation veto thresholds using EvABOR-III.

The role of the indifference and the preference thresholds is translated into a higher or a lower acceptance of the difference between the performances of a solution with the respective reference profiles. To illustrate this influence in EvABOR approaches, new values for the indifference and the preference thresholds are considered for the cost objective function (6000€ and 10000€, respectively). All the other parameters are the same of the first instance defined in Table 3.2.

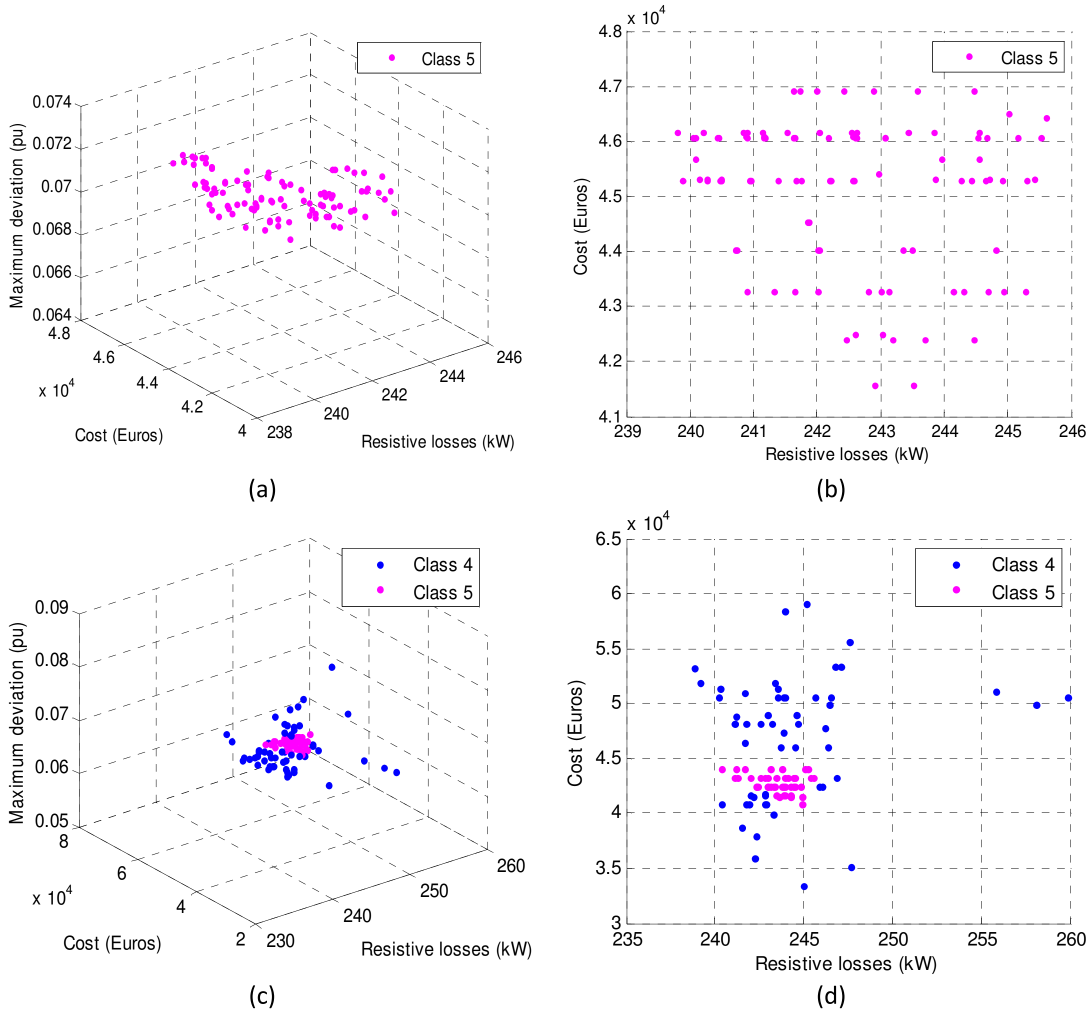


Figure 3.28 – Illustrative example of the indifference and the preference thresholds influence in EvABOR approaches: (a), (b): 1st instance; (c), (d): current instance.

In this instance, for a solution to be classified in the best class of merit it is necessary that the value of the cost objective function is nearer to the inferior reference profile for this objective function (38000€) than in the initial scenario. Consequently, the number of solutions in the best class of merit tends to decrease but these solutions have a lower cost. An example, obtained using EvABOR-III, is presented in Figure 3.28, where in the initial instance all non-dominated solutions obtained belong to class 5 (Figure 3.28 (a) and (b)) and in the current instance fewer solutions are obtained in this class (Figure 3.28 (c) and (d)). Table 3.12 presents the average values from 30 runs obtained with EvABOR-III. In the new instance just about 47% of the solutions are classified in the best class of merit while in the initial instance about 98% of the solutions are in this class and are obtained without reaching the maximum number of iterations.

Indifference and Preference Thresholds	**Initial Instance** q_{cost} **=8000€** p_{cost} **=15000€**	**New Instance** q_{cost} **=6000€** p_{cost} **=10000€**
Maximum no. of iterations	200	200
Average no. of iterations executed	**112.53**	200
Dimension of the population	100	100
Dimension of the non-dominated front	100	100
Average no. of non-dominated solutions in class 5	**98.67**	**47.27**
Average no. of non-dominated solutions in class 4	**1.33**	**52.73**

Table 3.12 – Average values from 30 runs with different indifference and preference thresholds obtained with EvABOR-III.

The level of exigency of the classification (majority requirement) is defined by the cutting-level λ. The minimum value of λ has been considered in the previous instances. When the cutting-level λ is increased to 0.7, thus increasing the level of exigency, solutions found by EvABOR approaches can be classified in different classes of merit. In this case a majority requiring two objective functions is necessary. An example to illustrate this fact is presented using EvABOR-III. All the parameters are the ones in Table 3.2 with the exception of the cutting-level that is 0.7 in the current instance. The increase of the exigency induced by the new value of the cutting-level leads that, for example, solution (240.4; 46440; 0.07211) is not classified in class 5 (Figure 3.29 (c)) but a similar solution (240; 46060; 0.072) is classified in class 5 when λ is equal to 0.5 (Figure 3.29 (a)).

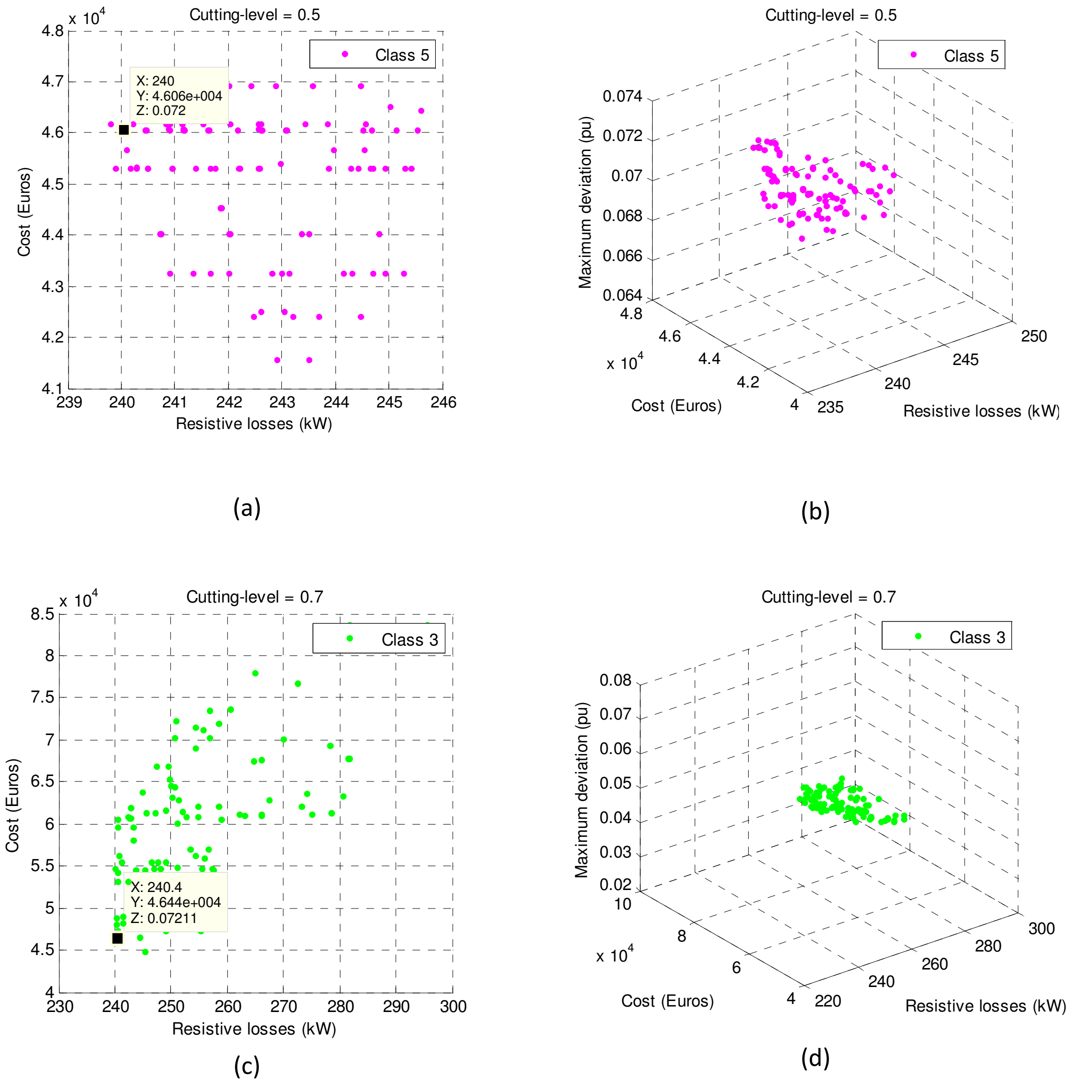


Figure 3.29 – Illustrative example of the cutting-level influence in EvABOR approaches.

Table 3.13 shows the average results for 30 runs with population sizes of 50 and 100 and a maximum number of iterations equal to 200 using EvABOR-III. It is possible to see that when the cutting-level λ=0.7 the algorithm was not able to compute solutions belonging to the best class of merit even using all the maximum number of iterations allowed, while when λ=0.5 it was able to identify a significant number of solutions belonging to the best class even without using all the iterations. This is illustrative of the effect of the cutting-level parameter in the evolutionary process and also in the classification of solutions.

Cutting-level value		**0.5**	**0.7**	**0.5**	**0.7**
Dimension of the population		50		100	
Maximum no. of iterations		200	200	200	200
No. of iterations executed		136.57	200	112.53	200
Dimension of the non-dominated front		50	50	100	100
No. of non-dominated solutions in class 5		**44.6**	**0**	**98.67**	**0**
No. of non-dominated solutions in class 4		**5.4**	**0**	**1.33**	**0**
No. of non-dominated solutions in class 3		0	**50**	0	**100**
No. of non-dominated solutions in class 2		0	0	0	0
No. of non-dominated solutions in class 1		0	0	0	0
Losses	Minimum	240.85	238.13	240.29	237.87
	Maximum	246.41	264.27	246.29	270.50
	Average	243.26	244.30	243.12	244.97
	Standard deviation	1.41	5.90	1.46	6.40
Cost	Minimum	42253.5	47798.7	41662.4	44610.7
	Maximum	49688.4	75556.7	47509.6	76674.2
	Average	46243.6	59923.4	45151.1	58437.3
	Standard deviation	1746.2	6365.6	1391.1	6702.2
Maximum Voltage Deviation	Minimum	0.06538	0.04563	0.06548	0.04148
	Maximum	0.07369	0.07289	0.07281	0.07381
	Average	0.06985	0.06207	0.06953	0.06113
	Standard deviation	0.00202	0.00647	0.00180	0.00700

Table 3.13 – Average values from 30 runs and different cutting-level values obtained using EvABOR-III.

COMPARISON BETWEEN EVABOR APPROACHES AND AN EA WITH INCORPORATION OF PREFERENCES *A POSTERIORI*

To analyze how the EvABOR algorithm guides the search to the region of the space more in accordance with the preferences elicited from the DM, it is interesting to compare the non-dominated fronts obtained previously with the one achieved with an EA without incorporation of preferences. The EA parameters are the same in all approaches.

As this case study is a real-world problem, the true Pareto-optimal front is not known. However, an approximation to this set has been obtained with an EA similar to NSGA-II [Deb et al. (2002)]. The binary tournament selection is used to select parents in the crossover operator. If the individuals belong to different fronts, the one of the best front is chosen to be a parent. If the individuals belong to the same front, the solution to become a parent is picked randomly. The mutation operator is applied also according to the mutation probability but the choice of the operator from the five forms described in Section 3.3.2 is random, i.e., the choice of the mutation operator is not guided. In the selection the population is sorted using the non-dominance relation. The new population is filled with the solutions of the first non-dominated fronts as in NSGA-II. However, if the last front has more solutions than needed, the solutions are chosen randomly, i.e., no crowding strategy is used to choose the remaining individuals. Despite the diversified and well-spread non-dominated set in the objective space obtained with this EA, its efficiency to attain solutions in the best class of merit is rather poor due to the large number of non-dominated solutions in the population. Figure 3.30 presents a comparison between a non-dominated front obtained with the EA without preference incorporation and the non-dominated sets obtained with EvABOR-III for the different instances considered before for a maximum of 100 iterations. This comparison is done with EvABOR-III since this is the version that proved to have the best performance.

The solutions obtained with EvABOR-III are mostly concentrated in a region of the search space with the cost and losses objective functions values lower than the solutions obtained with the EA without incorporation of preferences. In this case there is an intensive exploration of regions that do not match with the preferences elicited despite the computational effort. This assertion can be confirmed applying the ELECTRE TRI method *a posteriori* to evaluate the quality of the solutions obtained by the EA regarding the preferences. When this analysis is done with the preferences considered in the previous instances, none or only a very small number of solutions is assigned to the best class of merit. The non-dominated set obtained with the EA without preference incorporation displayed in Figure 3.30 is also shown in Figure 3.31 after the sorting of the solutions considering the set of preferences defined in Table 3.2 (the first instance considered). Note that no solution belongs to the best class of merit, despite the non-exigent preferences elicited in this instance, while the non-dominated front obtained with EvABOR-III is, in general, only composed by solutions in that class (Table 3.3).

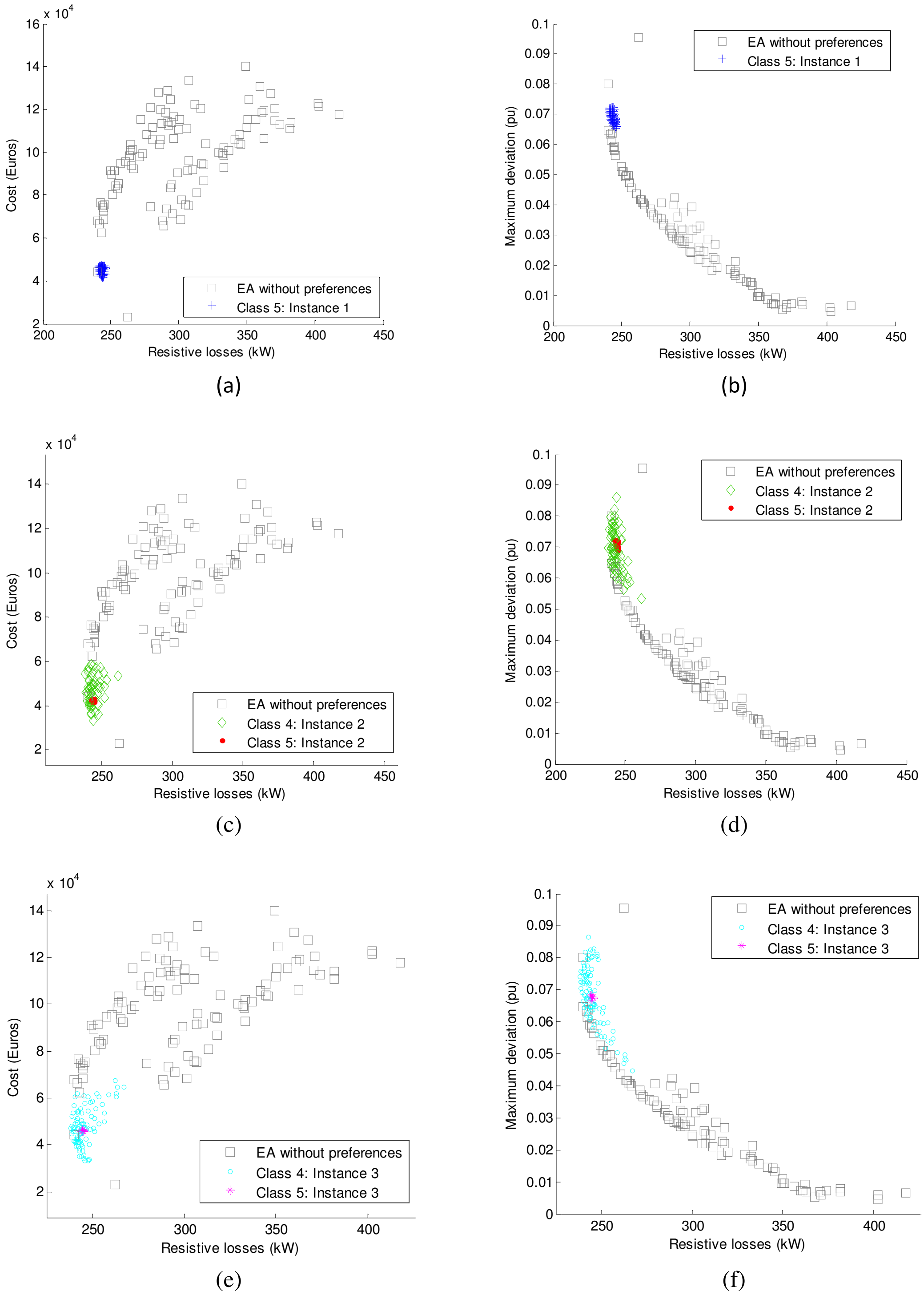


Figure 3.30 – Comparison of non-dominated fronts obtained with an EA without incorporation of preferences and EvABOR-III. (a), (b): Instance 1; (c), (d): Instance 2; (e), (f): Instance 3.

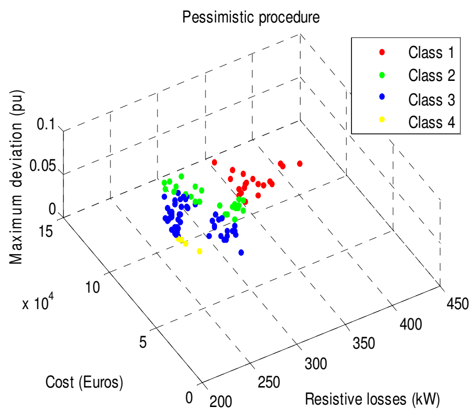


Figure 3.31 – Sorting of the non-dominated front obtained with an EA without incorporation of preferences.

To understand the behavior of the EA without incorporation of preferences concerning solutions belonging to the best class of merit, the evolution of these solutions during the evolutionary process has been analyzed. The number of solutions in the best class of merit (class 5) has been evaluated and compared before and after the non-dominance procedure (Figure 3.32). As it is possible to observe, despite some solutions in the best class of merit are obtained in EA, the selection of individuals to the next generation using only the non-dominance relation does not guarantee the continuity of these solutions in the population. The large number of non-dominated solutions, due to the contradictory nature of objective functions and the combinatorial characteristics of the problem, leads the sorting of the population by levels of non-dominance to be insufficient, because solutions belonging to the best class may be lost during the selection procedure using the non-dominance relation only. Note that solutions in the best class of merit can be non-dominated with respect to solutions belonging to other classes. For this reason, it is necessary to include in the algorithm some mechanism guaranteeing that these solutions go to the next generation, otherwise they can be lost.

Finally, to exemplify a possible practical implementation for the reactive power compensation problem described in Section 3.3., a set of solutions belonging to the best class of merit (class 5) is presented. These solutions have been obtained with EvABOR-III in one of the runs performed with the third instance defined previously. Due to the stringent preference parameters established, the number of solutions is small, which somehow facilitates the choice of a final solution for practical implementation. However, if the number of solutions in the best class of merit is high it is possible to redefine a more exigent set of preference parameters (as explained before), or even to apply a method devoted to the choice, or the ranking, problem to deal with solutions belonging to the best class of merit (therefore in accordance with the preferences elicited).

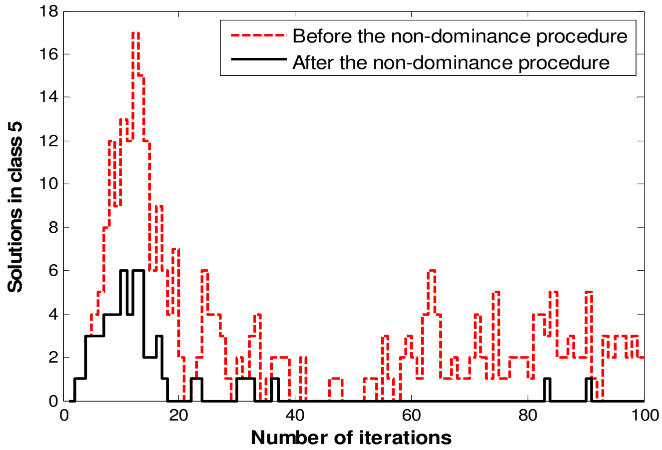


Figure 3.32 – Number of solutions in class 5 during the evolutionary process.

The solutions and the corresponding objective functions values are presented in Table 3.14 and Table 3.15. Let us suppose that a DM would select solution 2 for practical implementation, as a well-balanced solution regarding the values obtained for the three objective functions of economical, technical and quality of service nature. Figure 3.33 displays the location and the types of capacitors to be installed in the network.

Solution	Solution encoding
1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 5 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0
2	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 5 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0
3	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 7 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0
4	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 7 0 0 4 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0

Table 3.14 – Physical characterization of solutions in class 5 (0 = no capacitor installed; values 1 to 8 = type of capacitor installed in that node 1-94).

Solution	Cost	Resistive losses	Maximum voltage deviation
1	45304	244.6	0.068094
2	45304	244.42	0.068413
3	45407	244.9	0.067853
4	45676	244.47	0.068161

Table 3.15 – Objective function values for solutions in Figure 3.14.

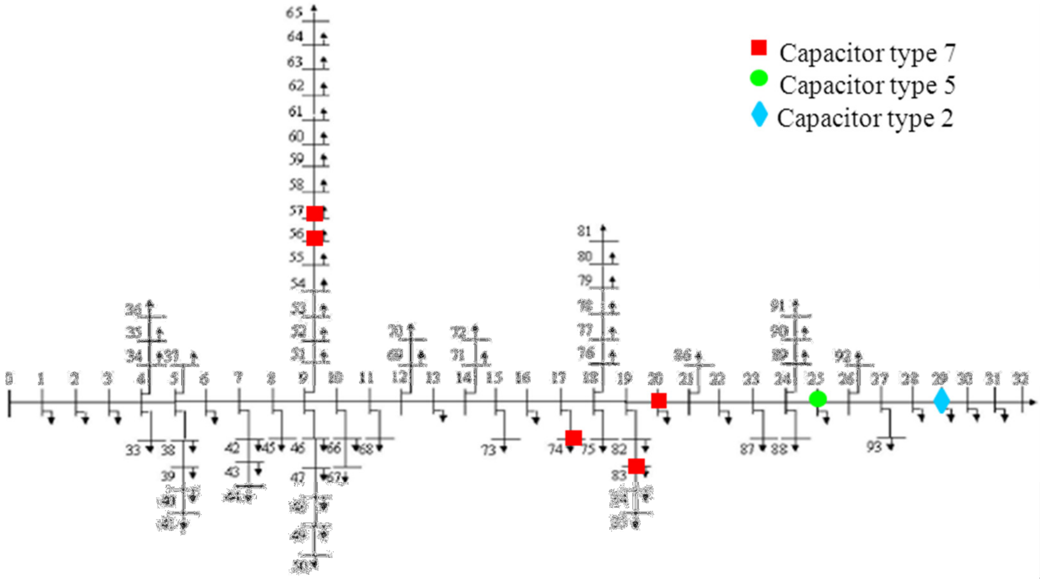


Figure 3.33 – The compensated network.

Other Results

As referred to in Section 3.2.1, the parents for the crossover operator are selected using a tournament technique. In this procedure the parents that are more in accordance with the preferences elicited are chosen to generate the offspring. Figure 3.34 presents the percentage of parents chosen according to one of the following conditions: belong to the best class or present better performance in more objective functions. These values are the average of 30 runs with 100 individuals and a maximum of 100 iterations, and the crossover and mutation probabilities are 1 and 0.2, respectively. The ELECTRE TRI parameters are the ones considered in Table 3.2.

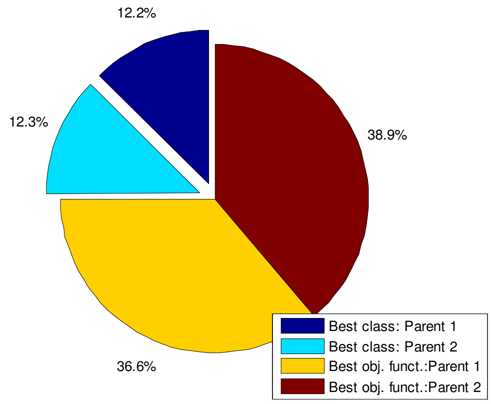
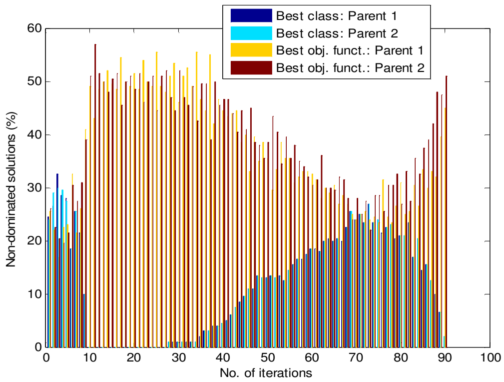


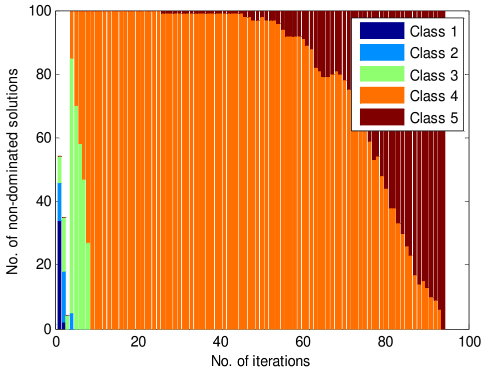
Figure 3.34 – Selection of the parents for tournament.

The analysis of these percentages during the evolutionary process (Figure 3.35 (a)) compared with the number of classes existing in a population at a certain instant (Figure 3.35 (b)) enables to conclude that when the number of classes decreases the parents tend to be chosen according to their performance in the objective functions.

However, the previous percentages strongly depend on several factors, such as the maximum number of classes (a higher number of classes increases the probability to have more than one class in a particular generation), the number of classes in each generation and the maximum number of iterations.



(a)



(b)

Figure 3.35 – (a) Selection of the parents for tournament during the evolutionary process; (b) Evolution of the population by classes of merit using EvABOR-III.

The computational effort associated with the main operations within EvABOR-III is analyzed from a set of 30 runs performed considering the parameters presented in Table 3.2. The average percentage of the CPU time devoted to each task in each generation is evaluated. As it was expected the larger percentage of time (56.51%) is expended in the power flow algorithm, which computes the (active and reactive) power and the voltage at each network node resulting from a given compensation scheme (that is, a solution representing a given location and sizing of the capacitors). The average percentage of time devoted to the outranking relation is 8.24% of the total time, considering all the steps in which preferences are used (crossover, mutation and selection operators). As the non-dominance relation is only performed in the selection operator it consumes 0.91% of the total time. The crossover, mutation and the selection operators spend 3.95%, 1.29% and 0.27%, respectively. The remaining time is consumed in other auxiliary tasks.

The average time expended in a run is 25.8 seconds. The experiments are done in a Core 2 Duo, 2.80 GHz with 4GBof RAM and the algorithm has been developed using Matlab® R2008b software.

CHAPTER 4

HYBRID ALGORITHMS WITH INCORPORATION OF PREFERENCES[4]

In the previous chapter the incorporation of preferences into an EA has been presented. This approach have revealed good results having in mind the search of a set of non-dominated solutions according to the preferences elicited from the DM. However, whenever the expressed preferences are more exigent, the computational effort of EvABOR to obtain all population belonging to the best class of merit may increase significantly, even after obtaining a solution in the best class of merit. This point is especially relevant in MOOPs in which the search space is large due to the high number of objective functions and/or the combinatorial characteristics of the problem. Moreover, if the evaluation of solutions is computationally heavy, the execution time may become prohibitive if it is necessary to have a solution in real time. Consequently, to improve the exploitation in regions more in accordance with the expressed preferences a local search phase has been included in the EvABOR algorithm. The Simulated Annealing (SA) algorithm has been used in this phase and incorporation of preferences is also considered in this algorithm.

In the previous EvABOR algorithms, the initial solutions are generated randomly. However, the quality of these solutions may influence the performance of the algorithm. To assess the

[4] This chapter is partially based on [Antunes CH, Lima P, Oliveira E, Pires DF. A Multi-Objective Simulated Annealing Approach to Reactive Power Compensation. Engineering Optimization, 2011; 43(10), 1063-1077], [Antunes CH, Oliveira E, Lima P. A Multi-Objective GRASP Procedure for Reactive Power Compensation Planning. Optimization and Engineering (submitted)], [Gomes A, Henggeler Antunes C, Oliveira E. Direct load control in the perspective of an electricity retailer – a multi-objective evolutionary approach. In: Gaspar Cunha A, Takahashi R, Schaefer G, Costa L (Eds.), Soft Computing in Industrial Application, Advances in Intelligent and Soft Computing, 96, Springer, 2011. p. 13-26] and [Gomes A, Henggeler Antunes C, Martinho J, Oliveira E. Otimização multiobjetivo com algoritmos evolutivos - uma aplicação no sector elétrico. In: XVI Latin-Ibero-American Conference on Operations Research and XLIV Brazilian Symposium on Operations Research, 2012].

influence of initial solutions on the overall results, a hybrid algorithm has been developed combining a construction phase, as in the Greedy Randomized Adaptive Search Procedure (GRASP), and a local search phase using the SA with incorporation of preferences. In the construction phase solutions are generated based on the knowledge about the problem at hand. To compare these two algorithms a direct load control problem with seven objective functions has been used.

4.1. THE GRASP ALGORITHM

The GRASP (Greedy Randomized Adaptive Search Procedure) meta-heuristic is a multi-start process consisting of two main phases: in the construction phase a feasible solution is built using a greedy randomized algorithm and its neighborhood is then explored in the local search phase until a local optimum is found [Feo and Resende (1995), Resende and Ribeiro (2003)]. These two phases are repeated at each iteration until a stop condition is reached.

The aim of the construction phase is to generate a diverse set of good-quality starting solutions by introducing a controlled randomization component into a greedy algorithm. A solution is composed of elements that are progressively integrated. At each iteration of the construction phase, a set of candidate elements is defined consisting of all elements that can be incorporated into the solution under construction while maintaining its feasibility. In some problems, repairing procedures may be required to restore feasibility of the constructed solution, provided that the associated computational burden is not too high. In general, the selection of the next element to be integrated into the solution results from the evaluation of all candidate elements according to a greedy evaluation function. This greedy function assesses the marginal cost (degradation of the objective function value) resulting from the integration of that element into the solution being constructed. The so-called restricted candidate list (RCL) is defined greedily, by including the elements with least marginal cost, i.e. the best elements. The probabilistic feature relies on the (controlled) random selection of the element to be integrated into the solution under construction from the RCL. The integration of the selected element into the solution leads to its removal from the RCL and re-computing the marginal costs of the elements still in the RCL. Therefore, this construction phase can be classified as a greedy randomized procedure. The construction phase, balancing the greedy and the randomized aspects, is responsible for providing good quality starting solutions to the local search phase.

The local search phase is aimed at improving the solution constructed by exploring its neighborhood in a single-objective optimization problem. Whenever a better solution is found in the neighborhood of the current solution then it replaces the latter. As in any local search scheme, a careful definition must be made of the neighborhood structure and search technique, for the sake of both efficiency (namely computational effort) and effectiveness (quality of the solution produced).

Pseudo-codes of a generic GRASP and the respective construction and local search phase to a single-objective optimization problem can be found in [Feo and Resende (1995)].

Despite the large number of works using GRASP in single-objective optimization problems and its application in several areas (see [Festa and Resende (2009a, 2009b, 2011)] for a vast bibliography), GRASP has not been used often to deal with MOOPs, unlike other metaheuristics such as SA or GA. Some works dealing with multi-objective GRASP approaches have been carried out in recent years and just a few uses the non-dominance concept in the optimization process. The GRASP algorithm presented by Vianna and Arroyo (2004) is based on the optimization of a weighted linear utility function for a multi-objective knapsack problem. In Higgins et al. (2008) a weighted-sum objective function is used to compute non-dominated solutions for an environmental planning problem involving biodiversity, water run-off and carbon sequestration objectives. The weight of each objective function is randomly generated. In Arroyo et al. (2008) a GRASP algorithm based on the optimization of different weighted utility functions is presented for the multi-objective minimum spanning tree problem. In the construction phase the Kruskal algorithm is used and in the local search phase a “drop-and-add” neighborhood transformation is applied. Li and Landa-Silva (2009) applied GRASP to a multi-objective quadratic assignment problem. The algorithm uses an elitist-based greedy randomized construction, cooperation between solutions and an adaptive weighting technique to guide the search. Chica et al. (2010) apply the MORGA (multi-objective random greedy search algorithm) procedure to solve a time and space assembly line balancing problem. The MORGA diversification generation mechanism is inspired by the construction phase of GRASP and in this approach the selection of the task at each point in time is guided by a stochastic greedy function. The algorithm proposed by Mauttone and Urquhart (2009), to obtain an optimal design of routes and frequencies in urban public transit systems (a transit network design problem), generates a set of routes in the construction phase, changing the maximum route duration between successive GRASP iterations to obtain different trade-off levels. In the local search phase, this is achieved using a random vector of weights. At the end of each GRASP iteration the non-dominated solutions are filtered.

Only some recent works using the concept of non-dominance have been proposed to extend GRASP to the multi-objective optimization context. Reynolds and Iglesia (2009) and Reynolds et al. (2009) use this concept in their multi-objective GRASP algorithm for partial classification and rule selection, respectively. In Arroyo and Pereira (2011) the multi-objective GRASP (named M-GRASP) algorithm blends the concepts of scalarizing functions and non-dominance. However, the search of non-dominated solutions uses a weighted linear utility function with the weighting vectors uniformly chosen at each iteration of the heuristic with the purpose of covering the entire Pareto front. The local search phase is based on the Variable Neighborhood Search heuristic and combines weighted utility functions and Pareto dominance approaches. In Antunes et al. (submitted) a multi-objective GRASP using the non-dominance concept is presented for a reactive power compensation problem. In this approach a construction method based on several RCLs is proposed to obtain solutions with distinct characteristics to cover different regions of the space and, consequently, to improve solution diversity. Martí et al. (2011) present some variants of an algorithm combining GRASP with path-relinking to obtain non-dominated solutions for two bi-objective optimization problems (a path dissimilarity problem and a bi-orienteering problem). In these approaches, different constructive methods for both problems are tested considering the multi-objective character of the problems. Also, three versions of the GRASP algorithm are compared for the same optimization problems combining different methods for the constructive and the local search phases.

As referred in [Resende et al. (2012)] one advantage of GRASP is the reduced number of parameters. Only two parameters are defined in the most usual GRASP algorithms: the maximum number of iterations and a parameter that controls the greedy/random selection in the construction phase. Another aspect that may be relevant is the GRASP facility for using parallelism due to the independence in the construction of each solution and its exploitation in the local search procedure.

4.1.1. THE CONSTRUCTION PHASE: SOME DETAILS

The construction phase in GRASP is closely related to the method used to construct the RCL, in particular the number of candidates to be included in the RCL, the level of greediness/randomness imposed to the algorithm, as well as the evaluation of each candidate element.

The number of elements to be included in the RCL may be defined *a priori* or be a dynamic value. Two main schemes to define the number of elements to be included in the RCL were proposed by Hart and Shogan (1987) and Feo and Resende (1989) for single-objective optimization problems: a cardinality based scheme and a percentage based scheme. In the first scheme the number of elements, *k*, to be included in the RCL is pre-defined. In the second scheme this number depends on the value of the greedy function for each candidate element. Only elements with objective function values with α% of the greedy value go to the RCL. The α parameter ($\alpha \in [0, 1]$) controls the level of greediness/randomness of the algorithm. An increase of the α value increases the greediness of the construction phase. Although the percentage based scheme is the most used one, some adaptations of this mechanism have been implemented depending on the applications at hand. In alternative implementations, the value of α can be, for example, a fixed value tuned after a set of initial experiments or a value belonging to the interval [0, 1] uniformly chosen at each iteration of the construction phase. A discussion about several possibilities to define the value of α is presented in [Prais and Ribeiro (1999, 2000)] and an overview can be found in [Pitsoulis and Resende (2002)] and [Resende and Ribeiro (2010)].

Approaches for the construction phase may also differ in the technique used to pick the elements from the RCL. In some cases the selection is not random, as usual, but guided by a bias function [Binato et al. (2001), Bresina (1996)] or optimized using Bayesian Heuristic Approach [Mockus et al. (1997)]. In most construction procedures, the element selected from the RCL is removed from it, but in some cases this does not occur in a deterministic manner and a probability of removal is associated with each element [Bresina (1996)].

All the previous aspects as well as the way to integrate the greediness and the randomness in the construction phase, the possible inclusion of a memory and learning mechanisms to improve the performance of this phase, among others, have led to the proposal of several construction phase procedures. In [Resende and Werneck (2004)] some proposals of alternative construction methods are used for solving a *p*-median problem and a summary of construction approaches used in the single-objective context are described in [Resende and Ribeiro (2010)] and [Resende et al. (2012)].

In addition to the issues referred to above in the context of single-objective optimization, the adaptation of the GRASP construction phase to MOOPs requires taking into account the existence of multiple objective functions in the construction of the RCL. While in single-objective optimization problems the construction is guided by the single greedy function, in MOOPs the multiple objective functions must be considered in the evaluation of elements to be included in the RCL. Different construction methods have been proposed with this aim in a multi-objective context. Martí et al. (2011) classify them as: pure construction and combined construction. In pure construction just one objective function is considered during a single construction. The objective function used can be randomly selected (pure-random construction) or selected using a pre-defined order (pure-ordered construction). In combined construction more than one objective function is considered in each construction phase. The objectives can be combined using a weighted sum or in a sequential way (in each step of the construction process a different objective is selected randomly or sequentially).

4.2. SIMULATED ANNEALING

The Simulated Annealing (SA) is an optimization algorithm based on a physical process, named annealing, consisting in the heating of a material until its fusion temperature so the material passes from the solid to the liquid state. Then the temperature is slowly decreased until the material achieves a minimum energy state. This annealing process was modeled by Metropolis and its co-workers in 1953 [Metropolis et al. (1953)]. Later Kirkpatrick et al. (1983) and Černý (1985) showed, in independent ways, that the referred model could be used to solve combinatorial optimization problems. In this analogy the energy state corresponds to the objective function cost, the slight perturbation in the energy state imposed by the change of temperature is compared to a movement into a neighborhood space, and the minimum energy state achieved at the end of the annealing process may be viewed as the minimum value of the objective function at the end of the optimization process.

The SA algorithm is classified as a metaheuristic belonging to the set of local search algorithms. Initially, SA was created to deal with combinatorial optimization problems. However, despite the large majority of SA applications is in discrete optimization problems, also continuous optimizations problems can be solved. Vanderbilt and Louie (1984), Corana et al. (1987), Siarry et al. (1997), Tekinalp and Karsli (2007), Suppapitnarm et al. (2000) present examples of works dealing with continuous optimization problems.

The first SA approaches were used to solve single-objective optimization problems. In this setting a neighbor solution x' is generated from the current solution x. The new solution x' is accepted as the current solution if it improves the objective function, else it may be accepted based on the value of an acceptance probability function. This function depends on a parameter, usually called temperature (due to the analogy with the physical process), which decreases along the search process. The probability of accepting worse solutions depends on the temperature, and as such also decreases during the search process.

The use of SA to MOOP requires the adaptation of some components from single-objective to multi-objective SA algorithms, namely with respect to the evaluation and selection of neighbor solutions and the definition of the acceptance function. Concerning to the evaluation and acceptance function, two main approaches may be considered: one based on an aggregating function and other based on the non-dominance concept. The first approaches of SA applied to MOOP use an aggregating function to transform a MOOP into a single-objective optimization problem and therefore it is possible to use a similar version of single-objective SA. Serafini (1994) uses a target-vector to solve a bi-objective optimization problem. In this approach an archive is used to store the non-dominated solutions found from the direct comparison of the new neighbor solutions and the current solution. If the neighbor solution is non-dominated with respect to the current solution, it will be included in the archive. Consequently, solutions in the archive may be dominated with respect to each other, requiring a further processing to determine the non-dominated solutions. In [Ulungu (1993), Ulungu et al. (1999)] the proposed multi-objective simulated annealing (MOSA) uses also an aggregating function. A main difference between MOSA and the Pareto Simulated Annealing (PSA) proposed by Czyzak and Jaszkiewicz (1997, 1998) relies on the fact that PSA uses a population instead of a single solution at each iteration. A review of other works ([Hansen (1997), Chang et al. (1998), Lučić & Teodorović (1999), Thompson (2001)]) using aggregating functions can be found in [Coello et al. (2002)]. The Pareto Simulated Annealing has been applied in several MOOPs in different areas and, more recently, Drexl and Nikulin (2008) use PSA to address an airport gate assignment problem with multiple objectives (minimize the number of ungated flights and the total passenger walking distances or connection times as well as to maximize the total gate assignment preferences). Li and Landa-Silva (2011) present EMOSA, which incorporates SA and adapts the search directions (weighting vectors) corresponding to various subproblems. In EMOSA, the weight vector of each subproblem is adaptively modified at the lowest temperature in order to diversify the search towards the unexplored parts of the Pareto-optimal front. Abdelsalam and Mohamed (2013) apply PSA to select a partner in a virtual enterprise considering two main objective functions (project completion time and total cost).

Issues associated with the use of a weighted sum in MOOP are well known and consequently some SA approaches not using an aggregating function have been proposed in the last years. The concept of non-dominance is applied in the acceptance of neighbor solutions and/or in the evaluation of the acceptance function. Ruiz-Torres et al. (1997) use this concept as the selection criterion in a scheduling problem where the objective is to minimize both the number of late jobs and the average flow-time. Suppapitnarm et al. (2000) combine the advantages of a local search with the non-dominance of Pareto. It does not use a “composite” (as called in this work) function, the values of the objective functions being used to decide about the non-dominance of a solution with respect to an archive composed by the non-dominated solutions. To avoid the convergence to local optima and to maintain diversity, the current solution can be replaced by one solution in the archive (the “return to base” strategy). Smith et al. (2004, 2008) incorporate the concept of relative dominance of a solution with respect to the other solutions in the evaluation of neighbor solutions. The relative dominance of a solution is a measure that essentially evaluates the quantity of solutions dominated by this solution. In AMOSA [Bandyopadhyay et al. (2008)] the non-dominated neighbor solutions are included in an archive at each iteration. When the dimension of this set is larger than a pre-defined value a clustering method is applied to guarantee the diversity of the solutions. AMOSA also uses a concept of dominance intensity to measure how strongly a solution is dominated with respect to other solution. This concept is used in the evaluation of the acceptance probability function. In [Singh et al. (2010)] AMOSA is extended for multi-objective constrained optimization problems, using the Constrained Pareto Simulated Annealing (C-PSA). Antunes et al. (2010) also use an archive to maintain the non-dominated neighbor solutions found in the search process. However, in this work a population is used at each iteration. The non-dominance concept is used to compare the neighbor solution with the current solution and also to evaluate its non-dominance with respect to the archive. In [Suman and Kumar (2006)] a survey about single and multi-objective SA is provided.

4.2.1. THE ACCEPTANCE PROBABILITY FUNCTION

SA is a local search method in which neighbor solutions are generated from a current solution. In a local search scheme a solution may get trapped in a local optimum. To overcome this problem SA can accept solutions that do not improve the objective function(s) based on an acceptance probability function.

In single-objective optimization the acceptance function is usually considered as a probability function depending on the temperature and the performance of the neighbor solution in comparison to the current solution. In this case if the neighbor solution improves the single objective function the neighbor solution is accepted, else it can be accepted depending on the value of the acceptance probability function. In its original form the acceptance function is given by (4.1):

$$e^{\frac{\Delta f}{T}} \tag{4.1}$$

where Δf is the difference between the objective function for the original solution and the neighbor solution and T is the temperature.

However, in a multi-objective context three possibilities must be considered: the neighbor solution dominates the current solution, and it is accepted; the neighbor solution and the current solution are non-dominated between them or the neighbor solution is dominated by the current solution. In these two latter situations the new solution may be accepted depending on the value of the acceptance probability function. Consequently, the application of SA in a multi-objective optimization context requires also the adaptation of the acceptance probability function to deal with more than one objective function. Distinct acceptance probability functions have been used and tested in multi-objective SA algorithms. Some of the most usual are: scalar linear, strong and weak rules [Serafini (1994), Kubotani and Yoshimura (2003), Tekinalp and Karsli (2007)]. Defining the difference between the performance of the competing solutions x' and x in objective function j by

$$\delta_j = f_j(x) - f_j(x') \qquad j = 1,\dots,p \tag{4.2}$$

and the aggregation of these differences by the weighted-sum $\Delta = \sum_{j=1}^{p} w_j \, \delta_j$, in which w_j is the "weight" assigned to the objective function f_j, the previous acceptance probability functions are defined as follows:

Scalar Linear: $$prob = \min\left(1, e^{\frac{\Delta}{T}}\right) \tag{4.3}$$

Strong rule: $$prob = \min\left[1, \min_j\left(e^{\frac{w_j \delta_j}{T}}\right)\right] \tag{4.4}$$

Weak rule: $$prob = \min\left[1, \max_j\left(e^{\frac{w_j \delta_j}{T}}\right)\right] \tag{4.5}$$

Ulungu et al. (1999) compare the use of the weak rule with the scalar linear function. The weights are defined based on a minimum satisfaction level for each objective function.

In Serafini (1994) other possible acceptance functions are presented and a new composite function based on the weak rule and the product rule (Equation 4.6) is evaluated. The weights are initialized to one and modified during the search process.

$$prob = \prod_{j=1}^{k} \min\left[1, e^{\frac{w_j \delta_j}{T}}\right] \tag{4.6}$$

Thompson (2001) uses the acceptance function, called simple product by Serafini (1994), and defined as:

$$prob = \prod_{j=1}^{k} \min\left[1, e^{\frac{\delta_j}{T}}\right] \tag{4.7}$$

This acceptance probability function is the product of the single probabilities associated with each objective function.

In Antunes et al. (2010) an acceptance probability function based on the logistic curve is used (Equation 4.8). The difference of performance between the competing solutions is a weighted sum of the difference of the normalized objective function values. This aggregation takes into account the ranges of values that each objective function attains in the non-dominated frontier computed so far (for normalization purposes, thus avoiding the undesirable effects of aggregating objectives functions expressed in different orders of magnitude).

$$prob = \frac{2}{1 + e^{-\frac{\Delta}{T}}} \tag{4.8}$$

4.3. Hybrid Approaches

4.3.1. A Hybrid Evolutionary Simulated Annealing Algorithm

As referred to above one of the most popular hybrid metaheuristics is the combination of EAs with a local search method. EvABOR-III (described in Chapter 3) has been improved with a local search procedure to refine the search of solutions belonging to a certain class of merit. This method is triggered when a solution belonging to a new (higher) class of merit is found. The idea is to intensify the exploitation in the neighborhood of that new solution to increase the convergence to solutions belonging to a better class of merit than the current one. The local search method is based on SA due to the good results it has provided in

MOOPs. The preferences elicited from the DM are also incorporated in the probability acceptance function in SA.

The main structure of the EvABOR algorithm with the SA hybridization, called Hybrid Evolutionary Simulated Annealing (HESA) algorithm, is presented in the flowchart in Figure 4.1. After the crossover and the mutation operators are applied, the non-dominated offspring are obtained and classified in classes of merit. If any offspring belongs to a higher class of merit (regarding the best class in the previous iteration), SA is applied to exploit the neighborhood of these offspring (the "best" offspring).

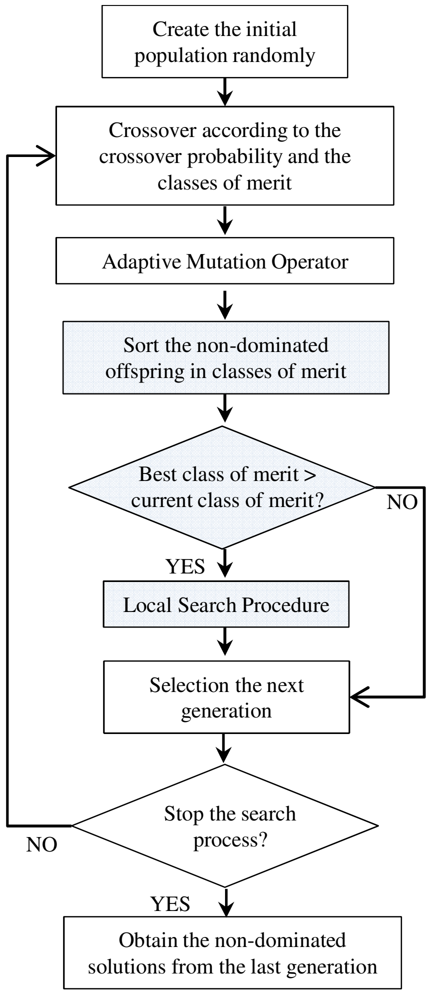


Figure 4.1 – Flowchart of HESA.

SA WITH INCORPORATION OF PREFERENCES

The exploitation of the neighborhood of each "best" offspring is performed for each temperature value (as in classical SA), which is successively decreased until a minimum value. The temperature decrease implies that the probability to accept a solution for further exploration also decreases, as usual. An archive is used during the local search to save the solutions belonging to the best class. This archive is initialized with offspring that belong to the new best class of merit found after crossover and mutation operators are applied. The neighbor solutions of each offspring in the archive are obtained using a set of neighborhood structures specifically designed for the problem at hand and are classified using the ELECTRE TRI method using the preferences elicited. This preference information included in the probability acceptance function is also used to decide about the exploitation of neighbor solutions. As the aim of this local search is to increase the convergence to a region more in accordance with the preferences elicited from the DM, it makes sense that the preference information is used during the search process. Three main cases are considered in this phase:

1) If the new neighbor solution dominates (consequently it belongs to the same or a higher class of merit) the current solution, this is replaced by the neighbor solution in the archive.
2) If the neighbor solution and the current solution are non-dominated then the acceptance depends on the quality of the solution according to the preferences elicited. If the class of merit of the neighbor solution is not inferior to the class of the current solution then the neighbor solution is added to the archive. If the class of merit of the neighbor solution is inferior to the class of the current solution then the neighbor solution can be accepted depending on the value of the credibility degree. This value is used in ELECTRE TRI to classify a solution into a class of merit in comparison with the cutting-level λ, so it can be viewed as an indicator of the quality of the solution. Additionally, being a value within [0,1], it is suitable to be used as an acceptance function.
3) If the neighbor solution is dominated with respect to the current solution then it can yet be accepted for further exploitation. If the neighbor and current solutions belong to the same class of merit then the acceptance depends on the credibility degree (as in the previous point), else a probability acceptance function is used (in this case the current solution belongs to a higher class than the neighbor solution).

Before the exploitation of a solution in the archive the non-dominated solutions are filtered. The dimension of the archive is reduced and priority is given to the non-dominated solutions belonging to the best (and new) classes of merit.

The pseudo-code of SA with the incorporation of preferences is shown in Figure 4.2.

```
temp = t0
archive = set of "best" solutions
while temp >= final temperature
  for i = 1 to #(archive)
    current solution = archive(i)
    while current solution≠{}
      Select randomly a neighborhood structure
      Obtain the neighbor solution
      Evaluate the neighbor solution
      Determine the class of the neighbor solution
      if neighbor solution dominates current solution
         archive(i) = current solution
         current solution = {}
      end
      if neighbor and current solutions are non-dominated between them
         if class(neighbor solution)>= class(current solution)
            archive = archive ∪ {neighbor solution}
            current solution = {}
         else
            if random value in [0,1]<= credibility degree
               current solution = neighbor solution
            else current solution = {}
            end
         end
      end
      if neighbor solution is dominated by current solution
         if class(neighbor solution)= class(current solution)
            if random value in [0,1]<= credibility degree
               current solution = neighbor solution
            else current solution = {}
            end
         else
            if random value in [0,1]<= probability function
               current solution = neighbor solution
            else current solution = {}
            end
         end
      end
    end while
    archive = non-dominated solutions of the archive
  end for
temp = coef*temp
end while
```

Figure 4.2 – Pseudo-code of SA with incorporation of preferences.

4.3.2. GRASP+SA

In metaheuristics a usual process to obtain the initial solutions is randomly generate one or more solutions. However, it is known that the quality of initial solutions may influence the global performance of metaheuristics. Consequently, whenever information about the characteristics of the problem or about preferences regarding the final solutions is available, it may be advantageous to use this information in the construction of solutions.

In this hybrid approach, this idea is included in a construction phase. The procedure implemented is based on the principles of the GRASP algorithm (Section 4.1.1). For this purpose different RCLs are used, each one based on a different objective function or on a specific characteristic of the problem. If preferences are elicited from a DM, these may also be used to sort the candidate elements in the RCL.

The construction phase is followed by a local search to exploit the neighborhood of each solution. The local search is done according to the algorithm presented above. These phases are repeated until the stop condition is achieved (e.g., a maximum number of iterations, all solutions belong to the best class of merit).

The selection of solutions that go to the next iteration is done using the procedure presented in Section 3.2.3. The pseudo-code of this hybrid algorithm is displayed in Figure 4.3.

```
while not stop condition do
    Greedy randomized construction phase
    Local search phase
    Update non-dominated solutions
    Selection of the next generation
end
Update the non-dominated set of solutions
```

Figure 4.3 – Pseudo-code of GRASP+SA.

4.3.3. PARALLELISM IN THE PROPOSED APPROACHES

One the advantages associated with metaheuristics is the possibility to use parallelism in their implementation. Although this aspect is not as explored as others in this area, some works use this mechanism. Some references about such works can be found in [Coello (2002), Konak et al. (2006), Talbi (2009), Resende and Ribeiro (2010)]. The importance of the parallelism led Raidl (2006) to consider it as a criterion in his classification of metaheuristics (Figure 2.1) depending on the way memory is used (shared or distributed) and data allocation is done (dynamic or static), among others. Talbi (2009) also explores this aspect in the classification proposed and presents some mechanisms used in the implementation of parallel metaheuristics.

In this work parallelism is used in the three algorithms to decrease the computational time. This aspect is particularly important in the second case-study (the direct load control problem) in which physically based load models are used in the evaluation of solutions. The evaluation done in parallel decreases substantially the computational time involved in this process. In EvABOR approaches parallelism is used in the evaluation of solutions. In the HESA and the GRASP+SA algorithms parallel computation is also used in the exploration of the neighborhood of each solution and in the construction phase of GRASP+SA.

4.4. The Direct Load Control Problem

4.4.1. Case Study

The changes in the electrical sector pushing the evolution of power systems towards smart-grids, with the increasing integration of information and communication technology and the dissemination of distributed generation, allows the use of demand as a resource to increase energy efficiency [Mohsenian-Rad et al. (2010), Karangelos and Bouffard (2012)]. In this setting, demand is viewed as a resource with some potential of management and control, by changing the usual demand patterns without degrading the quality of the energy services provided. Demand-side management (DSM) programs have been used by utilities to modify the load pattern of consumers, either for operational benefits (e.g. increase load factor, reduce peak power demand or reliability concerns) or supporting energy or environmental policies.

DSM programs can be classified in two main groups: price-based actions (consumers are encouraged to reduce peak demand taking advantage of variable electricity prices thus reducing their electricity bills) or incentive-based programs (giving customers some financial stimuli for reducing peak demand during critical periods) [US Department of Energy (2006)]. Direct load control (DLC) actions are one of the possibilities to change consumption patterns. In these programs consumers are stimulated to modify their load pattern to reduce peak load especially in critical periods by turning off some end-use loads during short periods of time. Loads providing an energy service that can be temporarily interrupted or deferred in time without decreasing the quality of service provided are suitable for being used in these programs. Thermostatic loads (air conditioners, heat pumps, electric water heaters and electric space heaters, for example) are loads with these characteristics and are commonly used in this kind of programs. However, control actions must be carefully implemented to prevent some undesirable effects, for instance, the so-called payback effect. This effect may occur when the power is restored to the controlled loads leading to an increase in peak power demand. Other undesirable consequences are a possible strong decrease in profits and the increase of the discomfort caused to consumers. This aspect is very important for the success of DSM programs, since it may impact negatively the willingness of consumers [Jorge et al. (2000)]. Consequently, the minimization of the discomfort as well as the maximization of profits have been considered as objective functions in the search and identification of load control strategies [Ng and Sheble (1998), Gomes et al. (2004, 2007, 2012)]. Besides, the minimization of peak demand and the minimization of operational costs are the most usual objectives in DSM programs, but the minimization of losses as well as the minimization of spinning reserve have been also considered [Gooi et al. (1999), Gomes et al. (2004, 2007, 2008, 2012)]. Despite in the literature some works are only focused on the selection of a control strategy [Bhatnagar and Rahman (1986), Kurucz et. al. (1996)], the design of a set of solutions guaranteeing the achievement of multiple objectives are crucial for the success of a load management program. The appropriate assessment of impacts at different levels of aggregation of demand (Figure 4.4), and therefore according to the interests of different potential economic actors in the various branches of activity in the electricity market, must be taken into account in the design of control strategies.

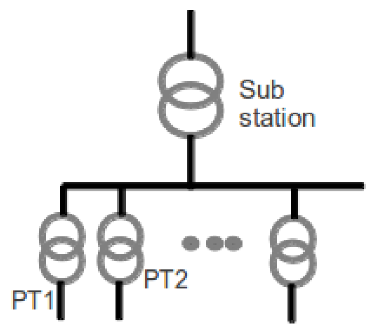


Figure 4.4 – Example of a sub-station and distribution power transformers.

In this chapter the aim is to design and select load control strategies to minimize the maximum peak power demand in a sub-station (SS) and in two distribution power transformers (PT1 and PT2), to maximize profits with the sale of electricity, to minimize the potential discomfort caused to consumers and the total time in which loads are in curtailment. The discomfort is assessed by a variable (usually associated with temperature) and increases whenever it is over or under a pre-specified threshold. To address this aspect, two objective functions are considered: the maximum continuous time interval in which discomfort has occurred and the total time of its occurrence.

The loads to be controlled are 500 air conditioners (ACs) units, grouped in 24 groups, as presented in Table 4.1. This table also presents some characteristics of the controlled loads. Loads are usually grouped according to the similarity of their characteristics and the geographic proximity. The demand patterns of ACs are obtained by simulation using physically based load models [Gomes et al. (2004, 2007)], in three aggregation levels: demand in PT1 (PT1), demand in PT2 (PT2) and aggregated demand. The use of physically based load models, which reproduce the physical phenomena occurring in loads, is an important tool (despite the additional computational effort) for assessing the impacts of control actions, including potential undesirable effects (as for example, the payback effect).

Groups	**ACs (#)**	**Power (kW)**	**PTs/SS**
1	15	1,6	PT2
2	20	2,8	PT1
3	30	2,7	SS
4	20	4	SS
5	25	1,6	SS
6	20	1,4	SS
7	35	2,4	SS
8	20	2,4	SS
9	20	4	PT1
10	30	3,8	SS
11	15	1,8	PT2
12	25	8	SS
13	30	2,4	SS
14	20	3,4	PT2
15	30	2,4	SS
16	25	1,6	SS
17	10	2,4	PT1
18	15	1,4	PT1
19	15	1,8	PT2
20	15	1,6	PT1
21	10	1,8	PT2
22	15	1,4	PT1
23	20	1,4	SS
24	20	2,4	SS

Table 4.1 – Some characteristics of controlled loads.

Typical load diagrams in each one of the aggregation levels are displayed in Figure 4.5. It is important to note that there are non-controlled loads (NCLs) that contribute to increase the maximum peak power demand in the different aggregation levels. These loads are also displayed in Figure 4.6 to Figure 4.8 and the maximum power in each one of the aggregation levels is presented in Table 4.2, allowing the comparative analysis between the load demand by the ACs and the non-controllable loads.

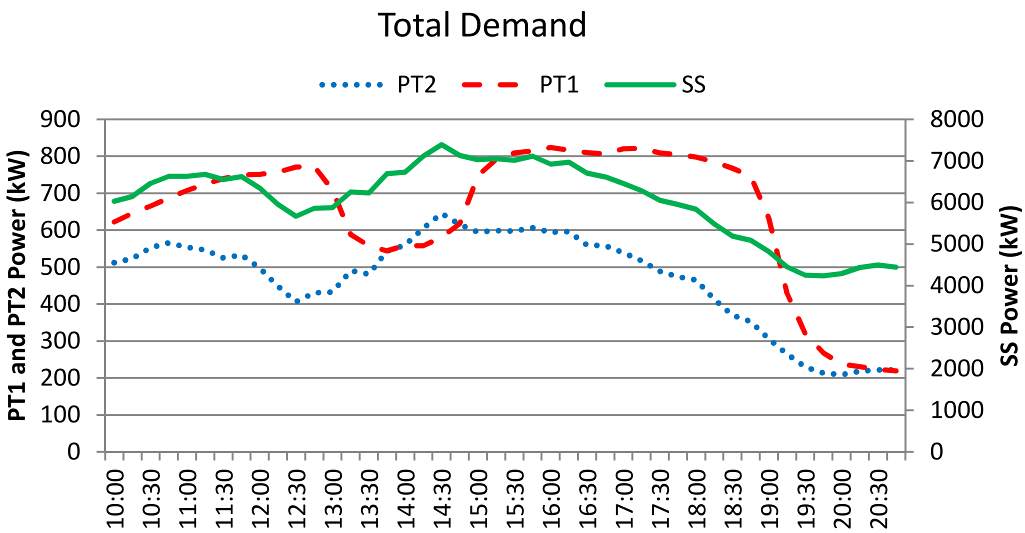


Figure 4.5 – Total demand in each one of the aggregation levels.

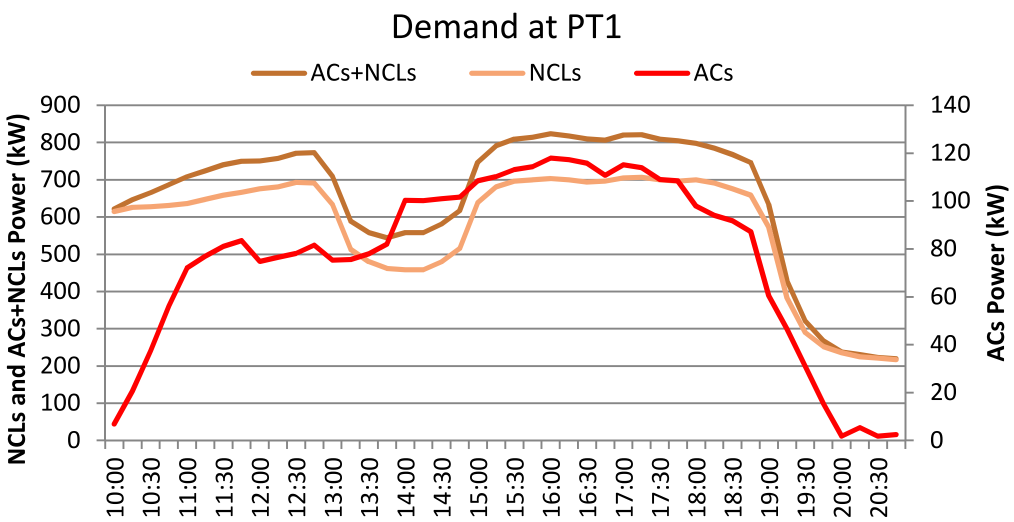


Figure 4.6 – Load diagrams at PT1.

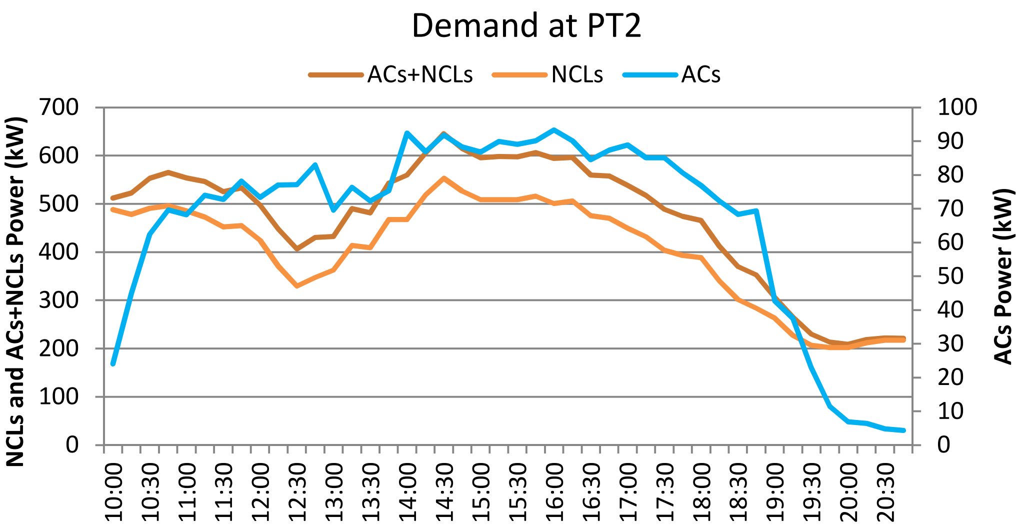


Figure 4.7 – Load diagrams at PT2.

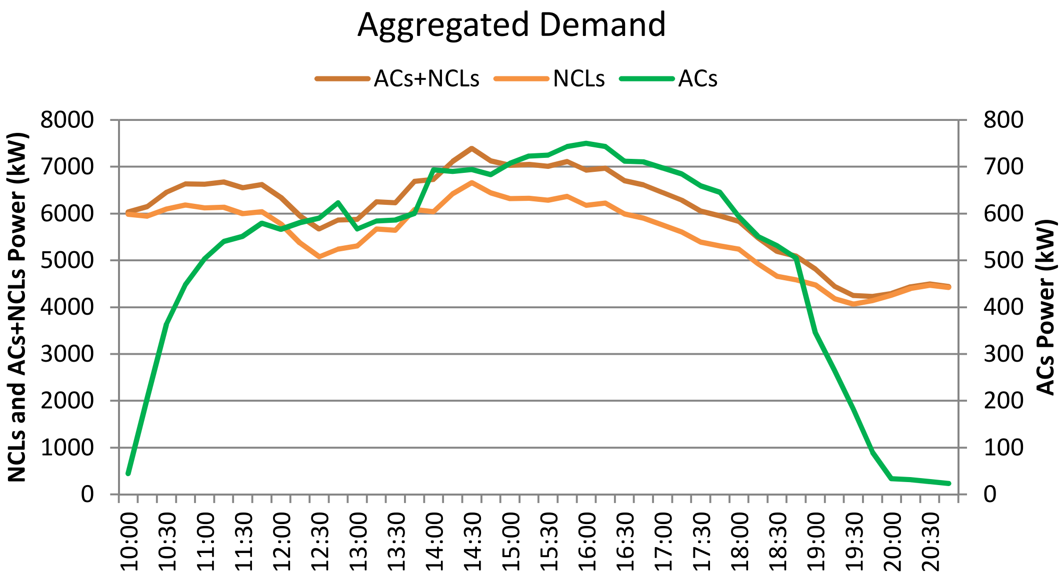


Figure 4.8 – Load diagrams at SS.

The curtailment actions have been applied in the period of time between 10:00 to 21:00 having in mind the objective functions to optimize and the usual period in which air conditioners are in use. The simulation restricted to this interval of time reduces the computational effort required from the physically based load models.

		Maximum (kW)
Air Conditioners (ACs)	PT1	117,973
	PT2	93,333
	SS	750,073
Non-Controlled Loads (NCLs)	PT1	707,000
	PT2	553,000
	SS	6660,139
ACs + NCLs	PT1	823,600
	PT2	645,333
	SS	7394,072

Table 4.2 – Maximum power in each one of the aggregation levels.

4.4.2. IMPLEMENTATION DETAILS

Finding solutions to the problem previously described is, in general, a hard task due to the multiple and conflicting objectives considered and the combinatorial nature of the problem. One control strategy encompasses the control actions (on/off patterns) to be applied to all groups and the same control actions are applied to the loads belonging to the same group. The two approaches described in Section 4.3 (HESA and GRASP+SA) are used to search and identify a set of curtailment patterns to apply to each group of loads and their results are also compared with the ones obtained using EvABOR-III.

For both algorithms the encoding of solutions is the same: a bi-dimensional array (number of groups x period of time) in which each row corresponds to each group during the period under evaluation. The binary representation is used to capture the on/off patterns to be applied to each group: a "1" in the ij^{th} position in the array means that a curtailment action is applied to group i in the instant of time j while a "0" means that no action is applied in the same instant.

THE EVABOR-III ALGORITHM

The implementation of the crossover operator in EvABOR-III to be used in the DLC problem is similar to the one implemented for the reactive power compensation problem, i.e. a 2-point crossover is used. However, in this case, the implementation is carefully designed to preserve the curtailment patterns of each group. Figure 4.9 exemplifies the structure used in the crossover operator to obtain an offspring. The selection of individuals that go to the next generation is the one described in Chapter 2. In the experiments presented in this chapter the value of the intra-class elitism is equal to 0.5. Concerning the mutation operator, an adaptive mutation operator, already used with the same case study in previous works with proven results [Gomes et al. (2008)], replaces the original mutation operator of EvABOR-III. As a binary representation is used there are two possibilities to occur a mutation in each gene (corresponding to an instant of time): "0" mutates to "1" and "1" mutates to "0". Two different mutation probabilities are associated with each one of the previous mutations. The performance of the solution in each objective function is used to compute the probability values. Details about this mutation operator can be found in [Gomes et al. (2008)].

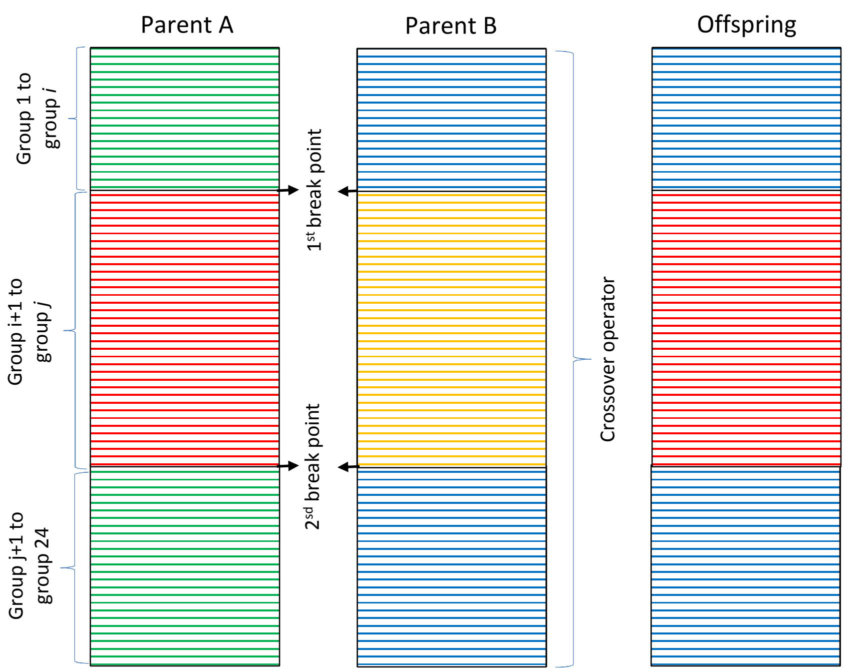


Figure 4.9 – Scheme of the crossover operator to the DLC problem.

THE HESA ALGORITHM

The genetic operators, crossover, mutation and selection, implemented in the HESA algorithm used in this case study are the same described previously and implemented in EvABOR-III. The exploitation of the solutions belonging to a new class of merit in the local search phase is carried out using the following different neighborhood structures specifically designed for the problem at hand:

1. Reduction of the maximum off interval: the largest off interval is chosen (randomly if there is a tie) and a reduction within the range [1…5] minutes (randomly chosen) is applied at the right or at the left of the pattern. The aim of this neighborhood structure is trying to reduce the maximum time interval and the total time in which

discomfort occurs. Moreover, this neighborhood structure also contributes to reduce the total time in which loads are in curtailment.

2. Increasing the minimum off interval: the minimum off interval is chosen (randomly if there is a tie) and a period (between 1 to 5 minutes) of curtailment is added at the right or at the left of the minimum interval. This neighborhood structure avoids the existence of potentially ineffective curtailments due to too short duration and prevents high on/off switching rates.
3. Shifting part of the curtailment pattern of a group: the shift dimension is selected randomly within the range [2-6] minutes and may be done to the right or to the left of the curtailment pattern. This neighborhood structure is applied to a maximum of eight groups chosen randomly and it produces small adjustments in the curtailment pattern. These adjustments can be useful to prevent the payback effect if the neighborhood structure generates solutions with diversified curtailments.
4. Exchanging curtailment patterns between groups supplied by the same PT: within a control strategy, two groups of loads supplied by the same PT are selected randomly and their control patterns are exchanged.
5. Exchanging curtailment patterns between any groups. In this case the existing curtailment patterns may be exchanged between any groups in every control strategy. The aim of this structure is to improve the diversity of the set of solutions.

The first and the second neighbor structures are applied to a minimum of 10 groups (the number is chosen randomly). Experiments have demonstrated that if these structures are applied to a small set of groups the variation of the objective functions is negligible.

In the local search of the HESA algorithm, if the neighbor solution is dominated by the current solution and this belongs to a higher class than the neighbor solution, the acceptance of this neighbor solution for exploration depends on the value of the credibility degree (as a way to assess the quality of the solution regarding the preferences elicited) and on the SA temperature. In Table 4.3 the values of the acceptance probability function considered in this section (temperature × credibility degree) are presented considering some examples of the credibility degree and the temperature. The probability to accept a neighbor solution decreases in two situations:

1. For the same value of the credibility degree with the decrease of the temperature.
2. For the same value of the temperature with the decrease of the credibility degree.

Credibility degree	Temperatures values						
	2	**1.6**	**1.024**	**0.65536**	**0.4194304**	**0.268435456**	**0.109951163**
0.0739217	0.147843	0.118275	0.075696	0.048445	0.031005	0.019843	0.008128
0.1573922	0.314784	0.251827	0.161170	0.103149	0.066015	0.042250	0.017305
0.2711419	0.542284	0.433827	0.277649	0.177696	0.113725	0.072784	0.029812
0.3648516	0.729703	0.583762	0.373608	0.239109	0.153030	0.097939	0.040116
0.4000000	0.800000	0.640000	0.409600	0.262144	0.167772	0.107374	0.043980
0.4144051	0.828810	0.663048	0.424351	0.271585	0.173814	0.111241	0.045564
0.6683639	1	1	0.684405	0.438019	0.280332	0.179413	0.073487
0.7736972	1	1	0.792266	0.507050	0.324512	0.207688	0.085069
0.8392210	1	1	0.859362	0.549992	0.351995	0.225277	0.092273
0.9740781	1	1	0.997456	0.638372	0.408558	0.261477	0.107101
1	1	1	1	0.655360	0.419430	0.268435	0.109951

Table 4.3 – Examples of values of the acceptance probability function.

These two situations are in accordance with the preferences elicited since a decrease in the value to the credibility degree corresponds to a decrease in the acceptance probability, as well as with the underlying idea of the SA algorithm since the acceptance probability also decreases with the decrease of the temperature.

This acceptance probability function has the advantage of being easy to evaluate and also reflecting the effect of both the temperature and the elicited preferences.

GRASP+SA ALGORITHM

The GRASP+SA algorithm consists in two main phases: a construction phase where a set of solutions (curtailment patterns) is generated based on the knowledge about the direct load control problem and a local search phase to exploit these solutions. These two phases are repeated until a maximum pre-defined number of iterations is achieved or all solutions in the population belong to the best class of merit.

In the construction phase of GRASP each element of a solution must be included in an incremental manner, as described in Section 4.1.1. In this case study the “atomic” element (the gene of the individual) is each minute in which a curtailment may be applied, as the size of a single potential solution is 24 (groups) x 660 (minutes). However, if the RCLs are composed by this type of candidate elements it will be impracticable, in real-time, to construct solutions due to the combinatorial characteristics of the problem and to the time required to evaluate each solution. Consequently, the candidate elements considered in this approach are possible curtailment patterns for each group.

In the construction phase the following steps are repeated until a curtailment pattern for all groups is obtained:

1) Randomly select a group in the range [1-24] (total number of groups). Each group is selected just once;
2) Obtain several curtailment patterns solutions for the current group using different methods;
3) Evaluate these solutions;
4) Construct the RCL;
5) Select the element from the RCL according to the value of the α parameter.

To improve the quality of the solutions these steps can be repeated using the curtailment pattern obtained in the previous iteration of the current construction phase. A balance between the number of iterations of the construction phase and the quality of solutions according to the preferences elicited must be considered due to the computational time required in the evaluation of solutions. The pseudo-code of the construction phase is presented in Figure 4.10. The number of solutions to be generated in the construction phase is an input parameter of the GRASP+SA algorithm and it may be different from the dimension of the population. The construction phase procedure is called for each solution to obtain in this phase. This task can be performed in a parallel mode.

In step 2) of the construction phase, two methods to construct a curtailment pattern to each group have been tested: patterns based on the adaptive mutation operator (used in EvABOR-III and in the HESA algorithm) and patterns based on mutation probability evaluated using a pre-defined threshold (a percentage of the maximum peak power demand in each PT). A third approach has been tested: cyclic patterns with diverse on/off periods. Despite this strategy has been used in real world, after a few initial experiments it has been concluded that solutions obtained using cyclic patterns performs worse, so the other two approaches have been considered in the experiments presented in this chapter.

The on/off patterns identified resorting to the adaptive mutation operator are applied during the period of time between 10:00 to 21:00, meaning that in this type of construction curtailments can be applied at any instant of time, `i`, according to the probability value. In the other approach based on a pre-defined threshold, curtailments are applied in a more restrict period of time, as control actions are only applied when demand is higher than the pre-defined threshold, which is usually identified as a percentage, `perc`, of the maximum peak power demand in the sub-station and in two distribution power transformers (Figure 4.11). For each instant of time and for each transformer (SS, PT1 and PT2) two probability values (the probability of change from “1” to “0” and the probability of change from “0” to “1”) are evaluated as displayed in pseudo-code of Figure 4.12.

```
for k=1:num_internal_cycles do
    list of groups={1,2,…,23,#groups}
    while list of groups~={} do
        Randomly select a group i from the list of groups
        Remove group i from the list of groups
        Obtain several curtailment patterns (solutions) for the
        group i using different methods
        Evaluate solutions
        Construct the RCL
        Select the element from the RCL according to the value of
        the α parameter
    end
end
```

Figure 4.10 – Pseudo-code of the construction phase.

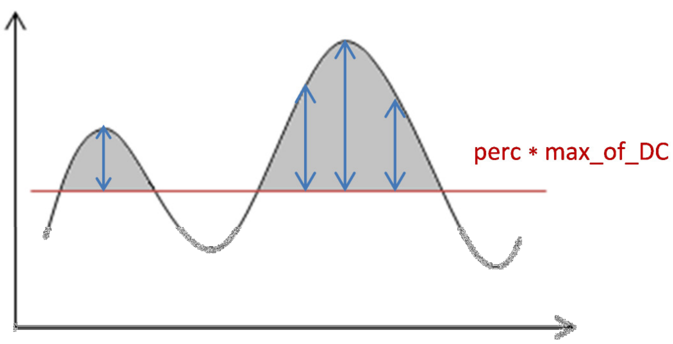


Figure 4.11 – Example of a period of time defined using a pre-defined threshold.

```
aux_superior[i]= max(0, demand_in_instant[i]-perc*max_of_DC)
aux_inferior = max(aux_superior[i])
Prob_01_PT[i]= aux_superior[i]/aux_inferior*max_prob_mutation
Prob_10_PT[i] = (1-aux_superior[i]/aux_inferior)*max_prob_mutation
```

Figure 4.12 – Pseudo-code to evaluate the mutation probability using a pre-defined threshold.

In Figure 4.12 `perc` is the percentage (80% in the experiments presented in the next section) of the maximum peak power, `max_of_DC` is the maximum peak power demand and the `max_prob_mutation` is the maximum mutation probability value. The value of this parameter is equal to 0.01 in the evaluation of the mutation probability based on a pre-defined threshold, as well as in the other algorithms using the adaptive mutation operator. This method of construction of curtailment patterns focuses the actions on the periods of time where the power demand is higher, while the first method allows spreading the curtailments along all period of time, thus complementing the first one. As can be observed, comparing Figure 4.13 and Figure 4.14, the curtailment patterns after the construction phase reflects the characteristics of the load diagrams at each transformer in the sense that a large number of curtailments is applied in periods in which the power demand is higher. Note the difference between the curtailment patterns applied to the groups supplied by PT1, where the curtailment actions are more expanded in time, when compared to the other groups. In Figure 4.14 this effect is particularly evident in groups 2, 18, 20 and 22.

The local search phase and the selection of solutions that pass to the next generation in GRASP+SA are the same described in the HESA algorithm.

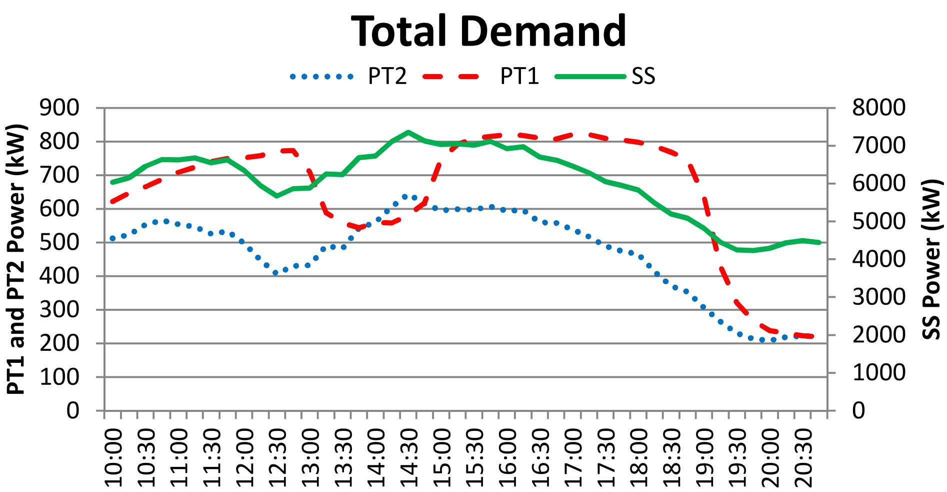


Figure 4.13 – Total demand in each one of the aggregation levels.

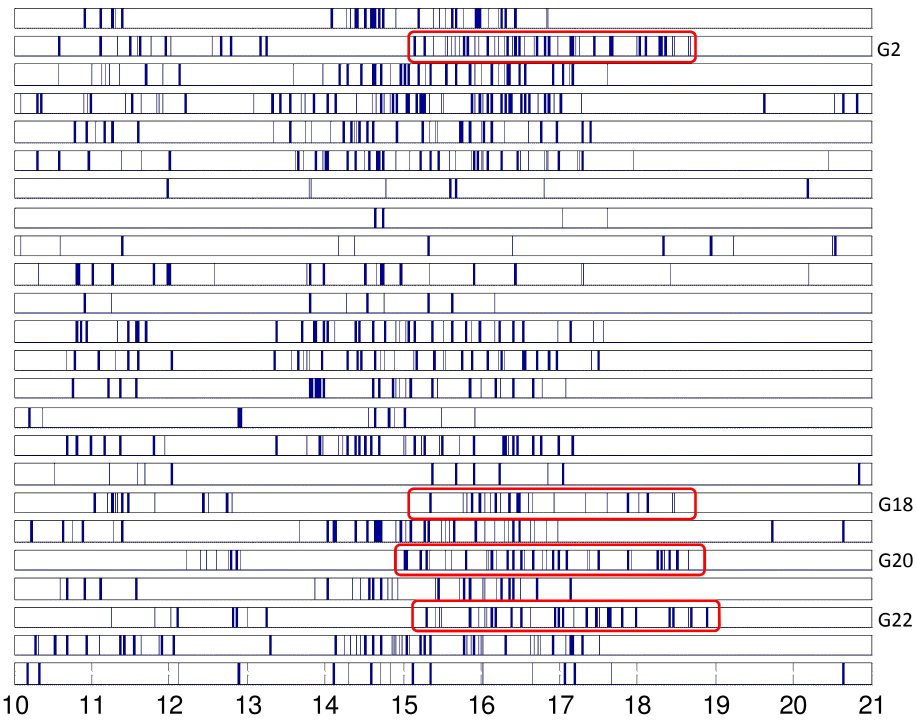


Figure 4.14- Example of a curtailment pattern obtained in the construction phase.

4.4.3. Analysis of Results

This section begins by analyzing some particular aspects of each proposed algorithm and then a comparative analysis between EvABOR-III, HESA and GRASP+SA algorithms is performed.

In this section the analysis of the results is done using the preferences captured by the ELECTRE TRI parameters presented in Table 4.4. The thresholds considered are defined as a percentage of the difference between the reference profiles. The dimension of the population considered in these experiments is 30 individuals, the crossover probability is set equal to 1, the maximum mutation probability is 0.01 and the maximum number of iterations is 4000 for EvABOR-III and HESA and 10 for GRASP+SA. The lower maximum number of iterations in GRASP+SA is due to the usage of internal cycles aimed at improving the solution during the construction phase. In this phase, only 10 solutions are generated due to the computational effort required in the evaluation of solutions. However, for comparing with the other two algorithms the dimension of the final population is composed by 30 individuals. The remainder solutions are obtained by means of local search.

		Maximum Power at PT1 (kW)	Maximum Power at PT2 (kW)	Maximum Power at SS (kW)	Profits (Euros)	No. Minutes in curtailment	Max. Interval	Total Time in discomfort
Reference Profiles		815 820	610 620	7100 7250	1500 1800	1000 1071	5 10	40 80
Weights		100/7	100/7	100/7	100/7	100/7	100/7	100/7
Thresholds	Indifference	10%						
	Preference	30%						
	Veto	80%						
λ		0.7						

Table 4.4 - ELECTRE TRI parameters.

IMPROVEMENT WITH THE LOCAL SEARCH PHASE

In some runs, EvABOR-III has presented some difficulties to complete the population with solutions belonging to the best class of merit after the achievement of one solution in this class. This motivated the development of a local search phase to be included in the EvABOR-III algorithm. Two examples of this fact are displayed in Figure 4.15:

- In the first one, the algorithm obtains all individuals of the population belonging to the best class of merit but the computational effort is too high. This situation is presented in the first column of Table 4.5, in which it is possible to observe that the algorithm needs more 2359 iterations to complete the population with solutions in the best class of merit after the first one has been found.
- In the second example, EvABOR-III obtains the first solution belonging to class 3 in the 934th iteration but until the final of the run only two more solutions in this class are obtained.

This situation had already been observed in the previous case study, but the execution time is not so problematic due to the characteristics of the problem. However, in the DLC problem, the use of physically based load models in the evaluation of solutions substantially increases the execution time, which may lead to a prohibitive computational effort, especially if solutions for practical implementation need to be computed in due time.

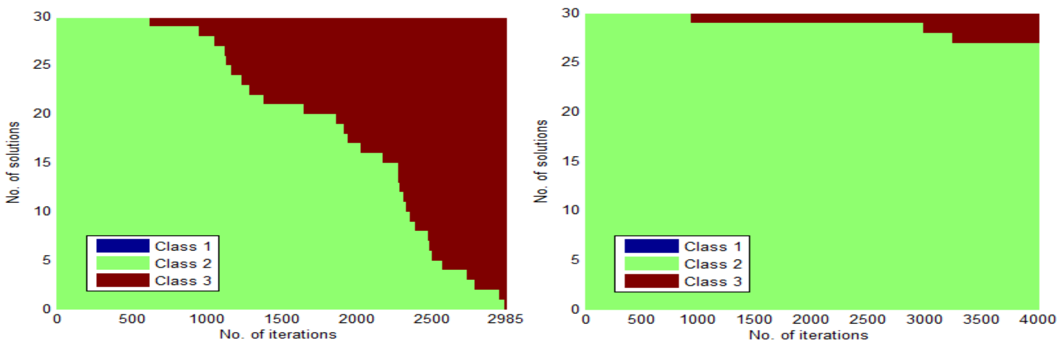


Figure 4.15 – Examples of the evolution of classes of merit in EvABOR-III.

Table 4.5 and Table 4.6 present similar situations to the EvABOR-III and HESA algorithms regarding the first iteration in which these algorithms obtain the first solution belonging to the best class of merit. Comparing, for example, the first column of both tables, the first iteration with at least one solution in the best class of merit is around the 600^{th} iteration. In this case, both algorithms obtain all solutions belonging to the best class of merit, but the execution time and the number of solutions evaluated in EvABOR-III are much higher than in the HESA algorithm. The comparison of the corresponding columns of both tables allows to conclude that the incorporation of the local search phase in EvABOR-III clearly improves its efficacy: it is possible to find all solutions in the best class of merit with HESA in much less time and in similar conditions EvABOR-III does not achieve this performance. The values shown in Table 4.5 and Table 4.6 for some particular runs are also presented in Tables B.1 and B.2 in Appendix B for the 30 runs considering in this analysis.

First iteration with at least 1 solution in class 3	626	272	934	1667	2540
Last iteration	2985	931	4000	4000	4000
No. of iterations to complete the population with solutions in class 3	2359	659	-	-	-
No. of solution evaluations	89551	27931	120001	120001	120001
Total execution time (second)	9833	3069	12902	12948	13121
Obtained classes	3	3	2 and 3	2 and 3	2 and 3

Table 4.5 – Values obtained with EvABOR-III.

First iteration with at least 1 solution in class 3	696	247	1001	1857	2539
Last iteration	696	247	1001	1857	2539
No. of iterations to complete the population with solutions in class 3	0	0	0	0	0
No. of solution evaluations	22310	8733	31633	56350	77425
Total execution time (second)	2555	1006	3437	6011	8529
Obtained classes	3	3	3	3	3

Table 4.6 – Values obtained with HESA.

Construction Phase of the GRASP+SA Algorithm

After the HESA algorithm finds the first solutions belonging to the best class of merit, more solutions belonging to the same class are easily obtained in the local search phase. However, as it is possible to observe from Table 4.5 and Table 4.6, the solutions belonging to the best class of merit may be firstly obtained at the initial iterations (97^{th} and 626^{th} iteration, for example) or at a later stage of the algorithm (2539^{th} iteration, for example). The aim of the construction phase is to establish a procedure that somehow guarantees more consistently the finding of solutions belonging to the best class of merit or, at least, more in accordance with the preferences elicited, using the available knowledge about the problem.

In GRASP+SA the construction phase can be improved using internal cycles. To assess the number of these cycles some experiments have been done considering 1, 3 and 6 internal cycles in the construction phase. Table 4.7 presents a set of results obtained using 1 and 3 internal cycles with 30 runs. A statistical data representation of this information is presented in Figure 4.16 and Figure 4.17. For the case of 6 internal cycles only a few runs have been done, since it has proved to be too computationally expensive without significant improvements in the construction of solutions. The values presented in Table 4.7 are obtained from the information available in Table B.4 and Table B.5 in Appendix B, in which the values corresponding to the 30 runs are displayed.

When 3 internal cycles are used, the percentage of runs only with solutions in the best class of merit in the front (93.33%) is significantly larger than when using 1 internal cycle (40%). Concerning the number of evaluated solutions and the execution time, two different cases must be analyzed: the set of all runs and the set of runs in which only solutions in the best class of merit are obtained in the final front. In the first case, the number of evaluated solutions and the execution time increase in average around 40%, when the number of cycles increases from 1 to 3. However, if the median is considered, those values are similar. In the second case, the number of evaluated solutions and the execution time increase, in average, also in the order of 40%, when 3 internal cycles are used. If the value of the median is considered, the increase is around 12%. Despite the increase of these values, the use of 3 internal cycles is justified due to the largest number of runs with solutions in the best class of merit only.

A relevant aspect to refer is that the solutions belonging to the best class of merit are always firstly found in the construction phase, underlining the importance of this phase. After this, the remaining solutions to complete the population are obtained in the local search phase, in the same iteration.

Maximum no. of iterations of the cycle in the construction phase		**1**	**3**
Classes in the non-dominated front		2 and 3	2 and 3
% of runs with solutions belonging to class 3 only in the non-dominated front		40.00%	93.33%
% of runs with solutions belonging to class 2 only in the non-dominated front		60.00%	6.67%
Number of solutions evaluated in all runs	Minimum	12023	17826
	Maximum	34634	91373
	Average	27506.53	39024.83
	Median	29881	29365
	Standard deviation	5404.72	18281.3
Number of solutions evaluated in runs with solutions belonging to class 3 only	Minimum	12023	17826
	Maximum	32950	64095
	Average	25139.94	35285.67
	Median	25638.5	29179.5
	Standard deviation	5804.99	12180.26
Execution time of all runs	Minimum	2137	3414
	Maximum	6826	17809
	Average	5279.23	7520.8
	Median	5744	5565.5
	Standard deviation	1120.23	3559.78
Execution time of the runs with solutions belonging to class 3 only	Minimum	2137	3414
	Maximum	6323	12312
	Average	4761.94	6786.18
	Median	4924	5524.5
	Standard deviation	1170.63	2341.39

Table 4.7 – Values from 30 runs of GRASP+SA.

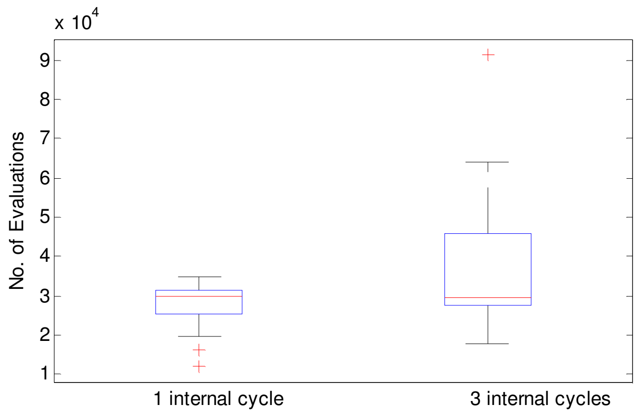


Figure 4.16 – Statistical data representation for the number of solution evaluations.

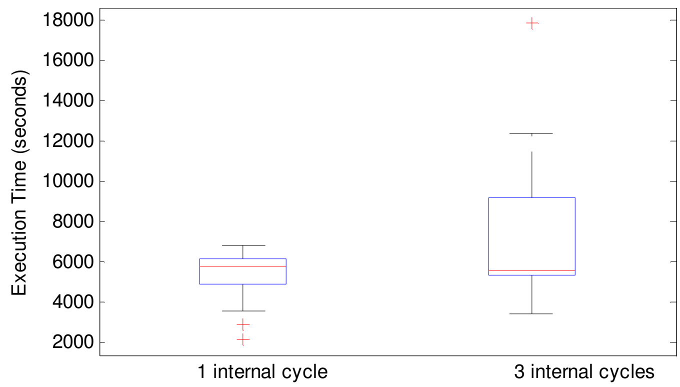


Figure 4.17 – Statistical data representation for the time execution.

To reinforce the previous conclusions about the number of the internal cycles a set of experiments with different preference information (Table B.3 in Appendix B) is performed. An analysis of the 30 runs obtained with these preferences is presented in Tables B.6 and B.7 in Appendix B. Despite the number of evaluated solutions and the execution time have increased significantly when 3 internal cycles are used, this option continues to be preferable due to the number of iterations in which solutions in the best class of merit are obtained.

A Comparative Analysis of the Proposed Algorithms

In this section a comparison of the three algorithms implemented to provide decision support in the DLC problem is presented. The preference information used for this analysis is displayed in Table 4.4. In the GRASP+SA algorithm three internal cycles are considered, according to the results presented in the last subsection.

The different approaches have been compared according to the classes of merit in the non-dominated front, the percentage of non-dominated solutions belonging to each class, the total number of solutions evaluated and the execution time. The latter two aspects are assessed considering all the runs performed and also the runs in which solutions in the best class of merit only are obtained. The results presented are obtained from an average of 30 runs for each algorithm implemented.

Comparing the algorithms regarding the quality of solutions according to the preferences elicited, it is possible to conclude about the efficiency of GRASP+SA. In around 93% of the runs performed, this algorithm obtains all solutions in the front belonging to the best class of merit (Table 4.8). This value is equal to 60% of the runs for the HESA algorithm. Concerning this aspect, the EvABOR-III algorithm is the one with worst performance. A significant difference between EvABOR-III and the other two algorithms is that EvABOR-III is the only algorithm in which some runs present solutions belonging to class 2 and class 3. In the other two algorithms if, at least, one solution belonging to the best class of merit is achieved, the local search is able to obtain the remaining solutions to complete the population with solutions in the same class of merit. Regarding the number of solutions evaluated, the GRASP+SA algorithm is again the algorithm with the best performance when all runs are considered and also for runs with solutions in the best class of merit only. The GRASP+SA performs worse than the other two algorithms regarding the execution time in the runs when solutions in the best class of merit only are achieved. However, this increase (around 10%, in average) is insignificant when comparing with EvABOR-III. In Table 4.8 a summary of values related to these aspects are displayed and Table B.1, Table B.2 and Table B.5 in Appendix B present this information obtained from the 30 runs performed. Statistical data representation for the number of solutions evaluated and the execution time for the three algorithms are also presented in Figure B.1 to Figure B.4 in Appendix B. If the previous analysis is restricted to EvABOR-III and HESA, it is possible to conclude that, in all aspects considered before, the HESA algorithm has better performance than EvABOR-III.

		EvABOR-III	**HESA**	**GRASP+SA**
Classes in the non-dominated front		2 and 3	2 and 3	2 and 3
% of runs with solutions belonging to class 3 only in the non-dominated front		46.(6)%	60%	93.33%
% of runs with solutions belonging to class 2 only in the non-dominated front		26.(6)%	40%	6.67%
% of runs with solutions belonging to classes 2 and 3 in the non-dominated front		26.(6)%	0%	0%
Number of solutions evaluated in all runs	Minimum	13651	6747	17826
	Maximum	120001	120141	91373
	Average	90296	72757.6	39024.83
	Median	120001	70723.5	29365
	Standard Deviation	40114.58	44758.81	18281.30
Number of solutions evaluated in runs with solutions in class 3 only	Minimum	13651	6747	17826
	Maximum	115891	104011	64095
	Average	56347.43	42101.29	35285.68
	Median	41266	37049	29179.5
	Standard Deviation	35878.88	29701.90	12180.26
Execution time of all runs	Minimum	1464	764	3414
	Maximum	13433	13607	17809
	Average	9849.43	8035.2	7520.8
	Median	12777.5	7834.5	5565.5
	Standard Deviation	4365.61	4913.13	3559.78
Execution time of the runs with solutions in class 3 only	Minimum	1464	764	3414
	Maximum	12540	11789	12312
	Average	6167.64	4683.47	6786.18
	Median	4484	4042	5524.5
	Standard Deviation	3922.87	3283.40	2341.39

Table 4.8 – Comparison of the three algorithms.

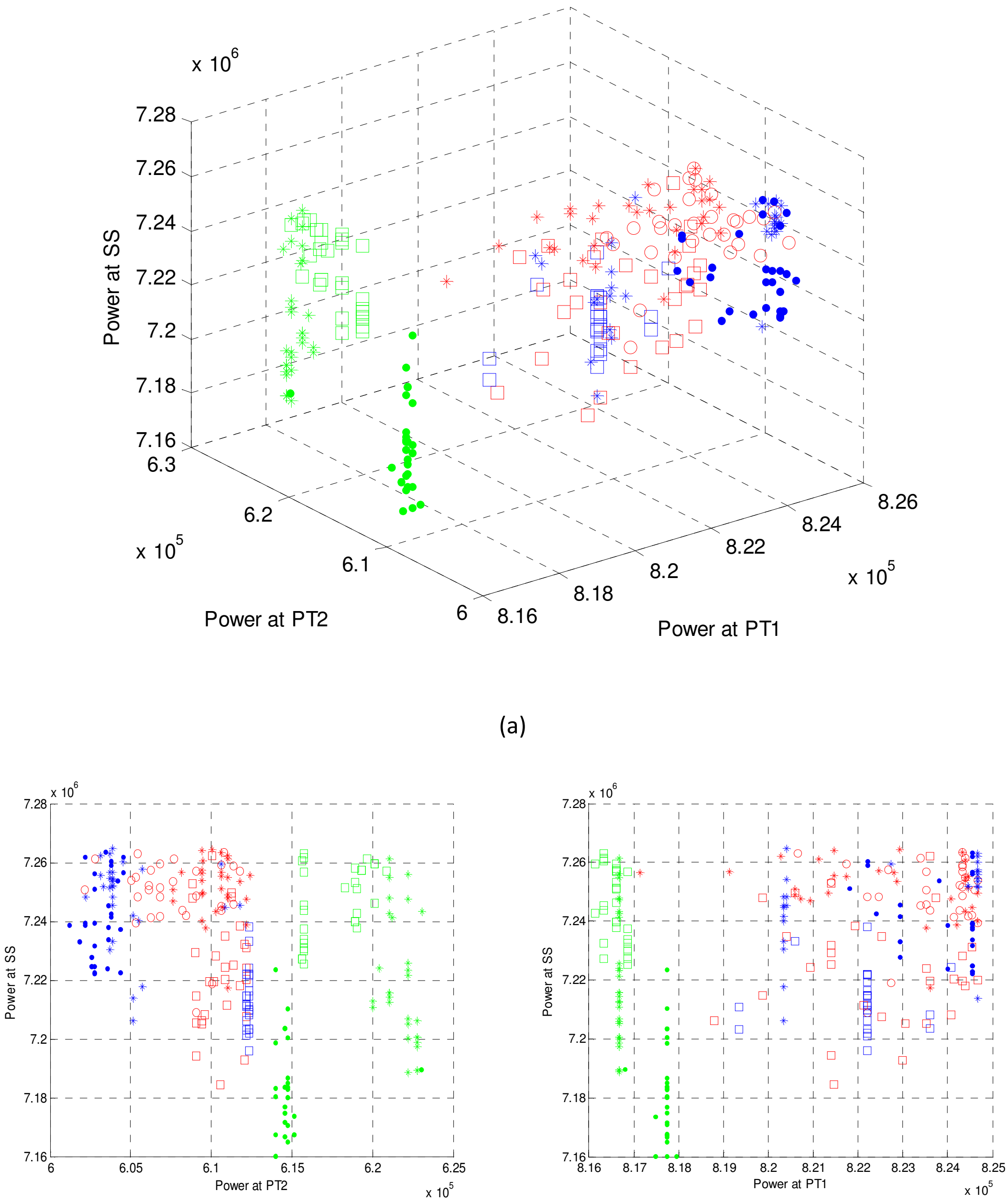
Besides these conclusions, regarding the number of solutions evaluated, the execution time required and the quality of solutions in the front according to the preferences elicited from the DM, it is also important to analyze the diversity of these solutions and the objective function values.

Comparing the objective function values obtained with the three algorithms, an aspect is clear: in most runs GRASP+SA is able to obtain a high number of solutions with inferior maximum power at PT1, and some of them at SS (Figure 4.18), with a much lower number of instants in curtailment (Figure 4.19), without decreasing too much the profits in a majority of solutions (Figure 4.20) and with a decrease in the total time in discomfort when comparing to the solutions obtained with EvABOR-III and HESA (Figure 4.19). Due to the performance in those objective functions, solutions with these characteristics obtained with GRASP+SA are classified in the best class of merit despite the exigency imposed by the cutting-level (λ=0.7) and the higher value of the maximum power at PT2 in comparison with solutions obtained with EvABOR-III and HESA.

The improvement in the reduction of the maximum peak power at PT1 with GRASP+SA is due to the iterative method used in the construction phase, where curtailments are progressively added to or excluded from the control patterns. This method decreases the total number of instants of time in curtailment; consequently the total time in discomfort tends to be inferior and the profits may not decrease too much (depending on the periods of time under control). These aspects lead that GRASP+SA obtains solutions in the best class of merit even if the maximum power at PT2 is higher than in the solutions obtained with EvABOR-III and HESA. Due to the characteristics of PT2, a higher number of curtailments must be applied to the groups (or at least, to some groups) supplied by this distribution power transformer.

From the previous discussion, it is expected that the control strategies obtained with GRASP+SA and the other two algorithms have different characteristics as well as resulting load diagrams. Three examples of control patterns are presented in Figure 4.21 to Figure 4.23 corresponding to two solutions that minimize the maximum peak power at PT2 and a third one that minimizes the maximum peak power at PT1, respectively. The two first control patterns have been obtained with EvABOR-III and HESA and the third with GRASP+SA. Note the similarity between the solutions obtained with EvABOR-III and HESA, with a high number of curtailments applied to the groups. The control strategies presented in Figure 4.21 and Figure 4.22 correspond to the solutions numbered in Figure 4.24 and the corresponding objective functions values are presented in Table 4.9. In Appendix B, Table B.8 to Table B.10

present the remaining objective functions values for all solutions displayed in Figure 4.24. From these tables it is possible to observe that the maximum peak power at any distribution transformer is improved in all solutions obtained with GRASP+SA without decreasing too much the profits. The reduction at PT2 is not so high as with the other algorithms but there are solutions with an improvement of 4.60% and the maximum value of the reduction of the peak at SS is similar to the one obtained with EvABOR-III and HESA.



(a)

(b) (c)

Figure 4.18 – Examples of non-dominated solutions obtained with: EvABOR-III (red marks); HESA (blue marks) and GRASP+SA (green marks).

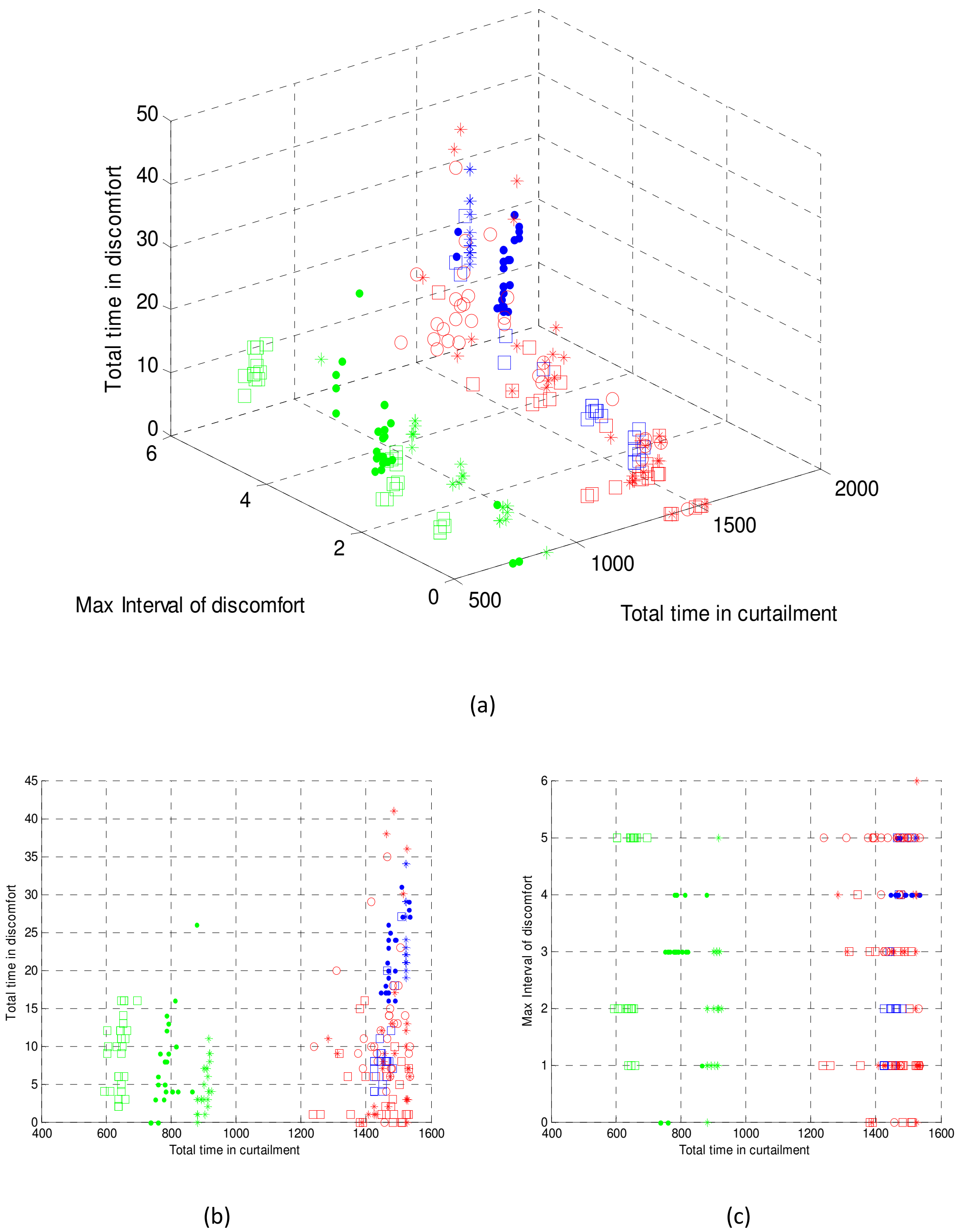


Figure 4.19 – Examples of non-dominated solutions obtained with: EvABOR-III (red marks); HESA (blue marks) and GRASP+SA (green marks).

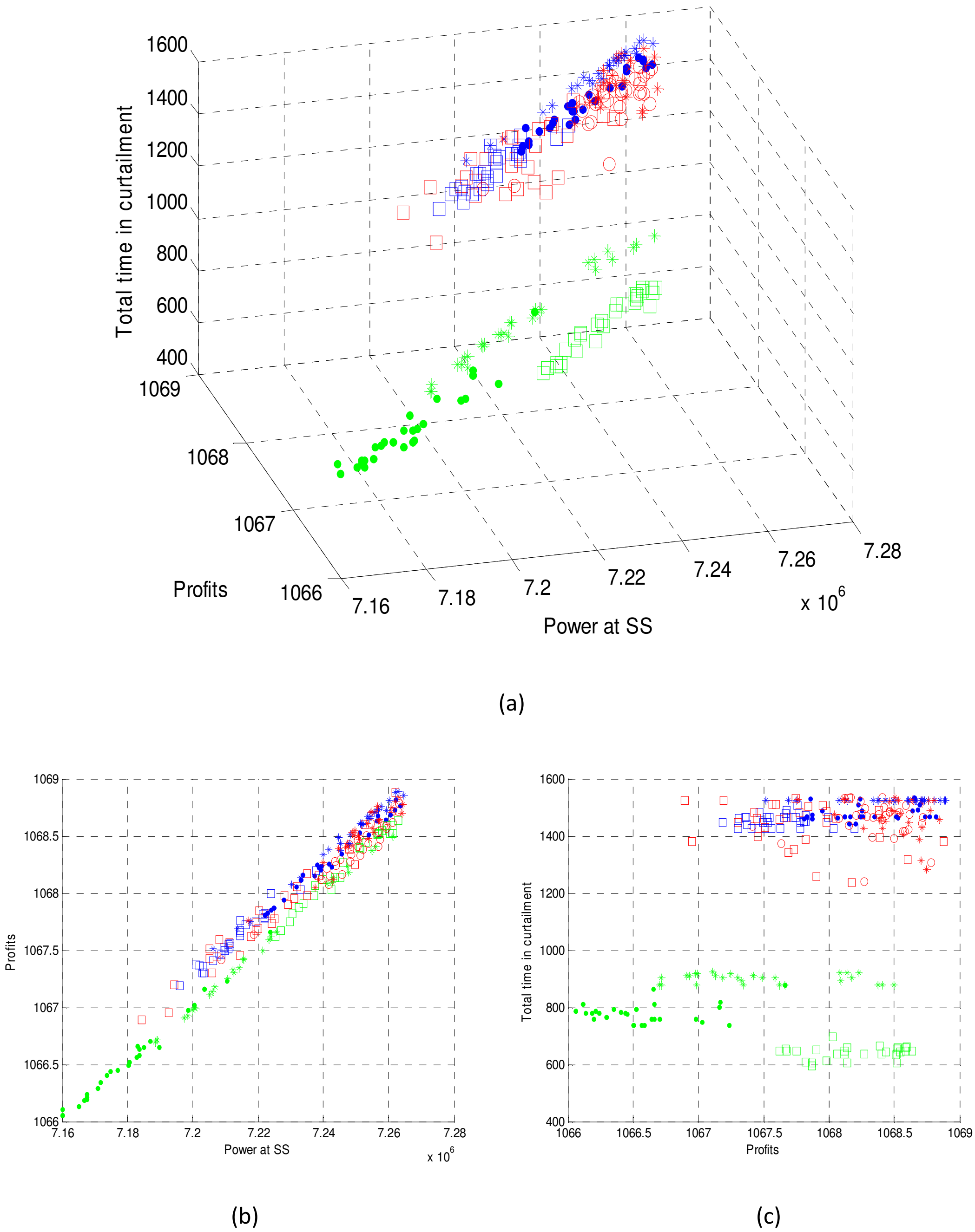


Figure 4.20 – Examples of non-dominated solutions obtained with: EvABOR-III (red marks); HESA (blue marks) and GRASP+SA (green marks).

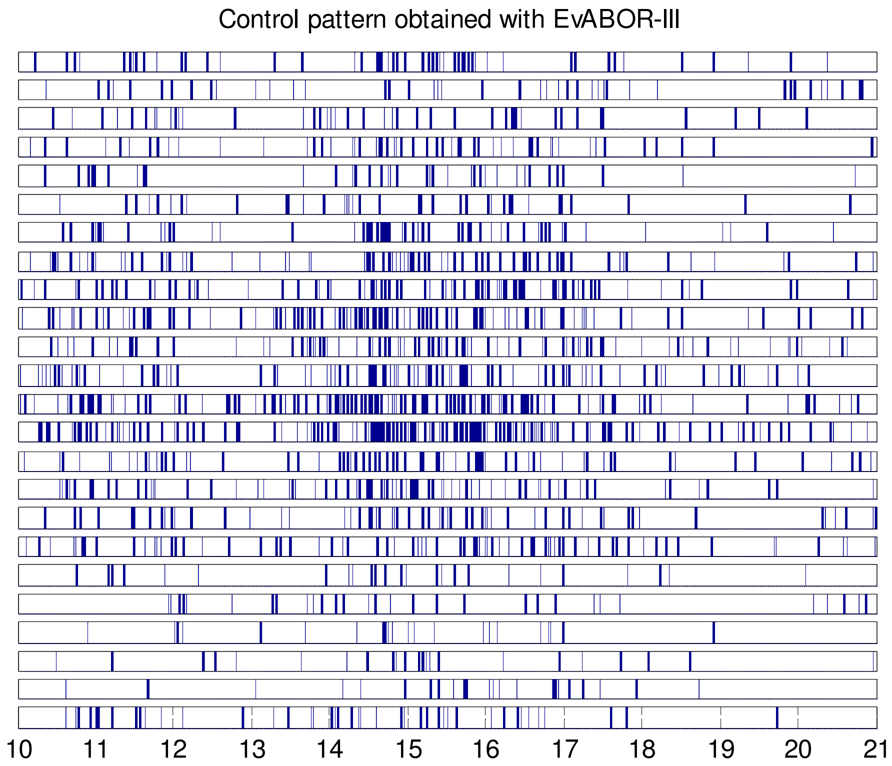


Figure 4.21 – Example of a control strategy obtained with EvABOR-III.

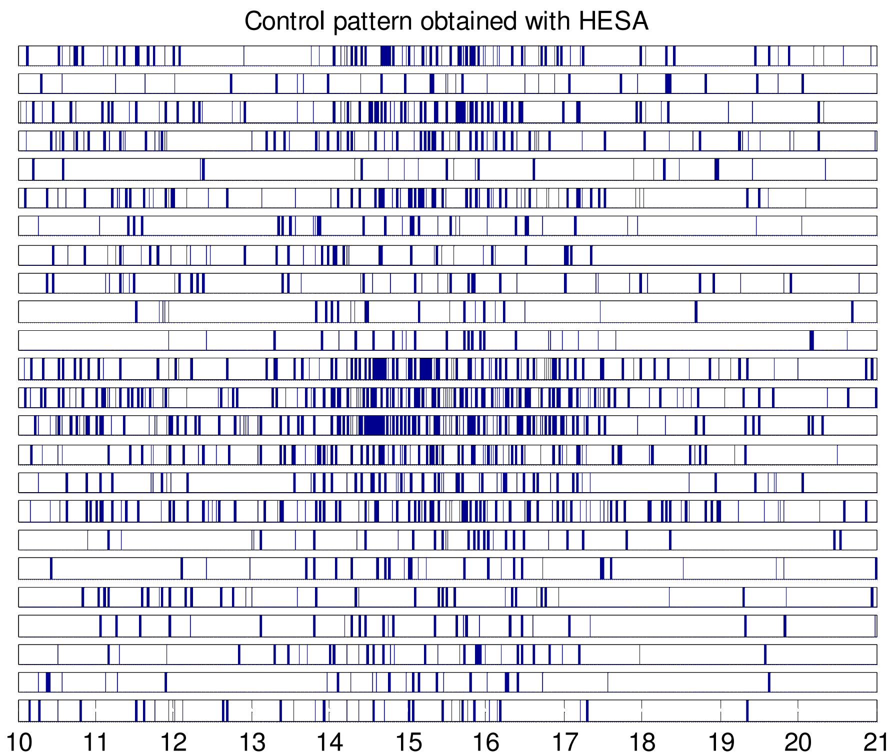


Figure 4.22 – Example of a control strategy obtained with HESA.

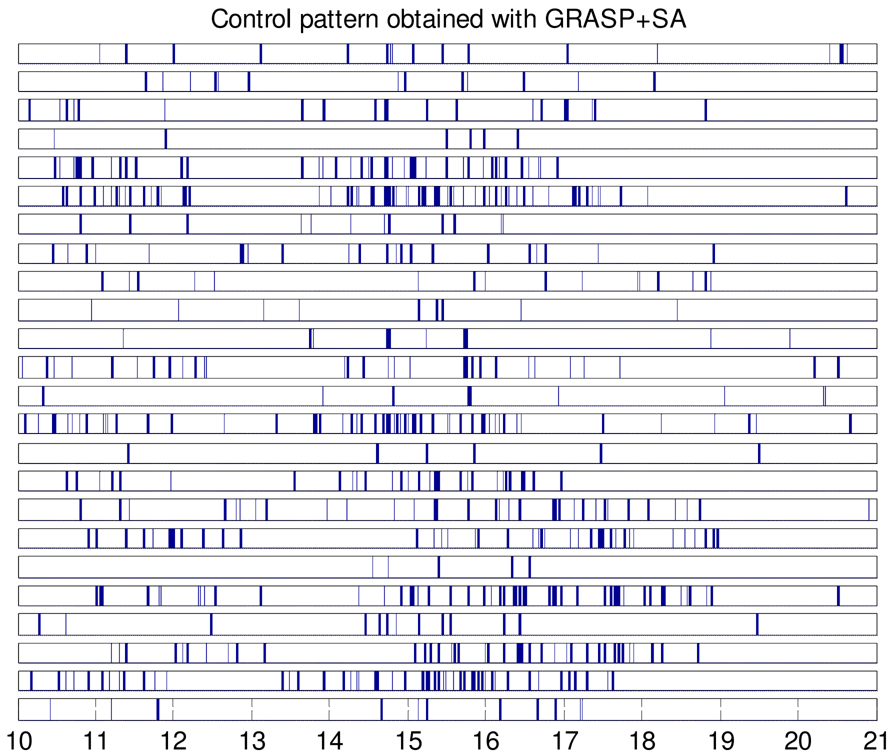


Figure 4.23 – Example of a control strategy obtained with GRASP+SA.

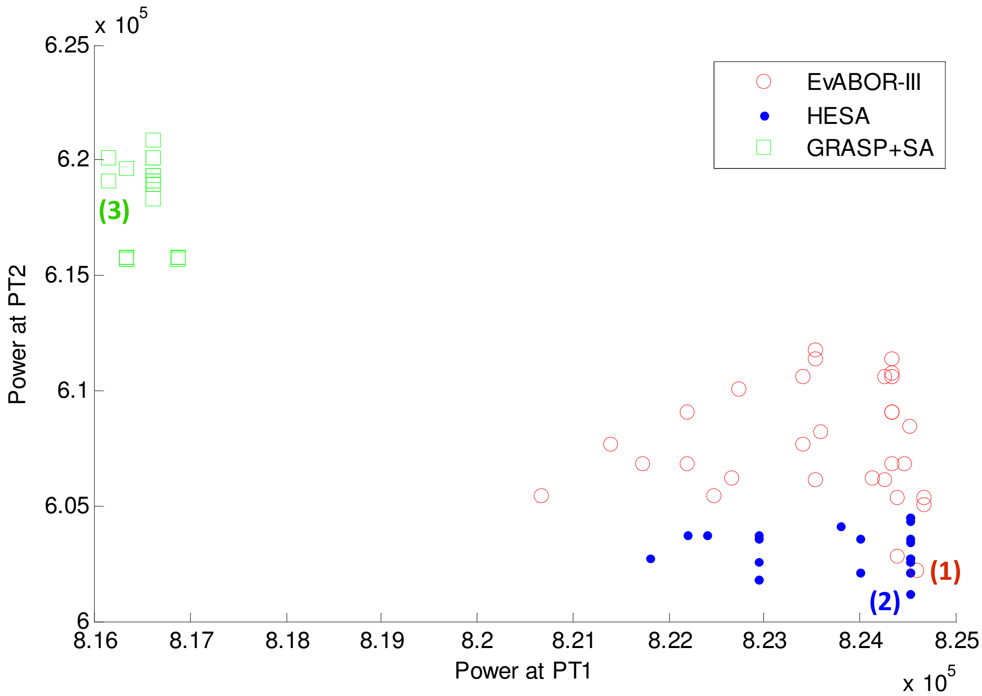


Figure 4.24 – Examples of solutions minimizing the maximum peak power at PT1 (GRASP+SA) and PT2 (EvABOR-III and HESA).

	EvABOR-III		HESA		GRASP+SA			
	Solution (1) **in** Figure 4.24		**Solution (2)** **In** Figure 4.24		**Solution (3)** **in** Figure 4.24		**Solution (4) in** Figure B.10 (b) (Appendix B)	
Maximum power at PT1 (W)	824600	-0.12%	824533.3	-0.11%	**816133.3**	0.91%	816666.7	0.84%
Maximum power at PT2 (W)	**602177.8**	**6.69%**	**601200**	**6.84%**	619066.7	4.07%	619155.6	4.06%
Maximum power at SS (W)	7250539	1.94%	7238939	2.10%	7242673	2.05%	**7133525**	**3.52%**
No. minutes in curtailment	1534	-	1469	-	697	-	951	-
Profits (Euros)	1068.407	0.31%	1068.234	0.33%	1068.026	0.34%	1065.523	0.58%
Maximum interval in discomfort	1	-	4	-	5	-	4	-
Total time in discomfort	6	-	26	-	16	-	16	-

Table 4.9 – Examples of objective function values obtained with the three algorithms.

The load diagrams at PT1 and PT2 (Figure 4.25 and Figure 4.26) show the reduction in peak power demand at these distribution transformers when loads are under control patterns displayed in Figure 4.23 and Figure 4.22. The final load diagram at PT1 displayed in Figure 4.25 corresponds to solution (3) in Figure 4.24 and, the final load diagram at PT2 displayed in Figure 4.26 corresponds to solution (2) in Figure 4.24. It is important to emphasize the length of peak at PT1 (it lasts about 4 hours) making harder the efforts to reduce the peak demand in this distribution transformer. GRASP+SA performs this task easier than the other two algorithms and, in addition, the minimum value of the maximum peak power at SS is also found by this algorithm in the experiments done. The load diagram at SS corresponding to this case is displayed in Figure 4.27 and the control actions to be applied are displayed in Figure 4.28. This is the solution (4) in Figure B.10 (b) in Appendix B, and the remaining objective functions values are presented in the 4th column of Table 4.9. Note that the reduction at PT2 is around 4% without increasing the peak at PT1, contrary to what happens with solutions obtained with EvABOR-III and HESA, with a higher improvement at PT2 but with a degradation of the maximum power at PT1 and a smaller improvement at SS. In Table 4.9 the percentage of improvement/degradation in each objective function is also presented.

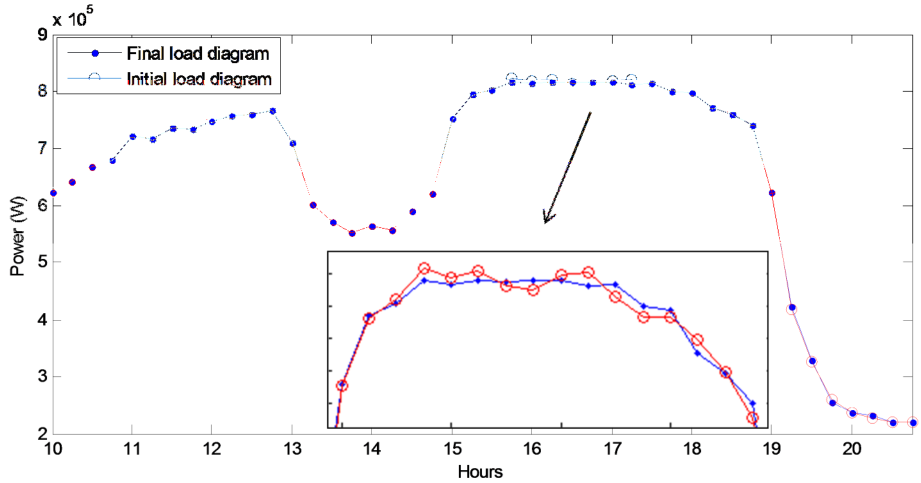


Figure 4.25 – Load diagrams at PT1.

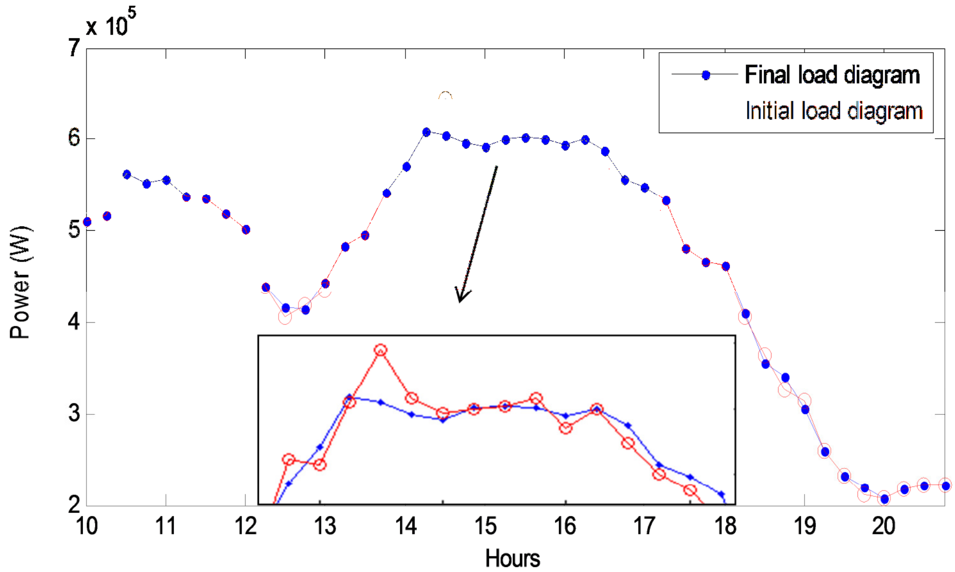


Figure 4.26 – Load diagrams at PT2.

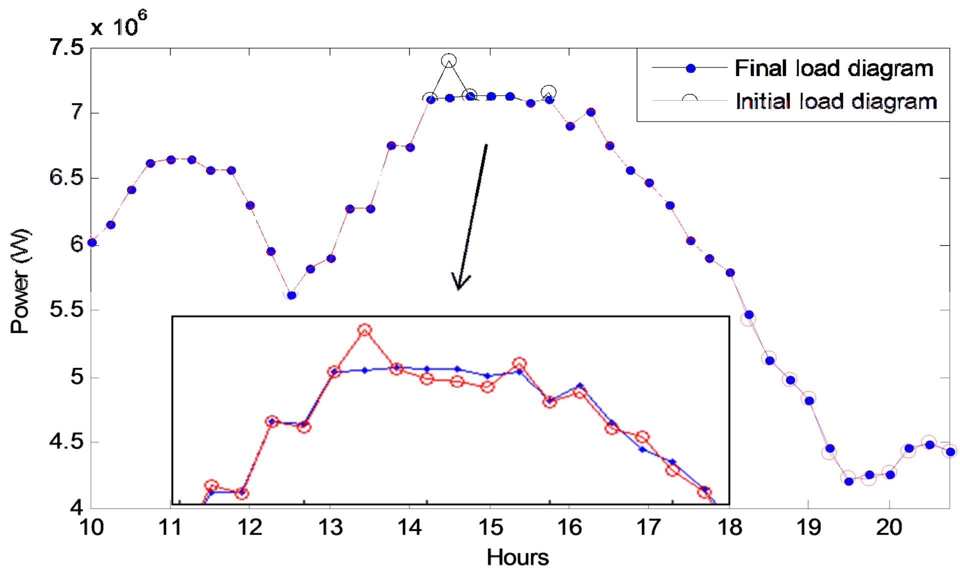


Figure 4.27 – Load diagrams at SS.

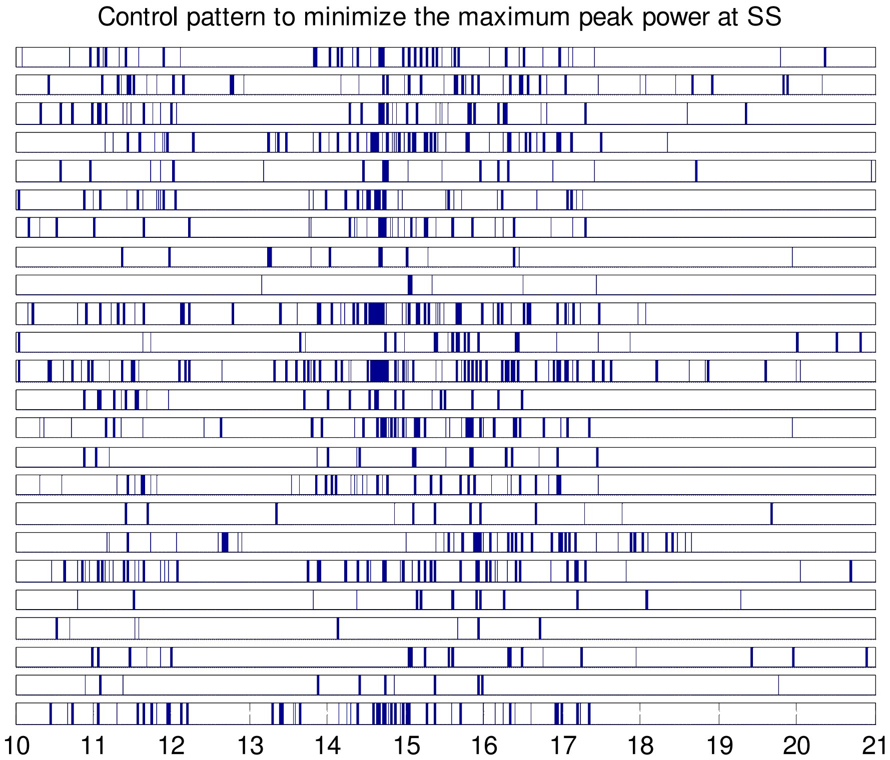


Figure 4.28 – Example of a control strategy to minimize the maximum peak power at SS.

Comparing solutions belonging to the best class of merit obtained in distinct runs of GRASP+SA, it is possible to conclude that these solutions are concentrated in different regions of the search space. This effect is a consequence of the local search around solutions belonging to the best class of merit, in general only one or two, obtained in the construction phase. The same effect appears in some runs performed with HESA, but this effect is not so visible as in GRASP+SA. In EvABOR-III the results from different runs are more consistent regarding the regions where solutions are located. Figure 4.18 to Figure 4.20 illustrate this aspect by displaying solutions obtained with three runs of each algorithm. In Appendix B, solutions obtained in 10 runs of each algorithm are displayed in Figure B.5 to Figure B.10, which illustrate better the situation described above.

With respect to the other approaches, GRASP+SA guarantees solutions in the best class of merit or at least more in accordance with the preferences elicited from the DM in a more consistent way, the total time loads under control is inferior and the best values for the minimization of the peak at PT1 is achieved.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

In this work a novel approach for incorporation of preferences into a multi-objective evolutionary algorithm has been developed using the ELECTRE TRI method to sort the solutions obtained during the evolutionary process into classes of merit. This is carried out based on a set of preferences elicited from a DM by combining the non-dominance and the outranking relations. This mechanism enriches the non-dominance relation and allows focusing the search on more significant regions with respect to preference satisfaction, thus reducing both the computational effort and the cognitive burden on the DM.

Three versions of the Evolutionary Algorithm Based on an Outranking Relation (EvABOR) have been developed, which have been illustrated and compared using a reactive power compensation problem. The computational experiments allow concluding about the superiority of EvABOR-III, showing that the outranking relation is more effective to improve the search when applied to non-dominated solutions only, i.e. after the non-dominance relation intervenes.

The results obtained show that the incorporation of preferences into an EA effectively reduces the computational effort, as it has been recognized by other authors [Branke and Deb (2004), Fernández et al. (2010), Deb et al. (2010)]. Since performing just the characterization of the non-dominated solution set does not provide the required information to support decision making in real-world problems, the choice of a solution to be implemented is facilitated by introducing the DM's preferences in a meaningful manner into the search process in such a way that the final set of non-dominated solutions presented to the DM is in accordance with the preferences elicited. The choice of the mutation operators in the EA is adapted to the solution characteristics taking into account the preferences elicited from the DM. This option proved to be more effective than letting operators be completely random.

The incorporation of a procedure based on SA into EvABOR-III confirms the advantage of including a local search phase into an EA. In this procedure, preferences are incorporated into the acceptance function: solutions more in accordance with the expressed preferences have a greater probability to be accepted for further exploitation. The Hybrid Evolutionary Simulated Annealing (HESA) algorithm has proven to be more effective in founding solutions in best classes of merit than EvABOR-III for the same set of preferences elicited from the DM. HESA overcomes the difficulty presented by EvABOR-III, in some runs, to complete the population with solutions belonging to the best class of merit after the achievement of one solution in this class. Moreover, in HESA the number of solutions belonging to the best class of merit is higher while the execution time is lower. This aspect is particularly important in dealing with real-world MOOPs in which solutions in a due time may be required.

Concerning the hybridization of GRASP with SA, a noticeable improvement of the quality of the initial solutions using the construction phase has been obtained. The existence of knowledge about the problem, which generally occurs when dealing with real-world problems, should be used to improve the construction of solutions. The use of different RCLs in the construction phase of each solution has revealed to be useful in MOOPs, improving the performance of the objective functions considered in each RCL. GRASP+SA requires an additional effort due to the evaluation of the candidate elements to be included in the RCL, but this is compensated by its effectiveness in obtaining solutions in the best class of merit.

EvABOR-III, HESA and GRASP+SA have been tested using a direct load control problem with seven objective functions. GRASP+SA has proven its superiority to find solutions that minimize the maximum peak power at PT1, PT2 and SS, using an lower number of curtailments, decreasing the total time in discomfort for the consumers without decreasing too much the profits.

Due to the local search phase, HESA and GRASP+SA tend to concentrate the non-dominated solutions belonging to the best class of merit in regions of the search space around the solutions in the best class of merit firstly found by the algorithm. This aspect is more visible in the GRASP+SA algorithm.

Comparing solutions belonging to the best class of merit obtained in distinct runs of GRASP+SA, it is possible to conclude that these solutions are concentrated in different regions of the search space. This effect is a consequence of the local search phase.

In future work already outlined additional approaches will be developed to improve the diversity of solutions belonging to the best class of merit and to obtain a more consistent set of non-dominated solutions in fronts obtained in distinct runs of GRASP+SA. The incorporation of preferences also in the construction phase should be also assessed to be included in new versions of GRASP+SA.

Further work will involve exploring the incorporation of preferences into other metaheuristics, by using meaningful parameters that could be easily expressed by the DM. Mechanisms of learning could also be explored associated with preference expression in dynamic environments. The results obtained in this work should in the near future be extended in other directions, including the assessment of solution robustness both taking into account the variation of model parameters and coefficients but also different forms of uncertainty in preference elicitation.

Another research avenue consists in dealing with mechanisms for solution diversity control and the adaptive nature of control parameters in hybrid metaheuristics whenever the DM's preferences play a role in the search process.

BIBLIOGRAPHY

Abdelsalam H, Mohamed AM. Multi-objective Simulated Annealing Algorithm for Partner Selection in Virtual Enterprises. In: Yang Xin-She (Ed.), Artificial Intelligence, Evolutionary Computing and Metaheuristics, Studies in Computational Intelligence, Springer Berlin Heidelberg; 2013, Vol. 427, 751–774.

Alpay Ş. GRASP with path relinking for a multiple objective sequencing problem for a mixed-model assembly line. International Journal of Production Research, 2009; 47(21), 6001–6017.

Anghinolfi D, Paolucci M. Simulated Annealing as an Intensification Component in Hybrid Population-Based Metaheuristics. Simulated Annealing. Vienna, Austria, 2008: I-Tech Education and Publishing.

Antunes CH, Barrico C, Gomes A, Pires DF, Martins AG. An evolutionary algorithm for reactive power compensation in radial distribution networks. Applied Energy, 2009; 86: 977–984.

Antunes CH, Oliveira E, Lima P. A Multi-Objective GRASP Procedure for Reactive Power Compensation Planning. Optimization and Engineering (submitted).

Antunes CH, Lima P, Oliveira E, Pires DF. A Multi-Objective Simulated Annealing Approach to Reactive Power Compensation. Engineering Optimization, 2011; 43(10), 1063-1077.

Arroyo JEC, Armentano VA. Genetic local search for multi-objective flowshop scheduling problems. European Journal of Operational Research, 2005; 167(3): 717-738.

Arroyo J, Pereira A. A GRASP heuristic for the multi-objective permutation flowshop scheduling problem. International Journal of Advanced Manufacturing Technology, 2011; 1: 1-13.

Arroyo J, Vieira P, Vianna DS. A GRASP algorithm for the multi-criteria minimum spanning tree problem. Annals of Operations Research, 2008; 159: 125–133.

Bandyopadhyay S, Saha S, Maulik U, Deb K. A Simulated Annealing-based Multiobjective Optimization Algorithm: AMOSA. IEEE Transactions on Evolutionary Computation, 2008; 12(3): 269–283.

Baños R, Gil C, Paechter B, Ortega J. A hybrid meta-heuristic for multi-objective optimization: MOSATS. Journal of Mathematical Modelling and Algorithms, 2007; 6(2): 213-230.

Bhatnagar R, Rahman S. Dispatch of direct load control for fuel cost minimization. IEEE Transactions on Power Systems, 1986; PWRS-1: 96–102.

Binato S, Faria Jr. H, Resende MGC. Greedy randomized adaptive path relinking. In: Sousa JP (Eds.), Proceedings of the IV Metaheuristics International Conference (MIC2001) 2001, 393–397.

Binato S, Hery WJ, Loewenstern D, Resende MGC. A greedy randomized adaptive search procedure for job shop scheduling. In: Hansen P and Ribeiro CC (Eds.), Essays and surveys on metaheuristics. Kluwer Academic Publishers, 2001.

Blum C, Roli A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys 2003; 35(3): 268-308.

Blum C, Roli A, Alba E. An introduction to metaheuristic techniques. In: Parallel Metaheuristics, a New Class of Algorithms. John Wiley; 2005, 3–42.

Blum C, Puchinger J, Raidl GR, Roli A. A brief survey on hybrid metaheuristics. In: B. Filipic, J. Silc (Eds.), Proceedings of BIOMA 2010 – 4th International Conference on Bio-Inspired Optimization Methods and their Applications, Jozef Stefan Institute, Ljubljana, Slovenia; 2010, 3–18.

Blum C, Puchinger J, Raidl GR, Roli A. Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing, 2011; 11(6): 4135–4151.

Branke J. Consideration of partial user preferences in evolutionary multi-objective optimization. In: Branke J, Deb K, Miettinen K, Slowinski R, editors. Multiobjective Optimization - Interactive and Evolutionary Approaches, LNCS 5252, Springer-Verlag; 2008, 157-177.

Branke J, Deb K. Integrating user preferences into evolutionary multi-objective optimization. In: Jin Y, editors. Knowledge Incorporation in Evolutionary Computation, Springer; 2004, 461-477.

Branke J, Greco S, Slowinski R, Zielniewicz P. Interactive evolutionary multiobjective optimization driven by robust ordinal regression. Bulletin of the Polish Academy of Sciences, Technical Sciences, 2010; 58 (3): 347–358.

Branke J, Kauβlera T, Schmeck H. Guidance in Evolutionary Multi-Objective Optimization. Advances in Engineering Software, 2001; 32: 499–507.

Brans JP, Mareschal B. How to select and how to rank projects: the PROMETHEE method for MCDM. European Journal on Operation Research, 1986; 24: 228–238.

Bresina JL. Heuristic-biased stochastic sampling. In: Proceedings of AAAI'96 Proceedings of the thirteenth national conference on Artificial Intelligence, 1996; 271–278.

Buer T, Pankratz G. Grasp with hybrid path relinking for bi-objective winner determination in combinatorial transportation auctions. BuR-Business Research, 2009; 3(2), 192–213.

Burke EK, Cowling P, Landa Silva JD, Petrovic S. Combining hybrid metaheuristics and populations for the multiobjective optimisation of space allocation problems. In Proceedings of the 2001 Genetic and Evolutionary Computation Conference (GECCO 2001), 1252-1259.

Cakir B, Altiparmak F, Dengiz B. Multi-objective optimization of a stochastic assembly line balancing: A hybrid simulated annealing algorithm. Computers and Industrial Engineering, 2011; 60(3): 376-384.

Caponio A, Neri F. Integrating cross-dominance adaptation in multi-objective memetic algorithms. Springer Berlin Heidelberg; 2009, 325-351.

Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. Journal of Optimization Theory and Applications, 1985; 45: 41–51.

Chang T-J, Meade N, Beasley JE, Sharaiha YM. Heuristics for cardinality constrained portfolio optimization. Computers and Operations Research, 2000; 27: 1271–1302.

Chen X, Ong YS, Lim MH, Tan KC. A multi-facet survey on memetic computation. IEEE Transactions on Evolutionary Computation, 2011; 15(5): 591-607.

Chaudhuri S, Deb K. An interactive multiobjective optimization and decision-making using evolutionary methods. Applied Soft Computing, 2010; 10, 496-511.

Chica M, Cordón O, Damas S, Bautista J. Multiobjective constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. Information Sciences, 2010; 180: 3465–3487.

Coello CAC. An updated survey of evolutionary multiobjective optimization techniques: State of the art and future trends. In Evolutionary Computation, 1999. CEC 99. IEEE Proceedings of the, 1999, Vol. 1.

Coello CAC. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In: Congress on Evolutionary Computation. Piscataway, New Jersey: IEEE Service Center, 2000; 30-37.

Coello CAC, Veldhuizen DV, Lamont GB. Evolutionary Algorithms for Solving Multiobjective Problems. New York: Kluwer Academic Publishers, 2002.

Coelho RF, Bersini H, Bouillard Ph. Parametrical mechanical Design with Constraints and Preferences: Application to a Purge Valve. Computational Methods Appl. Mechanical Engineering, 2003; 192: 4355-4378.

Corana A, Marchesi M, Martini C, Ridella S. Minimizing Multimodal Functions of Continuous Variables with the Simulated Annealing Algorithm, ACM Transactions on Mathematical Software March; 1987, 272–280.

Corne D, Knowles J. Techniques for Highly Multiobjective Optimisation: Some Nondominated Points are Better than Others. In: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO), New York: ACM Press; 2007, 773-780.

Cotta C. A study of hybridisation techniques and their application to the design of evolutionary algorithms. AI Communications, 1998; 11(3–4): 223–224.

Cotta C, Talbi EG, Alba E. Parallel hybrid metaheuristics. In: Alba E (Ed.), Parallel Metaheuristics, a New Class of Algorithms. John Wiley; 2005, 347–370.

Cvetkovic D, Parmee IC. Preferences and Their Application in Evolutionary Multiobjective Optimization. IEEE Transactions on Evolutionary Computation, 2002; 6 (1): 42-57.

Czyzak P, Jaszkiewicz A. Pareto simulated annealing. In: Fandel G, Gal T (Ed.), Multiple Criteria Decision Making, Lecture Notes in Economics and Mathematical Systems, 1997; 448: 297–307.

Czyzak P, Jaszkiewicz A. Pareto simulated annealing—A metaheuristic technique for multiple-objective combinatorial optimization. Journal of Multi-Criteria Decision Analysis, 1998; 7: 34–47.

Das I, Dennis J. A closer look at drawbacks of minimizing weighted sums of objectives for Pareto set generation in multicriteria optimization problems. Structural Optimization, 1997; 14(1): 63–69.

Deb K. Solving goal programming problems using multi-objective genetic algorithms. In: Proceedings of IEEE Congress on Evolutionary Computation (CEC 99); 1999, 77-84.

Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. Chichester: Wiley; 2001.

Deb K, Pratap A, Agrawal S, Meyarivan T. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 2002; 6(2): 182–197.

Deb K, Sinha A, Korhonen PJ, Wallenius J. An Interactive Evolutionary Multiobjective Optimization Method Based on Progressively Approximated Value Functions. IEEE Transactions on Evolutionary Computation, 2010; 14(5): 723 – 739.

Delfanti M, Granelli G, Marannino P, Montagna M. Optimal capacitor placement using deterministic and genetic algorithms. IEEE Transactions on Power Systems, 2000; 15 (3): 1041–1046.

di Pierro F, Soon-Thiam K, Savic DA. An Investigation on Preference Order Ranking Scheme for Multiobjective Evolutionary Optimization. IEEE Transactions on Evolutionary Computation, 2007; 11(1): 17-45.

Dréo J, Petrowski A, Siarry P, Taillard E. Metaheuristics for Hard Optimization, Springer, Germany, 2006.

Drexl A, Nikulin Y. Multicriteria airport gate assignment and Pareto simulated annealing. IIE Transactions (Institute of Industrial Engineers), 2008; 40(4):385–397.

El-Abd M, Kamel M. A taxonomy of cooperative search algorithms. In: Blesa MJ, Blum C, Roli A, Sampels M (Eds.), Hybrid Metaheuristics: Second International Workshop. Volume 3636 of LNCS., Springer; 2005, 32–41.

Farina M, Amato P. On the optimal solution definition for many-criteria optimization problems. In: Keller J, Nasraoui O, editors. Proc. of the NAFIPS-FLINT Int’l Conf. 2002. Piscataway: IEEE Press; 2002, 233–238.

Feo T, Resende M. A probabilistic heuristic for a computationally difficult set covering problem. Operations Research Letters, 1989; 8(2): 67-71.

Feo T, Resende M. Greedy randomized adaptive search procedures. Journal of Global Optimization, 1995; 6: 109-134.

Fernández E, López E, Bernal S, Coello C A C, Navarro J. Evolutionary multiobjective optimization using an outranking-based dominance generalization. Computers and Operations Research, 2010; 37 (2): 390-395.

Festa P, Resende M. An annotated bibliography of GRASP–Part I: Algorithms. International Transactions in Operational Research, 2009a; 16(1), 1–24.

Festa P, Resende M. An annotated bibliography of GRASP–Part II: Applications. International Transactions in Operational Research, 2009b; 16(2), 131–172.

Festa P, Resende MG. Hybrid GRASP heuristics. In Foundations of Computational Intelligence. Springer Berlin Heidelberg; 2009c; 3: 75-100.

Festa P, Resende M. Effective application of GRASP. In: Wiley Encyclopedia of Operations Research and Management Science. Cochran JJ, Cox Jr. LA, Keskinocak P, Kharoufeh JP, Smith JC (Eds.), John Wiley & Sons, 2011; 3, 1609-1617.

Festa P, Resende MG. Hybridizations of GRASP with path-relinking. In Hybrid Metaheuristics. Springer Berlin Heidelberg, 2013; 135-155.

Fonteix C, Massebeuf S, Pla F, Kiss L N. Multicriteria optimization of an emulsion polymerization process. European Journal of Operational Research, 2004; 153: 350–359.

Fonseca CM, Fleming PJ. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Stephanie Forrest (Eds.), Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, San Mateo, CA: Morgan Kaufman, 1993; 416-423.

Fonseca CM, Fleming PJ. An overview of evolutionary algorithms in multiobjective optimization. Evolutionary Computation, 1995; 3(1): 1-16.

Garza-Fabre M, Pulido GT, Coello CAC. Ranking Methods for Many-Objective Optimization. In: Proceedings of the 8th Mexican International Conference on Artificial Intelligence, MICAI'09. Heidelberg, Berlin: Springer-Verlag, 2009; 633-645.

Gooi HB, Mendes DP , Bell KRW, Kirschen DS. Optimal scheduling of spinning reserve. IEEE Transactions on Power Systems, 1999; 14 (4), 1485-1492.

Gomes A, Henggeler Antunes C, Gomes Martins A. A multiple objective evolutionary approach for the design and selection of load control strategies. IEEE Transactions on Power Systems, 2004; 19 (2), 1173-1180.

Gomes A, Henggeler Antunes C, Gomes Martins A. A multiple objective approach to electric load management using an interactive evolutionary algorithm. IEEE Transactions on Power Systems, 2007; 22 (3), 1004-1011.

Gomes A, Henggeler Antunes C, Gomes Martins A. Design of an adaptive mutation operator in an electrical load management case study. Computers and Operations Research, 2008; 35(9), 2925–2936.

Gomes A, Henggeler Antunes C, Oliveira E. Direct load control in the perspective of an electricity retailer – a multi-objective evolutionary approach. In: Gaspar Cunha A, Takahashi

R, Schaefer G, Costa L (Eds.), Soft Computing in Industrial Application, Advances in Intelligent and Soft Computing, 96, Springer; 2011, 13-26.

Gomes A, Henggeler Antunes C, Martinho J, Oliveira E. Otimização multiobjetivo com algoritmos evolutivos - uma aplicação no sector elétrico. In: XVI Latin-Ibero-American Conference on Operations Research and XLIV Brazilian Symposium on Operations Research, 2012.

Glover F. Future paths for integer programming and links to artificial intelligence. Decision Sciences, 1977; 8, 156–166.

Glover F, Kochenberger G. (Eds.) Handbook of Metaheuristics, volume 57 of International Series in Operations Research & Management Science. Kluwer Academic Publishers, 2003.

Hanoun S, Nahavandi S. A greedy heuristic and simulated annealing approach for a bicriteria flowshop scheduling problem with precedence constraints—a practical manufacturing case. The International Journal of Advanced Manufacturing Technology 2012; 60(9-12): 1087-1098.

Hansen MP. Generating a Diversity of Good Solutions to a Practical Combinatorial Problem using Vectorized Simulated Annealing. Technical report, Institute of Mathematical Modelling, Technical University of Denmark, August 1997. Working Paper.

Hart JP, Shogan AW. Semi-greedy heuristics: An empirical study. Operations Research Letters, 1987; 6:107–114.

Hart WE, Krasnogor N, Smith JE. Memetic evolutionary algorithms. In Recent advances in memetic algorithms. Springer Berlin Heidelberg; 2005; 3-27.

Higgins AJ, Hajkowicz S, Bui E. A multi-objective model for environmental investment decision making. Computers and Operations Research, 2008; 35: 253–266.

Hui S. Multi-objective optimization for hydraulic hybrid vehicle based on adaptive simulated annealing genetic algorithm. Engineering Applications of Artificial Intelligence, 2010. 23(1): 27-33.

Iba K, Suzuki H, Suzuki KI, Suzuki K. Practical reactive power allocation/operation planning using successive linear programming. IEEE Transactions on Power Systems, 1988; 3 (2): 558–566.

Ishibuchi H, Murata T. Multi-objective genetic local search algorithm. Proceedings of IEEE International Conference on in Evolutionary Computation, 1996, 119-124.

Ishibuchi H, Murata T. A multi-objective genetic local search algorithm and its application to flowshop scheduling. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 28(3), 392-403. 1998.

Ishibuchi H, Yamamoto T. Fuzzy rule selection by multi-objective genetic local search algorithms and rule evaluation measures in data mining. Fuzzy Sets and Systems, 2004; 141(1): 59-88.

Ishibuchi H, Yoshida T, Murata T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. IEEE Transactions on Evolutionary Computation, 2003; 7(2): 204-223.

Ishibuchi H, Hitotsuyanagi Y, Wakamatsu Y, Nojima Y. How to choose solutions for local search in multiobjective combinatorial memetic algorithms. In Parallel Problem Solving from Nature, PPSN XI. Springer Berlin Heidelberg; 2010, 516-525.

Ishida C , Pozo A, Goldbarg E, Goldbarg M. Multiobjective optimization and rule learning: Subselection algorithm or meta-heuristic algorithm?. In: N. Nedjah, L.M.Mourelle, and J. Kacprzyk (Eds.), Innovative Applications in Data Mining, Studies in Computational Intelligence, Springer, 2009; 169, 47–70.

Jaszkiewicz A. A metaheuristic approach to multiple objective nurse scheduling. Foundations of Computing and Decision Sciences, 1997; 22(3): 169–184.

Jaszkiewicz A. Genetic local search for multi-objective combinatorial optimization. European journal of operational research, 2002; 137(1): 50-71.

Jaszkiewicz A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-A comparative experiment. IEEE Transactions on Evolutionary Computation, 2002a; 6(4): 402-412.

Jaszkiewicz A. A comparative study of multiple-objective metaheuristics on the bi-objective set covering problem and the Pareto memetic algorithm. Annals of Operations Research, 2004; 131(1-4): 135-158.

Jaszkiewicz A, Branke J. Interactive Multiobjective Evolutionary Algorithms. In: Branke J, Deb K, Miettinen K, Slowinski R, editors. Multiobjective Optimization. Interactive and Evolutionary Approaches. Berlin, Germany: Springer; Lecture Notes in Computer Science 5252; 2008; 179-193.

Jaszkiewicz A, Zielniewicz P. Pareto memetic algorithm with path relinking for bi-objective traveling salesperson problem. European Journal of Operational Research, 2009; 193(3): 885-890.

Jaszkiewicz A, Ishibuchi H, Zhang Q. Multiobjective memetic algorithms. In Handbook of Memetic Algorithms. Springer Berlin Heidelberg. 2012; 201-217.

Jin Y, Sendhoff B. Incorporation of Fuzzy Preferences into Evolutionary Multiobjective Optimization. In: Proceedings of the 4th Asia Pacific Conference on Simulated Evolution and Learning, 1. Singapore, 2002; 26-30.

Jorge H, Antunes CH, Martins AG. A multiple objective decision support model for the selection of remote load control strategies. IEEE Transactions on Power Systems, 2000; 15(2), 865 - 872.

Kafafy A, Bounekkar A, Bonnevay S. A hybrid evolutionary metaheuristics (HEMH) applied on 0/1 multiobjective knapsack problems. GECCO'11 Proceedings ACM, 2011; 497-504.

Kirkpatrick S, Gelatt C, Vecchi M. Optimization by simulated annealing. Science 1983; 220 (4598): 671–680.

Konak A, Coit DW, Smith AE. Multi-objective optimization using genetic algorithms: A tutorial. Reliability Engineering and System Safety, 2006; 91: 992–1007.

Karangelos E, Bouffard F. Towards Full Integration of Demand-Side Resources in Joint Forward Energy/Reserve Electricity Markets. IEEE Transactions on Power Systems, 2012; 27 (1): 280 - 289.

Knowles J, Corne D. The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization. In: Proceedings of the 1999 Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press, 1999; 98-105.

Knowles J, Corne D. Approximating the non-dominated front using the Pareto archived evolution strategy. Evolutionary Computation, 2000; 8(2): 149-172.

Knowles J, Corne D. Memetic algorithms for multiobjective optimization: issues, methods and prospects. In Recent advances in memetic algorithms. Springer Berlin Heidelberg, 2005;

313-352.

Knowles J, Corne D. Quantifying the effects of objective space dimension in evolutionary multiobjective optimization. In: Proceedings of the Fourth International Conference on Evolutionary Multi-Criterion Optimization (EMO-2007). LNCS 4403, 2007; 757–771.

Krasnogor N, Smith J. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation, 2005; 9(5): 474-488.

Kubotani H, Yoshimura K. Performance evaluation of acceptance probability functions for multi-objective SA. Computers and Operations Research, 2003; 30(3): 427-442.

Kurucz C, Brandt D, Sim S. A linear programming model for reducing system peak through customer load control programs. IEEE Transactions on Power Systems, 1996; 11: 1817–1824.

Lara A, Sanchez G, Coello Coello CA, Schutze O. HCS: A new local search strategy for memetic multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation 2010; 14(1): 112-132.

Li H, Landa-Silva D. An elitist GRASP metaheuristic for the multi-objective quadratic assignment problem. In: M. Ehrgott, C.M. Fonseca, X. Gandibleux, J.-K. Hao, and M. Sevaux (Eds.), Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, Springer; 2009, 5467, 481–494.

Li H, Landa-Silva D. An Adaptive Evolutionary Multi-Objective Approach Based on Simulated Annealing. Evolutionary Computation, 2011; 19(4): 561–595.

Liu D, Tan KC, Goh CK, Ho WK. A multiobjective memetic algorithm based on particle swarm optimization. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 2007; 37(1): 42-50.

Lourenço HR, Paixão JP, Portugal R. Multiobjective metaheuristics for the bus driver scheduling problem. Transportation Science, 2001; 35(3): 331–343.

Lučić P, Teodorović D. Simulated annealing for the multi-objective aircrew rostering problem. Transportation Research Part A, 1999; 33: 19–45.

Martinez-Martin FJ, Gonzalez-Vidosa F, Hospitaler A, Yepes V. Multi-objective optimization design of bridge piers with hybrid heuristic algorithms. Journal of Zhejiang University SCIENCE A, 2012; 13(6): 420-432.

Martí R, Campos V, Resende MGC, Duarte A. Multi-objective GRASP with path-relinking. AT&T Labs Research TR, 2011.

Mauttone A, Urquhart ME. A Multi-Objective Metaheuristic approach for the Transit Network Design Problem. Public Transport, 2009; 1(4), 253–273.

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller A, Teller E. Equations of state calculations by fast computing machines. Journal of Chemical Physics, 1953; 21 (6), 1087–1092.

Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech concurrent computation program, C3P Report, 826, 1989.

Moscato P, Cotta C, Mendes A. Memetic algorithms. In New optimization techniques in engineering (pp. 53-85). Springer Berlin Heidelberg, 2004.

Mockus J, Eddy E, Mockus A, Mockus L, Reklaitis GV. Bayesian discrete and global optimization. Kluwer Academic Publishers, 1997.

Mohsenian-Rad A, Wong VWS, Jatskevich J, Schober R, Leon-Garcia A. Autonomous Demand-Side Management Based on Game-Theoretic Energy Consumption Scheduling for the Future Smart Grid. IEEE Transactions on Smart Grid, 2010; 1 (3): 320-331.

Montgomery J. The No Free Lunch Theorems for Optimisation: An Overview. 2002. Available at: http://www.martinsewell.com/evolutionary-computation/nfl/Mont02.pdf

Mousseau V, Slowinski R, Zielniewicz P. A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support. Computers and Operations Research, 2000; 27: 757–777.

Neri F, Cotta C, Moscato P. (Eds.). Handbook of memetic algorithms (Vol. 379). Springer. 2011.

Neri F, Cotta C. Memetic algorithms and memetic computing optimization: A literature review. Swarm and Evolutionary Computation, 2012; 2: 1-14.

Ng K-H, Sheble GB. Direct load control—a profit-based load management using linear programming. IEEE Transactions on Power Systems, 1998; 13: 688–695.

Nguyen QH, Ong YS, Krasnogor N. A study on the design issues of memetic algorithm. In Evolutionary Computation, 2007. CEC 2007; 2390-2397.

Nguyen ML, Hui SC, Fong AC. Divide-and-conquer memetic algorithm for online multi-objective test paper generation. Memetic Computing, 2012; 4(1): 33-47.

Oliveira E, Antunes CH, An Evolutionary Algorithm based on an outranking relation for sorting problems. In: Proceedings of the IEEE Int. Conf. on Systems Man and Cybernetics; 2010 Oct 10-13; Istanbul, 2010a; 2732–2739.

Oliveira E, Antunes CH. An evolutionary algorithm guided by preferences elicited according to the ELECTRE TRI method principles. In: Cowling P, Merz P, editors. Proceedings of the EvoCOP 2010, LNCS 6022, Springer; 2010b. 214–225.

Oliveira E, Antunes CH, Gomes A. A comparative study of different approaches using an outranking relation in a multi-objective evolutionary algorithm. Computers and Operations Research, 2013; 40 (6): 1602-1615.

Oliveira E, Antunes CH, Gomes A. Incorporation of preferences in an evolutionary algorithm using an outranking relation - the EvABOR approach, International Journal of Natural Computing Research, 2011; 2 (1): 63–85.

Osman IH, Kelly JP. Metaheuristics: Theory and Applications. Kluwer, Boston, 1996.

Osman IH, Laporte G. Metaheuristics: A bibliography. Annals of Operations Research, 1996; 63: 513–623.

Parreiras RO, Maciel JHRD, Vasconcelos JA. The a posteriori decision in multiobjective optimization problems with Smarts, Promethee II and a fuzzy algorithm. IEEE Transactions on Magnetics, 2006; 42(4): 1139-1142.

Parreiras RO, Vasconcelos JA. A multiplicative version of Promethee II applied to multiobjective optimization problems. European Journal of Operational Research, 2007; 183(2): 729-740.

Pires D F, Antunes CH, Martins A G. NSGA-II with local search for a multi-objective reactive power compensation problem. Electrical Power and Energy Systems, 2012; 43(1): 313-324.

Pitsoulis LS, Resende MGC. Greedy randomized adaptive search procedure. In: Pardalos P , Resende M (Eds.) Handbook of Applied Optimization, Oxford University Press, 2002; 168–183.

Prais M, Ribeiro CC. Parameter variation in GRASP implementations. In: Extended Abstracts of the Third Metaheuristics International Conference, Angra dos Reis, 1999; 375–380.

Prais M, Ribeiro CC. Parameter variation in GRASP procedures. Investigación Operativa, 2000; 9: 1–20.

Puchinger J, Raidl GR. Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In: Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation, Part II. Volume 3562 of LNCS., Springer, 2005; 41–53.

Rachmawati L, Srinivasan D. Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. In: IEEE Congress on Evolutionary Computation (CEC'2006), Vancouver, BC, Canada, July 2006; 3385-3391.

Rachmawati L, Srinivasan D. Multiobjective Evolutionary Algorithm with Controllable Focus on the Knees of the Pareto Front. IEEE Trans. Evolutionary Computation, 2009; 13 (4): 810-824.

Raidl GR. A unified view on hybrid metaheuristics. In: Almeida F, Blesa Aguilera MJ, Blum C, Moreno Vega JM, Pérez M, Roli A, Sampels M (Eds.), Proceedings of HM 2006 – Third International Workshop on Hybrid Metaheuristics, Vol. 4030 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, 2006, 1–12.

Rekiek B, De Lit P, Pellichero F, L'Eglise T, Falkenauer E, Delchambre A. Dealing With User's Preferences in Hybrid Assembly Lines Design. In Proceedings of the MCPL'2000 Conference.

Resende MGC. Metaheuristic hybridization with GRASP. In: Chen, Z.L., Raghavan, S. (eds.) Tutorials in Operations Research, INFORMS 2008.

Resende MGC, Mateus GR, Silva RMA. GRASP: Busca gulosa aleatorizada e adaptativa. In: Gaspar-Cunha A, Takahashi R, Antunes CH (Eds.). Manual de computação evolutiva. Imprensa da Universidade de Coimbra, 2012, 203–213.

Resende MGC, Ribeiro CC. Greedy randomized adaptive search procedures. In: F. Glover and G. Kochenberger (Eds.), Handbook of Metaheuristics, 219–249. Kluwer Academic Publishers, 2003.

Resende MGC, Ribeiro CC. Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. M. Gendreau and J.-Y. Potvin (Eds.), Handbook of Metaheuristics, 2nd Edition, Springer 2010, 283-319.

Resende MGC, Werneck RF. A hybrid heuristic for the p-median problem. Journal of Heuristics, 2004; 10, 59–88.

Reynolds AP, de la Iglesia B. A multi-objective GRASP for partial classification. Soft Computing, 2009; 13: 227–243.

Reynolds AP, Corne DW, de la Iglesia B. A multiobjective GRASP for rule selection. In: Proceedings of the 11th Annual Conference on Genetic and Evolutionary Computation, Montreal, Quebec, Canada, ACM, 2009, 643–650.

Roy B. The outranking approach and the foundation of ELECTRE methods. Theory and Decision, 1991; 31: 49–73.

Roy B. Multicriteria Methodology for Decision Aiding. Dordrecht: Springer; 1996.

Ruiz-Torres AJ, Emory Enscore E, Barton RR. Simulated annealing heuristics for the average flow-time and the number of tardy jobs bi-criteria identical parallel machine problem. Computers and industrial engineering, 1997; 33(1): 257-260.

Said L B, Bechikh S, Ghedira K. The r-Dominance: A New Dominance Relation for Interactive Evolutionary Multicriteria Decision Making. IEEE Transactions on Evolutionary Computation 2010; 14 (5): 801-818.

Schaffer JD. Some experiments in machine learning using vector evaluated genetic algorithms. Vanderbilt Univ., Nashville, TN (USA), 1985.

Serafini P. Simulated annealing for multiple objective optimization problems. In: Tzeng GH, Wang HF, Wen VP, Yu PL (Eds.), Multiple Criteria Decision Making. Expand and Enrich the Domains of Thinking and Application, Springer Verlag, 1994, 283-292.

Siarry P, Berthiau G, Durdin F, Haussy J. Enhanced simulated annealing for globally minimizing functions of many-continuous variables. ACM Transactions on Mathematical Software, 1997; 23(2): 209-228.

Singh HK, Ray T, Smith W. C-PSA: Constrained Pareto simulated annealing for constrained multi-objective optimization. Information Sciences, 2010; 180(13): 2499-2513.

Smith KI, Everson RM, Fieldsend JE. Dominance measures for multi-objective simulated annealing. In Evolutionary Computation, 2004. CEC2004. 23-30.

Smith KI, Everson RM, Fieldsend JE, Murphy C, Misra R. Dominance-Based Multiobjective Simulated Annealing. IEEE Transactions on Evolutionary Computation, 2008; 12(3): 323–342.

Steuer RE. Multiple Criteria Optimization: Theory, Computation and Application. Wiley Series in Probability and Mathematical Statistics. John Wiley, New York, 1986.

Suman B, Kumar P. A survey of simulated annealing as a tool for single and multiobjective optimization. Journal of the Operational Research Society, 2006; 57 (10): 1143–1160.

Suppapitnarm A, Seffen KA, Parks GT, Clarkson P. A simulated annealing algorithm for multiobjective optimization. Engineering Optimization, 2000; 33(1): 59–85.

Swarnkar R, Tiwari MK. Modeling machine loading problem of FMSs and its solution methodology using a hybrid tabu search and simulated annealing-based heuristic approach. Robotics and Computer-Integrated Manufacturing, 2004; 20(3): 199-209.

Talbi E-G. A taxonomy of hybrid metaheuristics. Journal of Heuristics, 2002; 8(5): 541–565.

Talbi E-G. Metaheuristics: from design to implementation. Wiley, 2009.

Tekinalp O, Karsli G. A new multiobjective simulated annealing algorithm. Journal of Global Optimization, 2007; 39: 49–77.

Thiele L, Miettinen K, Korhonen PJ, Molina J. A preference-based interactive evolutionary algorithm for multiobjective optimization. Evolutionary Computation Journal, 2009; 17 (3): 411-436.

Thompson M. Application of Multi Objective Evolutionary Algorithms to Analogue Filter Tuning. In: Zitzler E, Thiele L, Deb K, Coello CC, Corne D (Ed.), Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization; 2001 March 7–9, Zurich, Switzerland. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1993. 546-559.

Ulungu EL. Optimisation Combinatoire Multicritére: Détermination de l'ensemble des solutions efficaces et méthodes interactives, PhD thesis, Université de Mons-Hainaut, 1993.

Ulungu EL, Teghaem J, Fortemps P, Tuyttens D. MOSA method: A tool for solving multiobjective combinatorial decision problems. Journal of Multi-Criteria Decision Analysis 1999; 8, 221–236.

US Department of Energy, Benefits of Demand Response in Electricity Markets and Recommendations for Achieving Them, Report to the United States Congress, February 2006.

Vanderbilt D, Louie SG. A Monte Carlo simulated annealing approach to optimization over continuous variables. Journal of Computational Physics, 1984; 56: 259-271.

Veldhuizen DAV, Lamont GB. Multiobjective evolutionary algorithms: Analyzing the state-of-the-art. Evolutionary Computation, 2000; 8(2): 125-147.

Vianna DS, Arroyo JEC. A GRASP algorithm for the multi-objective knapsack problem. In: Proceedings of the XXIV International Conference of the Chilean Computer Science Society (SCCC04). IEEE Computer Society, 2004.

Voß S, Martello S, Osman IH, Roucairo C. Meta-Heuristics: Andvances and Trends in Local Search Paradigms for Optimization. Kluwer, Boston, 1999.

Wang N, Tai K. A hybrid genetic algorithm for multiobjective structural optimization. In Evolutionary Computation, 2007. CEC 2007. September, 2007, 2948-2955.

Wanner EF, Guimarães FG, Takahashi RH, Lowther DA, Ramirez JA. Multiobjective memetic algorithms with quadratic approximation-based local search for expensive optimization in electromagnetics. IEEE Transactions on Magnetics, 2008; 44(6): 1126-1129.

Wolpert DH, Macready WG. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997; 1(1): 67-82.

Xia W, Wu Z. An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. Computers and Industrial Engineering, 2005; 48(2): 409-425.

Yannibelli V, Amandi A. Integrating a multi-objective simulated annealing algorithm and a multi-objective evolutionary algorithm to solve a multi-objective project scheduling problem. Expert Systems with Applications. 2012.

Yogeswaran M, Ponnambalam SG, Tiwari MK. An hybrid heuristic using genetic algorithm and simulated annealing algorithm to solve machine loading problem in FMS, IEEE International Conference on Automation Science and Engineering, 2007. CASE 2007. 182-187.

Yogeswaran M, Ponnambalam SG, Tiwari MK. An efficient hybrid evolutionary heuristic using genetic algorithm and simulated annealing algorithm to solve machine loading problem in FMS. Int J Prod Res, 2009; 47: 5421–544

Yoshida T, Mori H. Application of Multi-objective Memetic Algorithm with Solution Diversity to Probabilistic Distribution Network Expansion Planning. IEEJ Transactions on Power and Energy, 2009; 129: 1495-1502.

Zhang W, Li F, Tolbert L. Review of reactive power planning: objectives, constraints, and algorithms. IEEE Transactions on Power Systems, 2007; 22 (4): 2177–2186.

Zhang H, Zhou J, Fang N, Zhang R, Zhang Y. Daily hydrothermal scheduling with economic emission using simulated annealing technique based multi-objective cultural differential evolution approach. Energy, 2013; 50: 24-37.

Zitzler E, Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Transactions on Evolutionary Computation, 1999; 3(4): 257-271.

Zitzler E, Laumanns M, Thiele L. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou KC and others (Eds.), Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001). International Center for Numerical Methods in Engineering (CIMNE); 2002, 95-100.

Zopounidis C, Doumpos M. Multicriteria classification and sorting methods: A literature review. European Journal of Operational Research, 2002; 138: 229–246.

Zhou A, Qu BY, Li H, Zhao SZ, Suganthan PN, Zhang Q. Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation, 2011; 1(1): 32-49.

APPENDIX A

Sorting alternatives in ELECTRE TRI - illustrative examples

		Resistive Losses	Cost	Maximum Voltage Deviation
Reference Profiles		240 260 290 320	38000 60000 85000 100000	0.01 0.03 0.065 0.09
Thresholds	Indifference	5	8000	0.005
	Preference	10	15000	0.01
	Veto	30	40000	0.08
Weights		1/3	1/3	1/3
λ		0.5		

Table A.1 – ELECTRE TRI parameters considered in the experiments.

The desirable situation would be obtaining solutions with performances lower than the inferior reference profiles (or, at least, near these values) in all objective functions, according to the preferences presented in the previous table, as happens with the solution marked with a red asterisk in Figure A.1. In this case the solution will be clearly classified in class 5. However, due to the conflicting nature of the objective functions the previous situation is, in general, very difficult to obtain. However, it is possible to achieve solutions belonging to the best class of merit with different but yet satisfactory (according to the preferences elicited) trade-offs between the objective functions. For example, the solution marked with a green rectangle in Figure A.1 belongs also to class 5. However, the value of the maximum voltage deviation is high (0.07147) when compared with the inferior reference profile. As, in this scenario, the cutting-level is 0.5 and the veto threshold is high (0.08), it does not preclude the solution from being classified in the best class of merit.

Based on the weight of each objective function, the reference profiles, the thresholds and the cutting-level, the ELECTRE-TRI method has the capability to classify solutions presenting different trade-offs between the objective functions. For example, the solution marked with a blue circle in Figure A.1 has different performances in the three objective functions. Based on the technical parameters elicited from the DM the solution is classified in class 3. A similar situation is represented with a pink triangle, but in this case the solution is classified in class 4. In fact, in the first case the discordance criterion index is greater than in the second one, which forces the solution to be classified in the class 3 instead of class 4.

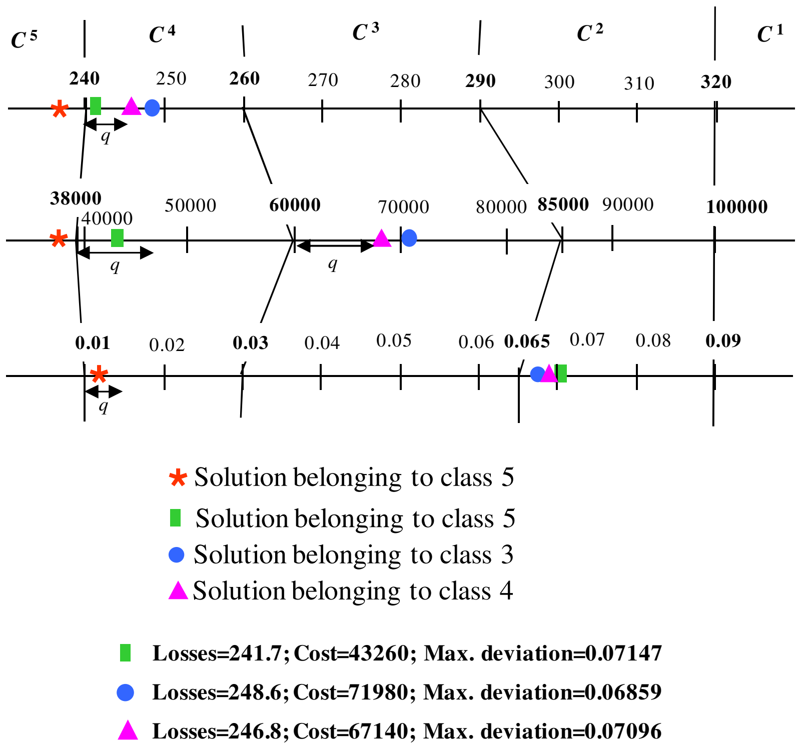


Figure A.0.1– Examples of classification of solutions using ELECTRE TRI.

APPENDIX B

Results related to the EvABOR-III algorithm

Run	Obtained classes at the final non-dominated set and respectively number of solutions	Execution time (seconds)	No. of solutions evaluated	No. of iterations performed	No. of the iteration with the first solutions in class 3	No. of iterations to complete the population
1	3	1464	13651	455	97	358
2	3	9833	89551	2985	626	2359
3	3	12504	115891	3863	2771	1092
4	2	12846	120001	4000	-	-
5	3	3389	30511	1017	767	250
6	2 (15) and 3 (15)	13116	120001	4000	2540	-
7	2	13433	120001	4000	-	-
8	2	13313	120001	4000	-	-
9	3	3410	31231	1041	375	666
10	3	5651	51301	1710	548	1162
11	2	12916	120001	4000	-	-
12	2 (21) and 3 (9)	12709	120001	4000	2655	-
13	2 (27) and 3 (3)	12902	120001	4000	934	-
14	3	3069	27931	931	272	659
15	2 (26) and 3 (4)	13074	120001	4000	3534	-
16	2 (5) and 3 (25)	13218	120001	4000	2781	-
17	2 (10) and 3 (20)	12948	120001	4000	1667	-
18	3	10606	96661	3222	876	2346
19	2 (29) and 3 (1)	13148	120001	4000	3458	-
20	3	1479	13651	455	97	358
21	3	10158	89551	2985	626	2359
22	3	12540	115891	3863	2771	1092
23	2	13105	120001	4000	-	-
24	3	3317	30511	1017	767	250
25	2 (15) and 3 (15)	13121	120001	4000	2540	-
26	2	13093	120001	4000	-	-
27	2	13063	120001	4000	-	-
28	3	3369	31231	1041	375	666
29	3	5558	51301	1710	548	1162
30	2	13131	120001	4000	-	-

Table B.1 – Values related to EvABOR-III.

Results related to the HESA algorithm

Run	classes obtained at the final non-dominated set	Execution time (seconds)	No. of solutions evaluated	No. of iterations
1	3	764	6747	204
2	2	12931	120105	4000
3	3	7140	64022	2101
4	3	5118	45389	1486
5	3	4448	39501	1283
6	2	13286	120079	4000
7	2	13594	120087	4000
8	2	13571	120126	4000
9	3	2555	22310	696
10	3	886	7509	221
11	3	6011	56350	1857
12	3	947	8068	189
13	3	3437	31633	1001
14	3	8529	77425	2539
15	3	2835	25667	835
16	2	12950	120141	4000
17	2	13070	120123	4000
18	3	1006	8733	247
19	2	13053	120103	4000
20	3	10685	97468	3186
21	2	13008	120129	4000
22	3	4042	37049	1210
23	2	13122	120108	4000
24	3	6530	59612	1949
25	2	13189	120128	4000
26	2	13221	120080	4000
27	3	11789	104011	3424
28	2	13607	120130	4000
29	3	2586	22701	736
30	3	3146	27194	867

Table B.2 – Results for the HESA algorithm.

In HESA, the iteration in which solutions belonging to the best class of merit are firstly found is the same as the total number of iterations performed, because in the local search phase at the same iteration the algorithm founds the remaining solutions belonging to the best class of merit to complete the population.

Results related to the GRASP+SA algorithm

Table B.4 and Table B.5 present results for the GRASP+SA algorithm using the set of preferences displayed in Table 4.4 in Chapter 4, while the results presented in Table B.6 and Table B.7 are related to the runs using a different set of preferences (different weights assigned to the objective functions, Table B.3).

	Maximum Power at PT1	Maximum Power at PT2	Maximum Power at SS	Profits	No. Minutes in curtailment	Max. Interval	Total Time in discomfort
Weights	20	20	20	10	10	10	10

Table B.3 – Different weights assigned to the objective functions.

The 5th column of Table B.4 to Table B.7 shows the number of the iteration in which solutions belonging to the best class of merit are firstly found and how many of these solutions are obtained in the construction phase. If a single value is indicated this means that only one solution belonging to class 3 is found in the construction phase.

Comparing the values in this column in Table B.4 and Table B.5 (equal weights assigned to the objective functions) and in Table B.6 and Table B.7 (different weights), it is possible to confirm that when only one internal cycle in the construction phase is used, a higher number of iterations of GRASP+SA is needed to obtain the first solutions in the best class of merit.

Run	Classes obtained at the final non-dominated set	Execution time (seconds)	No. of solutions evaluated	No. of the iteration with the first solutions in class 3 (number of these solutions in this iteration found in the construction phase)
1	3	3632	19838	5 (2 sol)
2	2	5705	29745	-
3	2	5919	30607	-
4	3	4904	25567	8
5	3	6183	32073	9 (2 sol)
6	2	6826	34634	-
7	3	5233	27673	8
8	3	2906	16243	4 (2 sol)
9	3	6323	32830	10
10	3	5379	28235	9
11	2	5926	30595	-
12	2	5895	30615	-
13	3	3531	19468	5
14	3	4312	22819	7 (2 sol)
15	3	6242	32950	10
16	3	2137	12023	3
17	2	6062	31256	-
18	3	5540	29141	9
19	2	6112	31491	-
20	3	4944	26151	8
21	2	5783	30017	-
22	3	4855	25283	8
23	3	4501	23797	7
24	3	3772	19911	6
25	3	6276	32807	10
26	2	6200	31540	-
27	2	5881	30040	-
28	2	6133	31186	-
29	3	5045	25710	8
30	2	6220	30951	-

Table B.4 – Results for GRASP+SA with 1 internal cycle in the construction phase.

Run	Classes obtained at the final non-dominated set	Execution time (seconds)	No. of solutions evaluated	No. of the iteration with the first solutions in class 3 (number of these solutions in this iteration found in the construction phase)
1	3	5498	28202	3
2	3	5428	27327	3
3	3	9185	45737	5
4	3	3796	19574	2
5	3	5476	27652	3 (2 sol)
6	3	7518	37444	4 (2 sol)
7	3	7421	38030	4
8	3	5157	27342	3
9	3	6902	36556	4
10	3	5354	28020	3
11	3	5531	29365	3
12	3	5600	28994	3 (2 sol)
13	3	11206	58629	6
14	3	5301	27759	3
15	3	6900	36263	4
16	2	17809	91373	-
17	3	5179	27342	3
18	3	6928	36556	4
19	3	5336	28020	3
20	3	5505	29365	3
21	3	5518	28994	3 (2 sol)
22	3	11241	58629	6
23	3	5295	27759	3
24	3	6968	36263	4
25	2	17802	91373	-
26	3	9184	48679	5 (2 sol)
27	3	5262	26968	3
28	3	11598	60609	6
29	3	12312	64095	7 (2 sol)
30	3	3414	17826	2

Table B.5 – Results for GRASP+SA with 3 internal cycles in the construction phase.

Run	Classes obtained at the final non-dominated set	Execution time (seconds)	No. of solutions evaluated	No. of the iteration with the first solutions in class 3 (number of these solutions in this iteration found in the construction phase)
1	2	6294	31760	-
2	3	6421	32850	10 (2 sol)
3	2	6155	31201	-
4	2	6170	31780	-
5	2	6466	33489	-
6	2	6172	31478	-
7	2	5919	30318	-
8	2	6020	30900	-
9	2	5974	30620	-
10	2	5999	30317	-
11	2	6116	30899	-
12	2	5874	30047	-
13	2	6168	30710	-
14	2	6036	30328	-
15	2	6808	34376	-
16	2	6128	30974	-
17	2	6142	30955	-
18	2	6241	31820	-
19	2	6243	32122	-
20	2	7280	37229	-
21	2	5993	31820	-
22	2	6200	32122	-
23	2	7186	37229	-
24	2	6038	31202	-
25	2	6012	30913	-
26	2	5839	30030	-
27	2	5765	29751	-
28	2	5870	30657	-
29	2	5964	30911	-
30	2	6217	32067	-

Table B.6 – Values related to GRASP+SA with 1 internal cycle in the construction phase considering different weights to the objective functions.

Run	Classes obtained at the final non-dominated set	Execution time (seconds)	No. of solutions evaluated	No. of the iteration with the first solutions in class 3 (number of these solutions in this iteration found in the construction phase)
1	3	17215	90184	9
2	3	6472	32555	4
3	2	16963	87151	-
4	2	18038	92219	-
5	2	17781	90522	-
6	3	14597	76446	8
7	2	18175	93102	-
8	3	17752	93133	9
9	2	18350	93999	-
10	3	11292	55451	7
11	3	14844	77432	8
12	3	18701	97881	10
13	3	5501	28436	3 (2 sol)
14	2	18025	92286	-
15	2	16984	87912	-
16	3	8865	45932	5
17	3	8854	46118	5
18	2	17943	92261	-
19	2	24405	123377	-
20	3	8864	45460	5 (2 sol)
21	3	14782	77432	8
22	3	19167	97881	10
23	3	5437	28436	3 (2 sol)
24	2	18000	92286	-
25	3	10823	55254	6 (2 sol)
26	3	10377	54416	5
27	2	18445	93110	-
28	3	10598	54050	6 (3 sol)
29	3	14431	77432	8
30	3	18686	97881	10

Table B.7 – Values related to GRASP+SA with 3 internal cycles in the construction phase considering different weights to the objective functions.

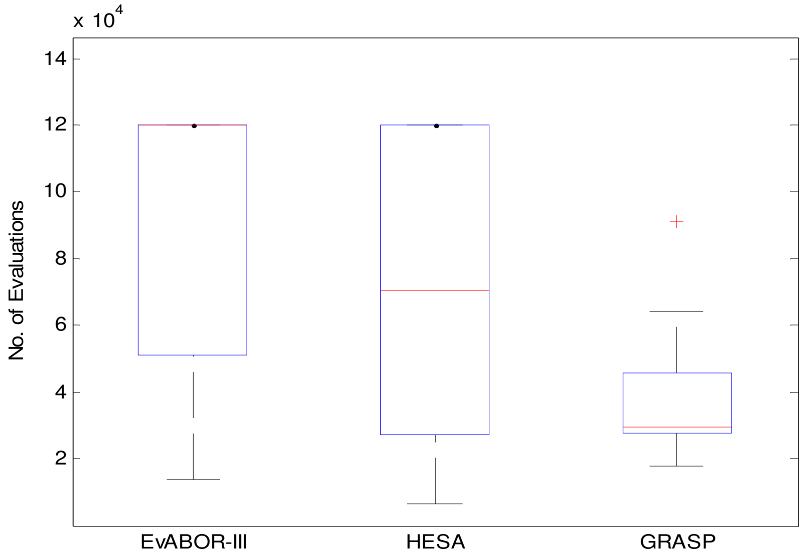


Figure B.1 – Statistical data representation for the number of solution evaluations.

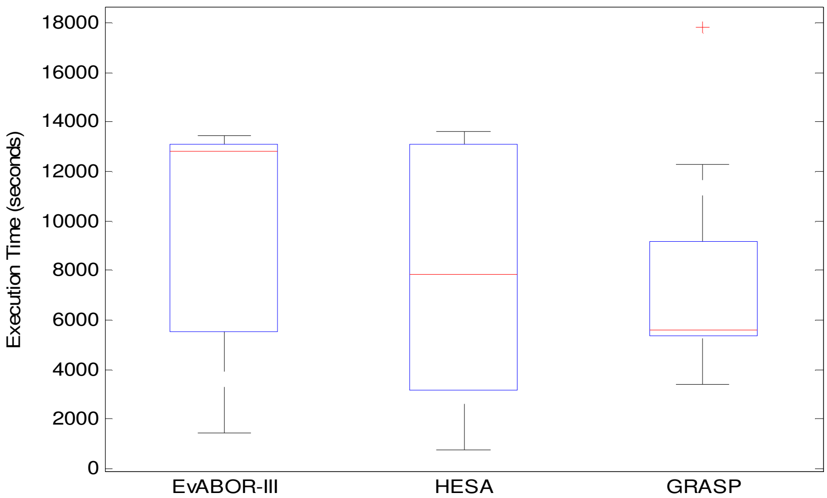


Figure B.2 – Statistical data representation for the execution time.

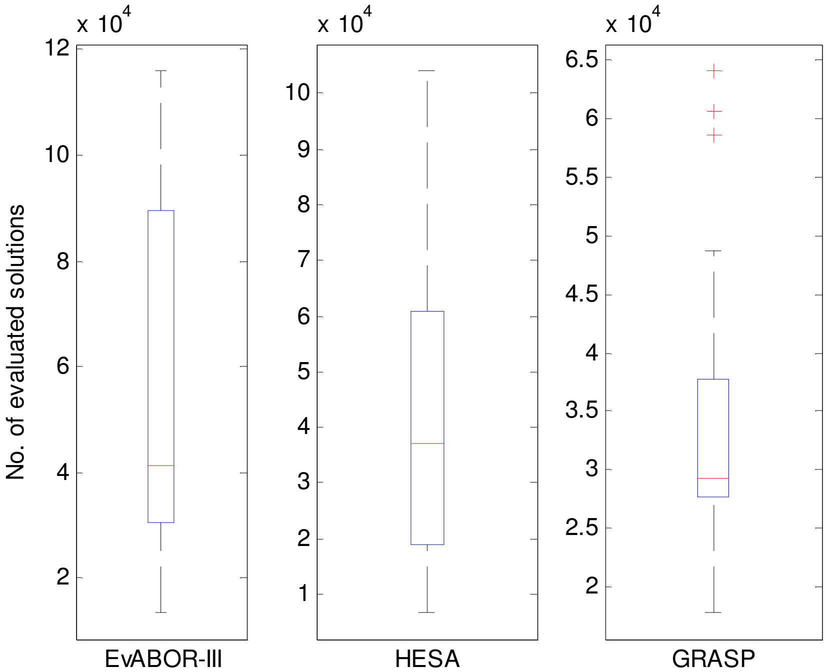


Figure B.3 – Statistical data representation for the number of solutions evaluated.

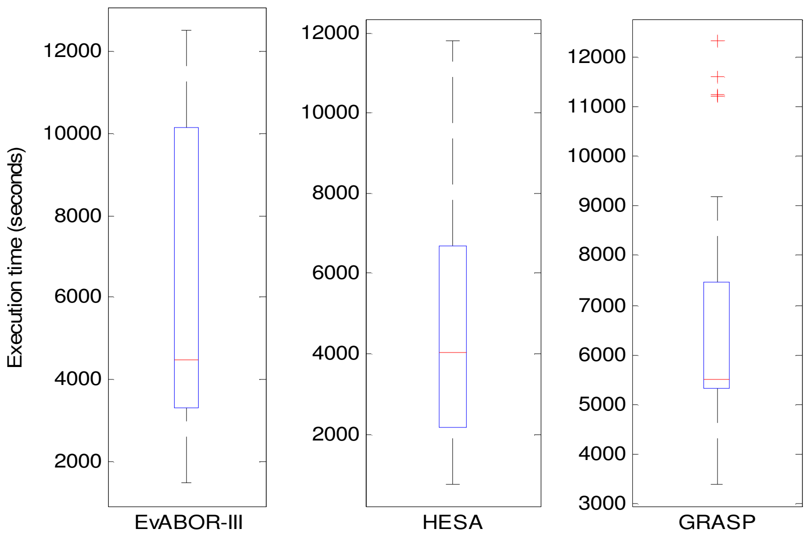
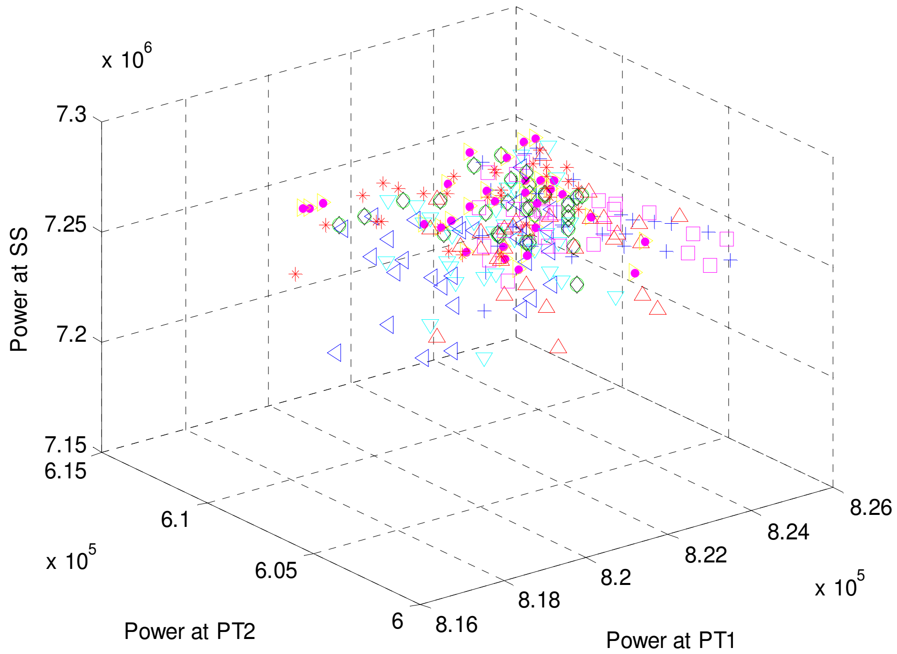
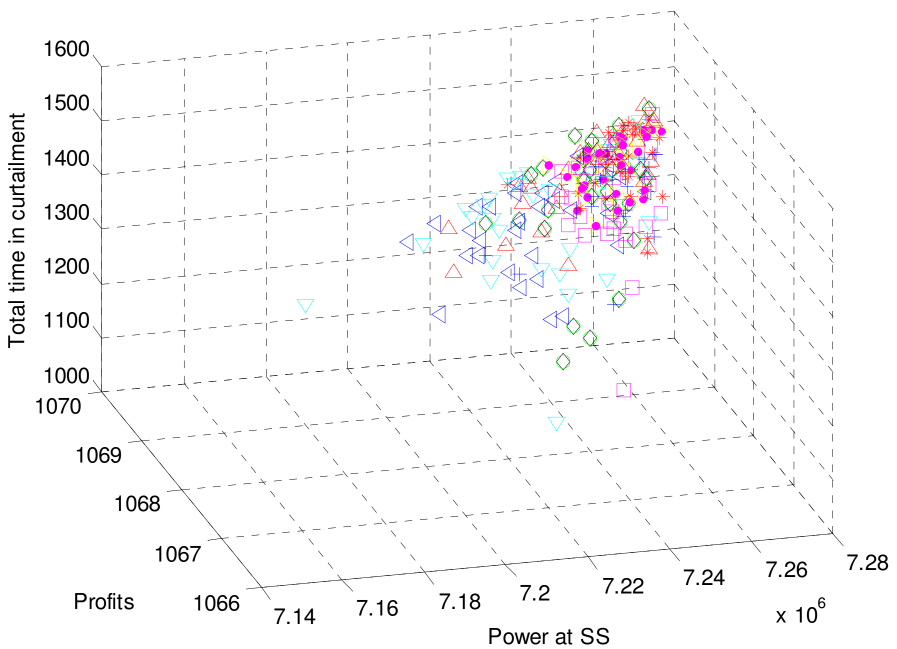


Figure B.4 – Statistical data representation for the execution time.

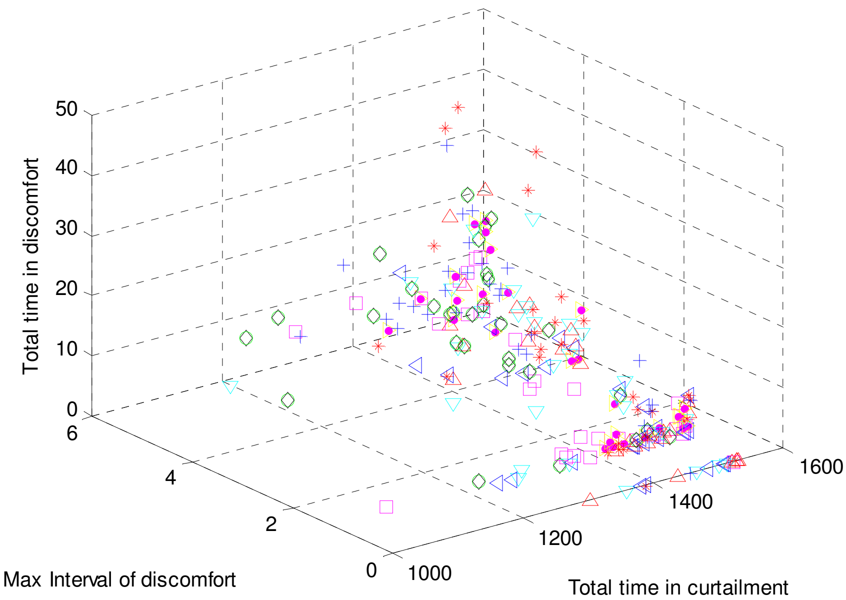
Results obtained from 10 runs of EvABOR-III



(a)



(b)



(c)

Figure B.5 – Examples of final generation obtained in 10 runs of EvABOR-III (3D representation).

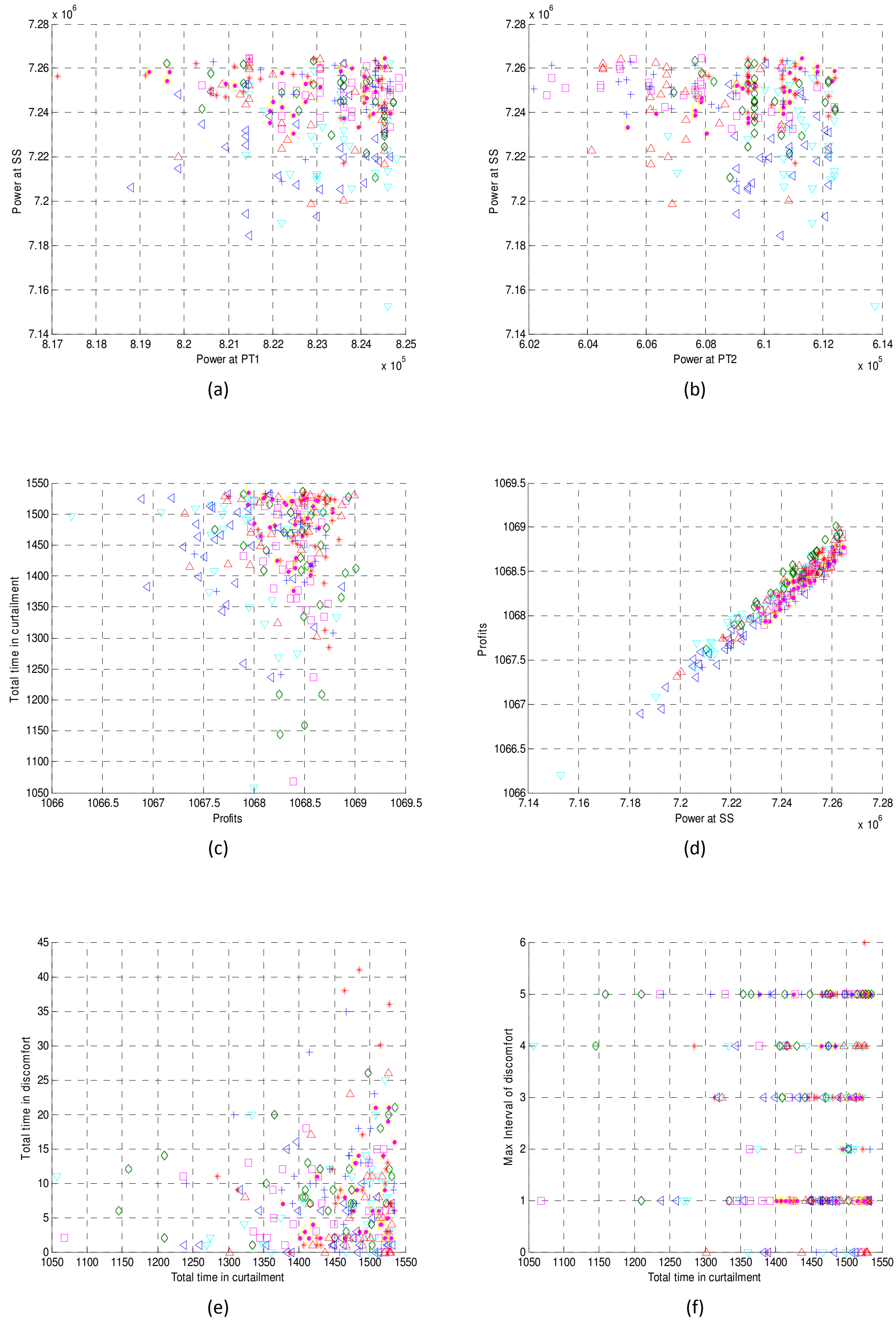
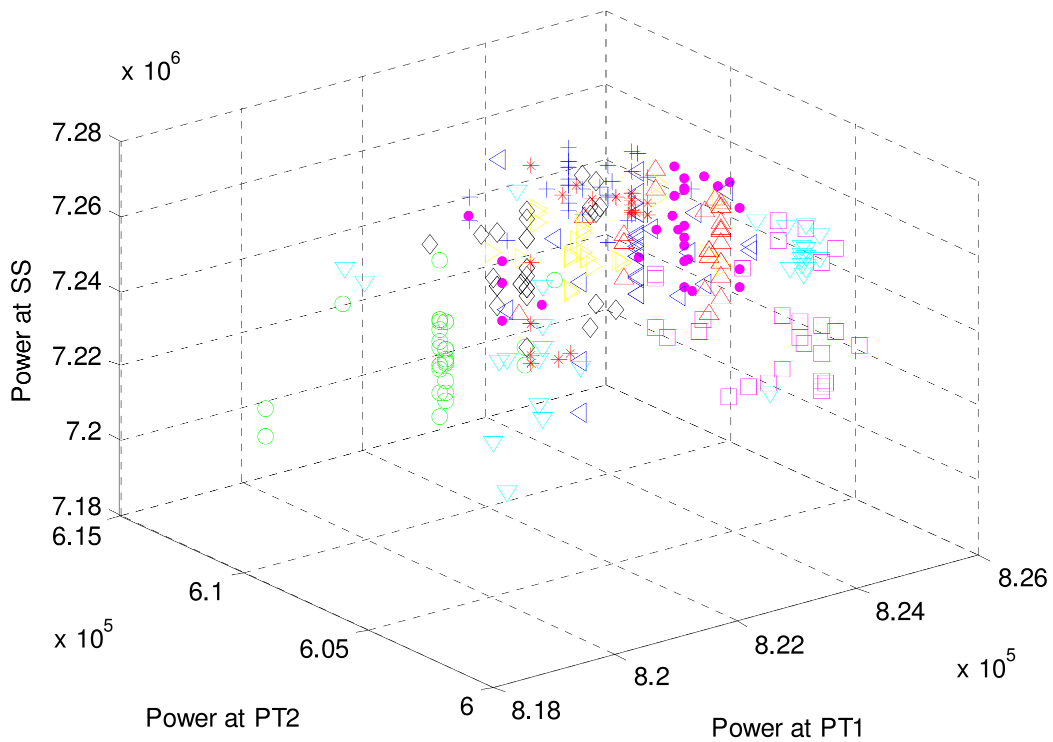
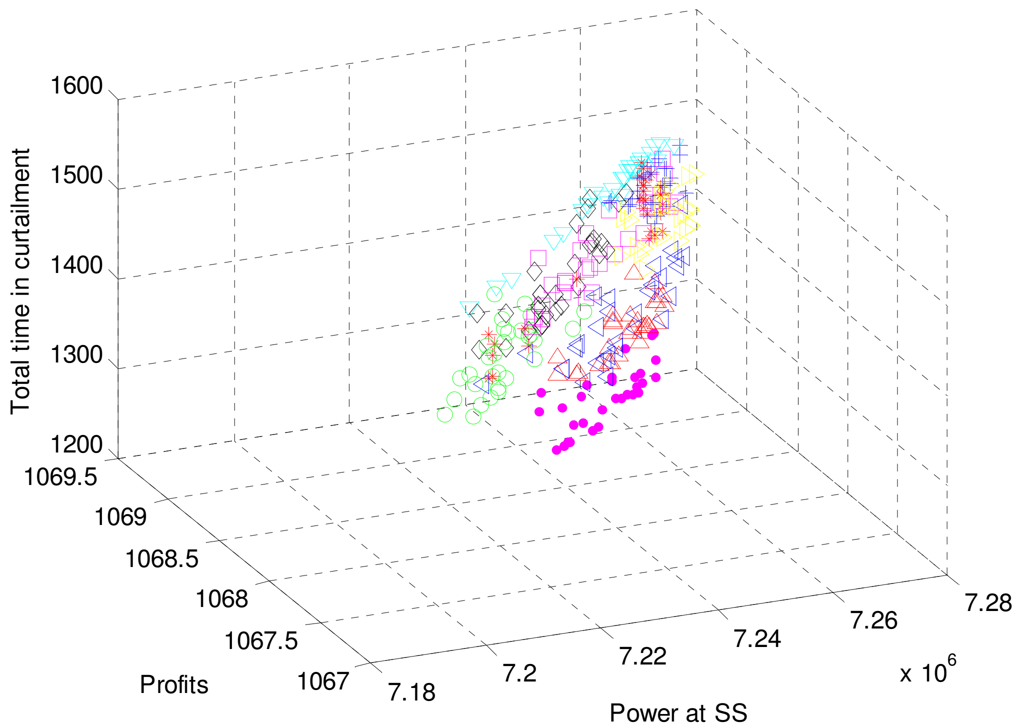


Figure B.6 – Examples of final generation obtained in 10 runs of EvABOR-III (2D representation).

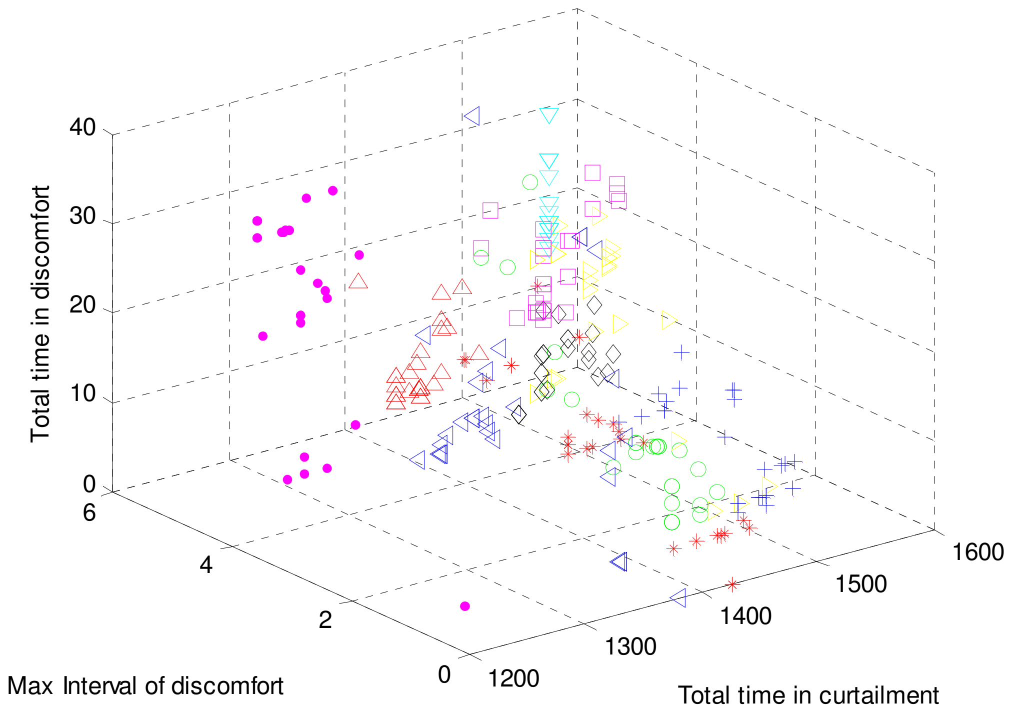
Results obtained from 10 runs of HESA



(a)



(b)



(c)

Figure B.7 – Examples of final generation obtained in 10 runs of HESA (3D representation).

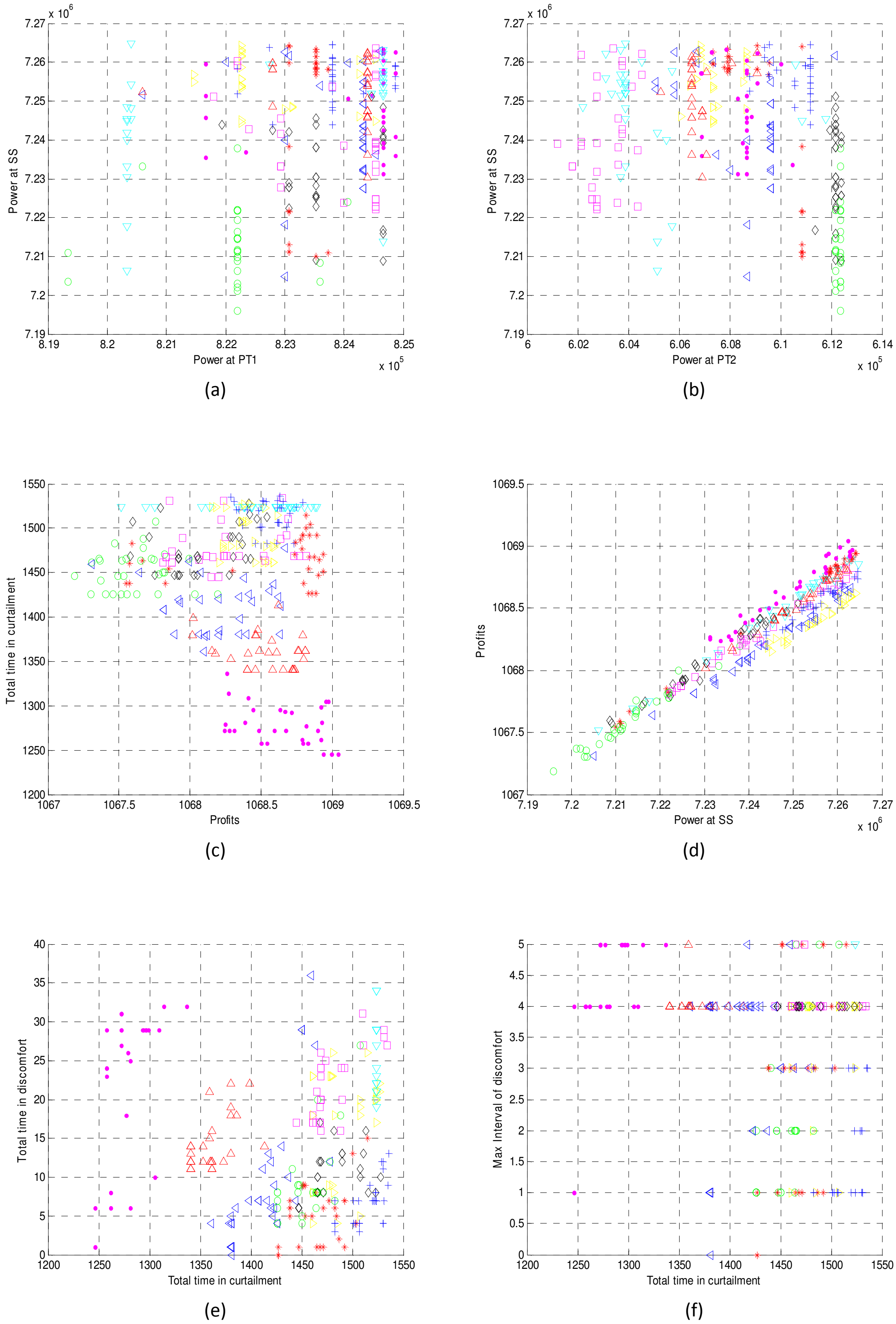
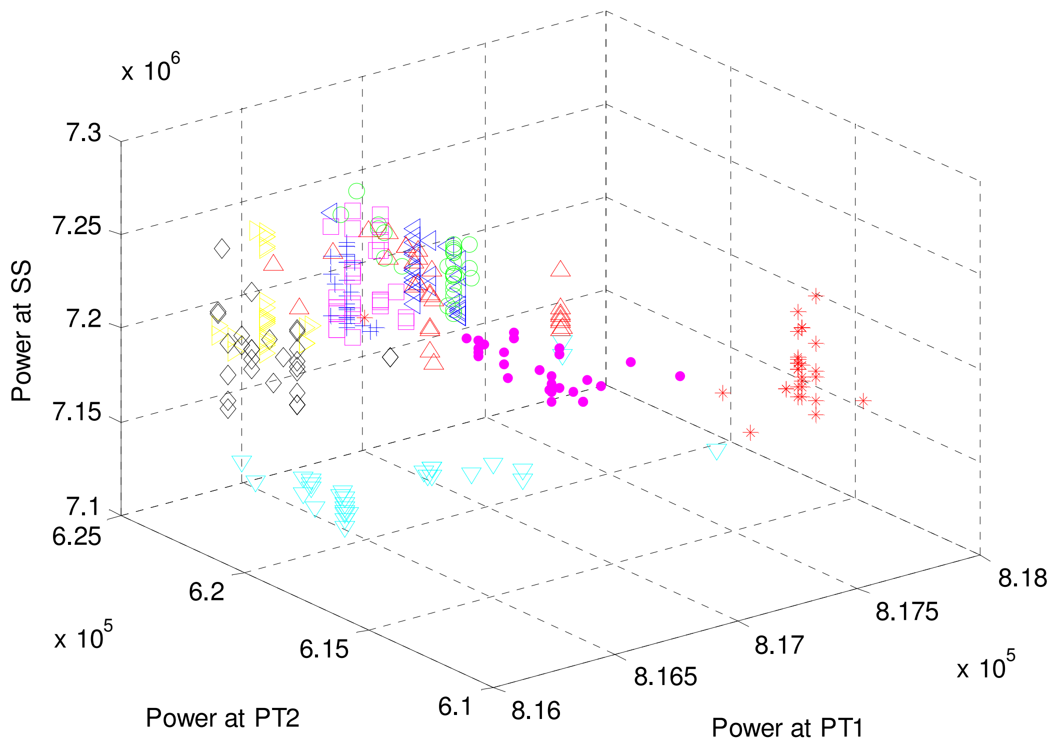
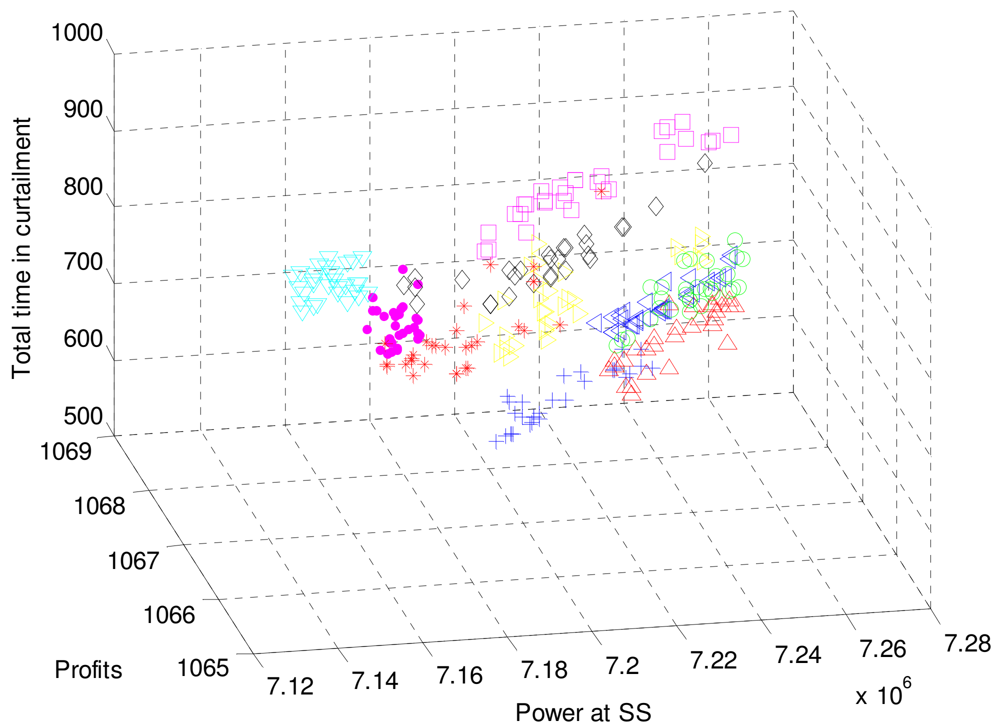


Figure B.8 – Examples of final generation obtained in 10 runs of HESA (2D representation).

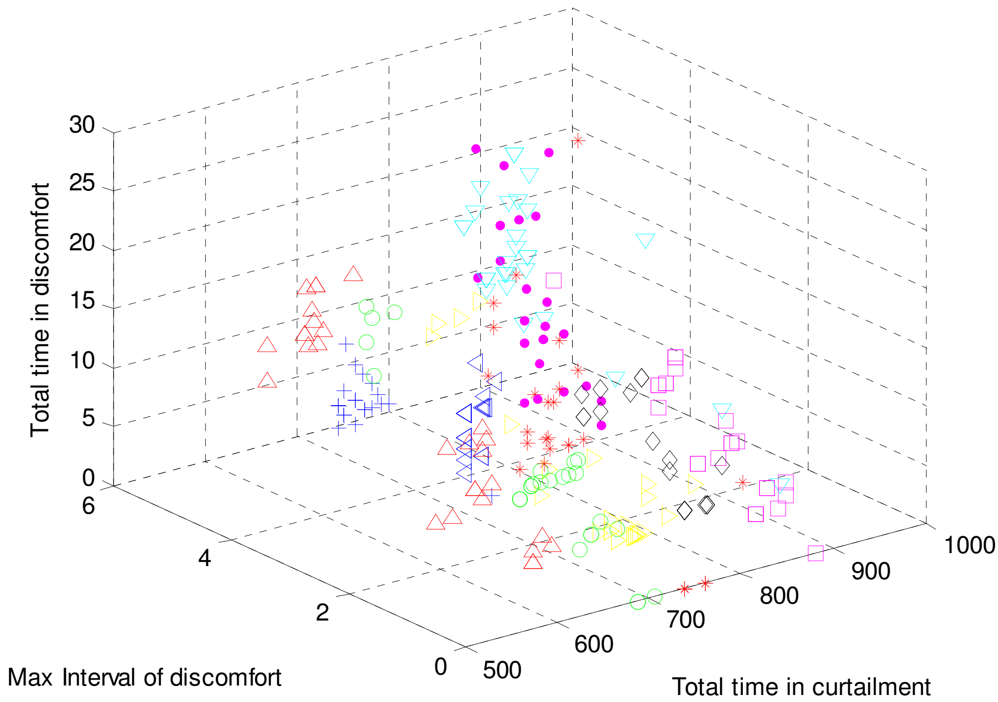
Results obtained from 10 runs of GRASP+SA



(a)



(b)



(c)

Figure B.9 – Examples of final generation obtained in 10 runs of GRASP+SA (3D representation).

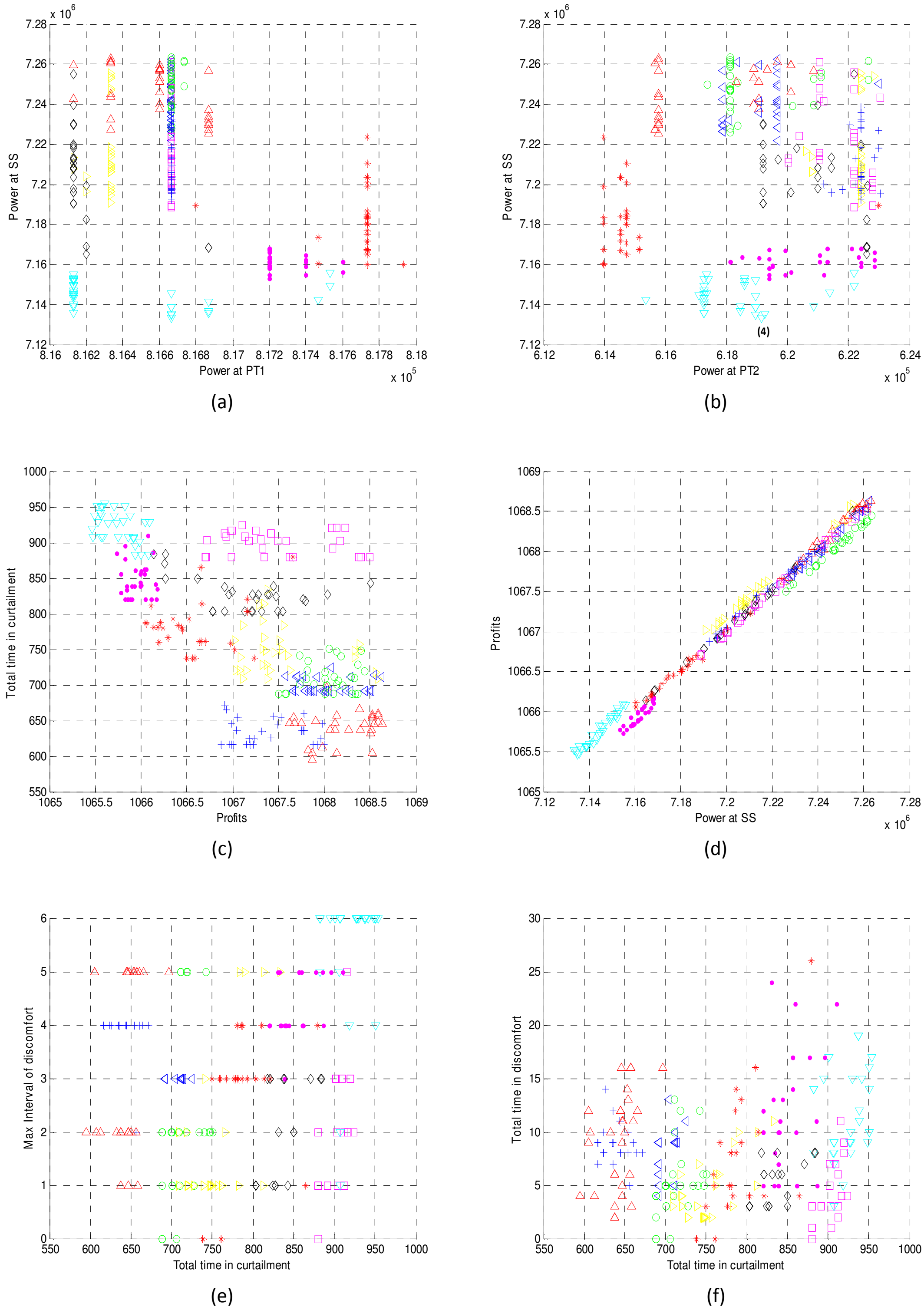


Figure B.10 – Examples of final generation obtained in 10 runs of GRASP+SA (2D representation).

PT1		PT2		SS		No. Curtailments	Profits		Max. Int. Discomfort	Total Time Discomfort
824400	-0.10%	602777.8	6.59%	7261339	1.80%	1476	1068.618874	0.29%	1	7
824666.7	-0.13%	605044.4	6.24%	7253806	1.90%	1471	1068.433839	0.31%	4	14
824333.3	-0.09%	610777.8	5.35%	7263273	1.77%	1308	1068.782582	0.27%	5	20
824600	-0.12%	602177.8	6.69%	7250539	1.94%	1534	1068.406644	0.31%	1	6
823400	0.02%	607666.7	5.84%	7261339	1.80%	1444	1068.713181	0.28%	3	12
822200	0.17%	606844.4	5.96%	7248273	1.97%	1241	1068.272467	0.32%	5	10
824400	-0.10%	605333.3	6.20%	7254806	1.88%	1424	1068.608268	0.29%	3	10
823533.3	0.01%	611377.8	5.26%	7245739	2.01%	1534	1068.166316	0.33%	5	10
822666.7	0.11%	606244.4	6.06%	7257406	1.85%	1484	1068.465999	0.30%	4	18
821733.3	0.23%	606844.4	5.96%	7259473	1.82%	1415	1068.567279	0.29%	4	29
824266.7	-0.08%	606133.3	6.07%	7252139	1.92%	1474	1068.381706	0.31%	4	15
822466.7	0.14%	605466.7	6.18%	7248139	1.97%	1501	1068.422321	0.31%	5	18
824333.3	-0.09%	611377.8	5.26%	7258606	1.83%	1507	1068.592152	0.29%	5	23
823400	0.02%	610644.4	5.38%	7245206	2.01%	1470	1068.290458	0.32%	5	14
822733.3	0.11%	610066.7	5.46%	7218873	2.37%	1375	1067.632052	0.38%	5	9
824333.3	-0.09%	606844.4	5.96%	7251473	1.93%	1393	1068.329869	0.32%	5	11
824533.3	-0.11%	608466.7	5.71%	7241939	2.06%	1437	1068.150675	0.33%	5	8
824333.3	-0.09%	609044.4	5.62%	7248606	1.97%	1465	1068.337548	0.32%	5	11
824133.3	-0.06%	606244.4	6.06%	7241206	2.07%	1473	1068.211854	0.33%	1	6
824466.7	-0.11%	606844.4	5.96%	7241606	2.06%	1463	1068.075314	0.34%	1	4
824333.3	-0.09%	610644.4	5.38%	7259006	1.83%	1416	1068.58439	0.29%	5	10
820666.7	0.36%	605466.7	6.18%	7262939	1.77%	1466	1068.717059	0.28%	5	35
821400	0.27%	607644.4	5.84%	7253339	1.90%	1390	1068.501824	0.30%	5	7
823533.3	0.01%	611777.8	5.20%	7256939	1.85%	1457	1068.412069	0.31%	0	0
823533.3	0.01%	606111.1	6.08%	7250539	1.94%	1487	1068.461786	0.30%	5	13
824266.7	-0.08%	610644.4	5.38%	7243606	2.03%	1498	1068.274395	0.32%	5	13
823600	0.00%	608200	5.75%	7243806	2.03%	1520	1068.24073	0.32%	5	14
824333.3	-0.09%	609066.7	5.62%	7256939	1.85%	1481	1068.537632	0.30%	5	7
824666.7	-0.13%	605333.3	6.20%	7239473	2.09%	1533	1068.12061	0.34%	2	9
822200	0.17%	609066.7	5.62%	7208873	2.50%	1435	1067.41697	0.40%	3	9
Max	0.36%		6.69%		2.50%	1534	1068.782582	0.40%	5	35
Min	-0.13%		5.20%		1.77%	1241	1067.416970	0.27%	0	0

Table B.8 – Objective function values obtained with EvABOR-III.

PT1		PT2		SS		No. Curtailments	Profits		Max. Int. Discomfort	Total Time Discomfort
821800	0.22%	602755.6	6.60%	7251073	1.93%	1464	1068.529348	0.30%	4	17
822200	0.17%	603755.6	6.44%	7260406	1.81%	1511	1068.689541	0.28%	4	27
822200	0.17%	603755.6	6.44%	7259206	1.82%	1534	1068.647799	0.29%	4	27
824533.3	-0.11%	601200	6.84%	7238939	2.10%	1469	1068.234117	0.33%	4	26
822933.3	0.08%	601800	6.75%	7233139	2.18%	1469	1068.116366	0.34%	4	20
824533.3	-0.11%	602755.6	6.60%	7222139	2.33%	1461	1067.807278	0.36%	4	17
824533.3	-0.11%	602755.6	6.60%	7222806	2.32%	1468	1067.829767	0.36%	4	17
824533.3	-0.11%	602755.6	6.60%	7239339	2.09%	1445	1068.203448	0.33%	4	17
824533.3	-0.11%	604355.6	6.35%	7237273	2.12%	1445	1068.149845	0.33%	4	17
824533.3	-0.11%	604355.6	6.35%	7222806	2.32%	1468	1067.830812	0.36%	4	17
824000	-0.05%	603555.6	6.47%	7223873	2.30%	1531	1067.855524	0.36%	4	29
824533.3	-0.11%	602155.6	6.69%	7239539	2.09%	1469	1068.242941	0.32%	4	24
824533.3	-0.11%	602755.6	6.60%	7224806	2.29%	1461	1067.872704	0.36%	4	17
824533.3	-0.11%	602755.6	6.60%	7256273	1.86%	1488	1068.63699	0.29%	4	16
823800	-0.02%	604155.6	6.38%	7253739	1.90%	1469	1068.515832	0.30%	4	16
824533.3	-0.11%	604533.3	6.32%	7256739	1.86%	1492	1068.680171	0.28%	4	24
824000	-0.05%	602155.6	6.69%	7238606	2.10%	1510	1068.219618	0.33%	4	31
822400	0.15%	603755.6	6.44%	7242673	2.05%	1531	1068.234773	0.32%	4	28
822933.3	0.08%	603600	6.47%	7245606	2.01%	1469	1068.34633	0.31%	4	20
824533.3	-0.11%	602755.6	6.60%	7231806	2.19%	1465	1068.059915	0.34%	5	21
824533.3	-0.11%	603600	6.47%	7233739	2.17%	1468	1068.158084	0.33%	4	19
824533.3	-0.11%	603600	6.47%	7238139	2.11%	1490	1068.247912	0.32%	4	20
822933.3	0.08%	602555.6	6.63%	7227806	2.25%	1489	1067.946558	0.35%	4	24
824533.3	-0.11%	603400	6.50%	7263539	1.77%	1469	1068.768471	0.28%	4	23
824533.3	-0.11%	602555.6	6.63%	7224806	2.29%	1461	1067.872806	0.36%	4	18
824533.3	-0.11%	604533.3	6.32%	7262073	1.79%	1469	1068.819169	0.27%	4	24
824533.3	-0.11%	602155.6	6.69%	7261939	1.79%	1469	1068.733796	0.28%	4	24
824533.3	-0.11%	602555.6	6.63%	7224806	2.29%	1474	1067.872827	0.36%	5	25
822933.3	0.08%	603755.6	6.44%	7241673	2.06%	1469	1068.259601	0.32%	4	20
822933.3	0.08%	601800	6.75%	7233139	2.18%	1469	1068.116366	0.34%	4	20
Max	0.22%		6.84%		2.33%	1534	1068.819169	0.36%	5	31
Min	-0.11%		6.32%		1.77%	1445	1067.807278	0.27%	4	16

Table B.9 – Objective function values obtained with HESA.

PT1		PT2		SS		No. Curtailments	Profits		Max. Int. Discomfort	Total Time Discomfort
816133.3	0.91%	619066.7	4.07%	7242673	2.05%	697	1068.026321	0.34%	5	16
816133.3	0.91%	620111.1	3.91%	7259539	1.82%	660	1068.592168	0.29%	5	11
816866.7	0.82%	615666.7	4.60%	7227139	2.26%	647	1067.674369	0.38%	2	5
816333.3	0.88%	615666.7	4.60%	7260473	1.81%	605	1068.521193	0.30%	5	9
816866.7	0.82%	615777.8	4.58%	7225339	2.28%	647	1067.622383	0.38%	2	11
816333.3	0.88%	615666.7	4.60%	7227139	2.26%	654	1067.668544	0.38%	5	13
816866.7	0.82%	615777.8	4.58%	7232473	2.19%	595	1067.87286	0.36%	2	4
816333.3	0.88%	615777.8	4.58%	7232473	2.19%	605	1067.871758	0.36%	5	12
816333.3	0.88%	615777.8	4.58%	7243673	2.03%	605	1068.137803	0.33%	5	12
816866.7	0.82%	615777.8	4.58%	7229858	2.22%	647	1067.749006	0.37%	2	9
816866.7	0.82%	615777.8	4.58%	7230673	2.21%	608	1067.823707	0.36%	2	10
816600	0.85%	618911.1	4.09%	7239806	2.09%	638	1068.118088	0.34%	1	2
816600	0.85%	619333.3	4.03%	7257206	1.85%	638	1068.519405	0.30%	1	3
816600	0.85%	620111.1	3.91%	7246206	2.00%	638	1068.269164	0.32%	1	2
816600	0.85%	619333.3	4.03%	7257206	1.85%	658	1068.525919	0.30%	1	3
816600	0.85%	619066.7	4.07%	7237673	2.12%	638	1068.061668	0.34%	2	6
816866.7	0.82%	615777.8	4.58%	7233873	2.17%	650	1067.906505	0.36%	2	10
816333.3	0.88%	615777.8	4.58%	7261473	1.79%	657	1068.588011	0.29%	5	16
816600	0.85%	618911.1	4.09%	7253139	1.91%	645	1068.439807	0.31%	5	12
816600	0.85%	619066.7	4.07%	7251006	1.93%	638	1068.399023	0.31%	2	6
816600	0.85%	618911.1	4.09%	7257939	1.84%	645	1068.547345	0.30%	5	12
816866.7	0.82%	615777.8	4.58%	7256673	1.86%	632	1068.5106	0.30%	2	10
816333.3	0.88%	615777.8	4.58%	7245339	2.01%	657	1068.138994	0.33%	5	16
816866.7	0.82%	615777.8	4.58%	7237273	2.12%	613	1067.982352	0.35%	2	4
816333.3	0.88%	619666.7	3.98%	7261206	1.80%	653	1068.525425	0.30%	5	14
816600	0.85%	618311.1	4.19%	7251339	1.93%	666	1068.388569	0.31%	5	12
816600	0.85%	618911.1	4.09%	7239806	2.09%	647	1068.118805	0.34%	1	4
816333.3	0.88%	615777.8	4.58%	7263073	1.77%	646	1068.635213	0.29%	5	16
816600	0.85%	620866.7	3.79%	7256939	1.85%	656	1068.521075	0.30%	2	6
816600	0.85%	620111.1	3.91%	7259539	1.82%	648	1068.591243	0.29%	5	11
Max	0.91%		4.60%		2.28%	697	1068.635213	0.38%	5	16
Min	0.82%		3.79%		1.77%	595	1067.622383	0.29%	1	2

Table B.10 – Objective function values obtained with GRASP+SA.