Optimization Methods for the Unit Commitment Problem in Electric Power Systems

Luís Augusto Correia Roque
Programa Doutoral em Matemática Aplicada
2014

Tese submetida à Faculdade de Ciências da Universidade do Porto para a obtenção do grau de Doutor em Matemática Aplicada.

Orientador
Professor Doutor Fernando Arménio da Costa Castro e Fontes,
Professor Associado, Faculdade de Engenharia
da Universidade do Porto

Coorientador
Dalila Benedita Machado Martins Fontes,
Professora Auxiliar, Faculdade de Economia
da Universidade do Porto

PhD

To my parents, Sandra,
Tomás, Miguel and those
who have left...

Resumo

Nesta tese é abordado o problema de escalonamento e pré-despacho (*unit commitment*, UC), que é um problema de otimização combinatória resultante do planeamento da operação em sistemas eletroprodutores. Neste problema pretende-se definir os períodos de funcionamento e paragem, num dado conjunto de unidades de geração e também determinar o seu nível de produção, a fim de satisfazer a procura de energia a um custo mínimo. Além disso, a solução deve satisfazer um conjunto de restrições tecnológicas.

Geralmente formulado como um problema de programação não linear inteira mista, tem sido abordado na literatura por uma grande variedade de métodos de otimização, que vão desde métodos exatos (como programação dinâmica, *branch-and-bound*) a heurísticas (algoritmos genéticos, *simulated annealing, particle swarm*,etc.).

O trabalho aqui apresentado desenvolveu-se em três frentes:
Na Primeira, o método "Hybrid Biased Random Key Genetic Algorithm" (HBRKGA) é proposto para resolver o problema tradicional de escalonamento e pré-despacho. A principal motivação para a escolha de algoritmos BRKGA foi o seu bom desempenho reportado em problemas de otimização de natureza combinatória. Na implementação do algoritmo HBRKGA, as soluções são codificados utilizando vetores de números reais no intervalo $[0,1]$. Foram realizadas simulações em sistemas de média e grande dimensão com até 100 unidades, envolvendo um horizonte de planeamento de 24 horas. Os resultados obtidos revelam que a metodologia proposta é eficaz e eficiente na

abordagem deste problema. Além disso, os resultados obtidos melhoram os resultados conhecidos até ao momento.

Na segunda, o algoritmo “Biased Random Key Genetic Algorithm” (BRKGA) foi combinado com o procedimento de ordenação de soluções não dominadas para encontrar aproximações à curva de Pareto para o problema multiobjetivo. Esta implementação assume importância devido às preocupações ambientais que estão a ter um impacto significativo sobre o funcionamento dos sistemas eletroprodutores. A abordagem ao problema tradicional de escalonamento e pré-despacho, em que apenas se procura minimizar o custo total de operação é inadequada quando as emissões ambientais também são consideradas. O procedimento de ordenação de soluções não-dominadas, similar ao algoritmo NSGA II, é aplicado para obter um conjunto aproximado da fronteira ou curva de Pareto. Os resultados das simulações para sistemas de 10, 20, 40, 60, 80 e 100 unidades de geração e horizonte temporal de 24 horas revelam a eficácia do método proposto.

Na terceira e última frente, é proposta uma formulação do problema UC como um problema de controlo ótimo inteiro misto, envolvendo variáveis de controlo binárias e reais. Posteriormente, recorrendo a um método de transformação da variável tempo esta formulação é convertida num modelo de controlo ótimo envolvendo apenas variáveis de controlo reais. Por fim, procede-se à sua discretização e o problema é transcrito como problema de programação não linear de dimensão finita, de modo a poder ser resolvido por um optimizador não linear.

Abstract

This thesis addresses the Unit Commitment (UC) problem, which is a well-known combinatorial optimization problem arising in operation planning of power systems. In the UC problem, one wishes to schedule a subset of a given set of generation units and also to determine their production output in order to meet energy demands at minimum cost over a given time horizon. In addition, the solution must satisfy a set of technological and demand constraints. This problem is typically formulated as a nonlinear mixed-integer programming problem and has been solved in the literature by a huge variety of optimizations methods, ranging from exact methods (such as dynamic programming, branch-and-bound) to heuristic methods (genetic algorithms, simulated annealing, particle swarm).

The work reported here can be divided into three parts:
First, a Hybrid Biased Random Key Genetic Algorithm (HBRKGA) is proposed to address the traditional UC problem. The main motivation for choosing a HBRKGA is its reported good performance on many combinatorial optimization problems. In the HBRKGA, solutions are encoded by using random keys, which are represented as vectors of real numbers in the interval $[0,1]$. The algorithm proposed is a variant of the random key genetic algorithm, since bias is introduced in the parent selection procedure, as well as in the crossover strategy. Computational experiments were carried out on benchmark large-scale power systems with up to 100 units for a $24-$hour period. The results obtained have shown the proposed methodology to be an effective and

efficient tool for finding solutions to large-scale UC problems. Furthermore, from the comparisons made it can be concluded that the results produced improve upon the solutions obtained by reported state-of-the-art methodologies.

Second, a multi-objective version of the problem is addressed, where environmental emissions are also considered. The environmental concerns are having a significant impact on the operation of power systems. The traditional Unit Commitment problem, which minimizes the total production costs is inadequate when environmental emissions are also considered in the operation of power plants. The Biased Random Key Genetic Algorithm (BRKGA) approach is combined with a non-dominated sorting procedure to find solutions for the multiobjective unit commitment problem. The non-dominated sorting procedure similar to NSGA II, is employed to approximate the set of Pareto solution through an evolutionary optimization process. Computational experiments with the existent benchmark systems with 10 up to 100 generation units for a $24-$ hour scheduling horizon have been performed. The comparison of the obtained results with those of other UC multiobjective optimization methods reveals the effectiveness of the proposed method.

Third and finally, the UC problem is formulated as a mixed-integer optimal control problem, with both binary-valued control variables and real-valued control variables. Then, through the use of a variable time transformation method it is converted into an optimal control problem with only real-valued controls. Finally, this problem is discretized and transcribed into a sparse finite-dimensional nonlinear programming problem and solved using a sparse optimization solver.

Acknowledgments

It is not always easy to find the appropriate words to express how much we are thankful to all the persons that somehow contribute to this work. Nevertheless, I would like to express my thanks to all those people who made this thesis possible and an unforgettable experience for me.

Special words have to be directed to my supervisors, Fernando Fontes and Dalila Fontes. I am deeply indebted to Prof. Fernando Fontes for giving me a research direction and for all the help, ideas, comments and constant encouragement throughout this work. I also thank Prof. Dalila Fontes, for her important contributions, valuable opinions, discussions, ideas and availability with immediate responses when necessary, supporting and encouraging me at all time. Her enthusiasm is inspiring to all of us.

I would like to thank also to my colleagues at laboratory I202 and the people of Institute for Systems and Robotics-Porto (ISR-P) at the Faculdade de Engenharia da Universidade do Porto (FEUP), for their friendship, support and valuable discussions about the topics of this work. Many thanks also go to my colleagues of Instituto Superior de Engenharia do Porto (ISEP), who were able to create a friendly atmosphere for the development of this work.

A special thanks to my parents, brothers, my wife Sandra, my two sons João Tomás and Miguel, and the rest of my family for their affections, trust, and support. Without them, nothing of this would be possible.

Finally, I would like to thank the support from ERDF through the Programme COMPETE and by the Portuguese Government through FCT - Foundation for Science and Technology, projects PTDC/EGE-GES/099741/2008 and PTDC/EEA-CRO/116014/2009.

Contents

List of Tables

List of Figures

Abbreviations and Symbols

Abbreviations

GA	Genetic Algorithm
BRKGA	Biased Random Key Genetic Algorithm
HBRKGA	Hybrid Biased Random Key Genetic Algorithm
SPEA	Strenght Pareto Evolutionary Algorithm
NSGA	Non-dominated Sorted Genetic Algorithm
NPGA	Niched Pareto Genetic Algorithm
NCO	Necessary Conditions of Optimality
OCP	Optimal Control Problem
CVP	Calculus of Variation Problem
MOP	Mutiobjective Optimization Problem
NLP	Non-Linear Programming

Symbols

$\mathbf{t}$	Time period index
$\mathbf{T}$	Number of time periods (hours) of the scheduling time horizon
$\mathbf{j}$	Generation unit index
N	Number of generation units
N_p	GA population size
N_c	Comparison set size in NPGA multiobjective approach
N_{Gers}	Number of GA generations
G_{max}	Maximum Number of GA generations
$\mathbf{y_{t,j}}$	Output generation of unit j at time period t, in $[MW]$
$\mathbf{u_{t,j}}$	Status of unit j at time period t (1 if the unit is on; 0 otherwise)
$\mathbf{T_j^{on}(t)}$	Number of time periods for which unit j has been continuously on-line until time period t, in $[hours]$
$\mathbf{T_j^{off}(t)}$	Number of time periods for which unit j has been continuously off-line until time period t, in $[hours]$
$\mathbf{R_t}$	System spinning reserve requirements at time period t, in $[MW]$
$\mathbf{D_t}$	Load demand at time period t, in $[MW]$
$\mathbf{Ymin_j}$	Minimum generation limit of unit j, in $[MW]$
$\mathbf{Ymax_j}$	Maximum generation limit of unit j, in $[MW]$
$\mathbf{T_{c,j}}$	Cold start time of unit j, in $[hours]$
$\mathbf{T_{min,j}^{on/off}}$	Minimum uptime/downtime of unit j, in $[hours]$
$\mathbf{S_{H/C,j}}$	Hot/Cold start-up cost of unit j, in [$]
$\Delta_j^{dn/up}$	Maximum allowed output level decrease/increase in consecutive periods for unit j, in$[MW]$
$\mathbf{Sd_j}$	Shut down cost of unit j, in [$]
$\mathbf{Se_{t,j}}$	Start-up atmospheric pollutant emission of unit j, in $[ton-CO_2]$ if CO_2 or $[mg/Nm^3]$ if nitrogen oxides

Chapter 1

Introduction

Power systems are one of the most important infrastructures in a country since the commodity involved is essential to everyday life. Nowadays, its availability and price are critical to many companies and business [91, 92]. In recent years, the power generation industry has seen considerable growth. Due to the increase of economy and productivity, the usage of electricity is rising. In the past 10 years (from 2000 to 2010), in the Euro area, the growth has been about 13%, while in Portugal it has been about 25%, as it can be seen in Figure 1.1. With the increasing importance of the role the power sector plays in the modern society, a lot of effort has been put into developing a secure, reliable and economic power supply. The Unit commitment is crucial in achieving this goal, thus the quality of its solution is of the highest importance.

The study and operation of power systems involve solving many different optimization problems [59]. Amongst these problems, the Unit Commitment (UC) problem stands out as it plays a key role in planning and operating power systems. The power generation industry utilizes unit commitment and economic dispatch to help make generation scheduling decisions. An optimal scheduling of the generating units has the potential of saving millions of euros. The objective of a UC problem is to identify a schedule of committing units to minimize the joint cost of committing and decommitting units

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	Share in EU-27, 2010 (%)
EU-27	2 863	2 943	2 964	3 050	3 119	3 140	3 182	3 196	3 203	3 045	3 181	100.0
Euro area	1 995	2 040	2 069	2 136	2 194	2 204	2 245	2 258	2 275	2 159	2 268	71.3
Belgium	80.3	76.2	78.1	80.9	81.7	83.4	82.0	85.1	81.4	87.5	91.4	2.9
Bulgaria	36.9	39.6	38.6	38.5	37.5	40.3	41.6	39.1	40.7	38.7	42.2	1.3
Czech Republic	68.0	68.8	70.4	76.7	77.9	76.2	77.9	81.4	77.1	76.0	79.5	2.5
Denmark	34.4	36.2	37.3	43.8	38.4	34.4	43.2	37.4	34.9	34.5	36.8	1.2
Germany	538.5	548.2	549.3	567.9	576.8	581.6	597.2	598.4	598.9	556.8	591.4	18.6
Estonia	7.6	7.6	7.7	9.2	9.2	9.1	8.7	11.0	9.5	7.9	11.7	0.4
Ireland	22.7	23.7	23.9	24.1	24.4	24.8	26.1	26.9	28.9	27.1	27.5	0.9
Greece	49.9	49.7	50.6	54.3	54.9	55.7	56.5	59.1	59.4	56.1	53.4	1.7
Spain	214.4	226.0	232.7	250.2	268.7	282.1	287.7	293.2	301.5	283.4	292.1	9.2
France	516.9	526.3	534.9	542.5	549.7	550.3	549.1	544.6	548.8	515.2	544.5	17.1
Italy	263.3	266.0	270.8	280.2	290.0	290.6	301.3	301.3	307.1	281.1	290.7	9.1
Cyprus	3.2	3.4	3.6	3.8	4.0	4.1	4.4	4.6	4.8	4.9	5.1	0.2
Latvia	3.7	3.7	3.5	3.5	4.2	4.4	4.5	4.4	4.9	5.2	6.1	0.2
Lithuania	10.0	13.2	16.1	17.9	17.7	13.6	11.4	12.9	12.8	14.1	5.3	0.2
Luxembourg	1.1	1.6	3.7	3.6	4.1	4.1	4.3	4.0	3.5	3.8	4.6	0.1
Hungary	32.3	33.7	33.5	31.4	31.3	33.2	33.3	37.2	37.4	33.3	34.6	1.1
Malta	1.8	1.8	1.9	2.1	2.1	2.2	2.1	2.2	2.2	2.0	2.0	0.1
Netherlands	86.0	89.9	92.1	93.0	98.4	96.2	94.4	100.9	103.4	108.9	114.3	3.6
Austria	59.1	60.1	59.9	57.4	61.4	63.5	62.2	63.2	64.6	66.8	68.3	2.1
Poland	132.2	132.7	131.4	138.4	140.8	143.6	147.7	145.4	141.5	137.9	142.9	4.5
Portugal	42.2	44.8	44.4	45.4	43.5	45.0	47.5	45.9	44.6	48.7	52.8	1.7
Romania	48.6	50.4	51.1	51.3	52.7	55.5	58.4	56.2	60.1	52.8	55.6	1.7
Slovenia	12.8	13.6	13.7	12.9	14.3	14.1	14.1	14.0	15.4	15.4	15.4	0.5
Slovakia	27.7	29.6	30.0	28.7	28.2	29.3	28.9	25.8	26.6	24.1	25.4	0.8
Finland	67.3	71.2	71.6	80.4	82.2	67.8	78.6	77.8	74.5	69.2	77.2	2.4
Sweden	141.6	157.6	143.2	132.5	148.5	154.6	140.4	145.1	146.4	133.3	145.3	4.6
United Kingdom	360.8	367.4	370.1	380.1	376.9	380.5	378.8	379.1	372.4	360.2	365.3	11.5
Norway	142.3	121.0	130.1	106.8	109.8	137.4	121.0	136.6	141.5	131.1	123.5	-
Switzerland	65.5	70.3	65.1	65.3	63.6	57.8	62.0	65.8	66.8	66.4	66.1	-
Croatia	10.2	11.7	11.7	12.1	12.8	12.0	12.0	11.7	11.8	12.4	13.6	-
Turkey	118.7	116.3	123.7	135.2	145.1	155.5	169.5	183.3	189.8	186.6	203.0	-

Source: Eurostat (online data code: nrg_105a)

Figure 1.1: Net electricity generation, in thousands of GWh, in Europe for 2000-2010.

and economic dispatch. At the same time, it meets the forecasted demand and spinning reserve requirements, that allow for uncertainty compensation and technological generating unit constraints.

The UC problem is computationally challenging due to the nonlinear objective function, the mixed-integer features, and the large dimension. For this reason, obtaining an optimal or even a good sub-optimal solution is a great challenge. Also, the UC problem has been addressed by many researchers using a large variety of optimization methods. Therefore, it can provide an excellent benchmark to test optimization methodologies that are being developed.

In recent years, environmental factors have been given increasingly importance. Keeping that in mind, the industry is also turning its attention to the emissions of pollutants, many of which come from fossil fuels. This issue has been addressed by minimizing the pollutants emissions, in addition to the costs. The main reasons for addressing these issues in this way, is that a trade-off between the two is of utmost importance since the taxes associated with such emission and also with the emissions trading market has been established, whereby companies can buy or sell emissions allowances.

The unit commitment problem is an important research challenge and a vital optimization task in the daily operational planning of modern power systems due to its combinatorial nature. In general, the UC problem may be formulated as a non-linear, large scale, mixed-integer combinatorial optimization problem with both binary (unit status variable) and continuous (unit output power) variables.

Since the main aim of this thesis is to develop approaches for addressing realistic sizes unit commitment problems, heuristic methods are proposed here. This type of methods have no guarantees of converging to an optimal solution, however, they generally provide good solutions in a reasonable computational time.

Another major achievement of this thesis is the fact that the proposed approach is flexible and can easily be successfully adapted to unit commitment problems with different characteristics and considering different issues. This is demonstrated by addressing the bi-objective unit commitment problem.

Finally, the other important contribution is due to the optimal control approach to the UC problem. This approach is of a different nature and this way it allows the exploration of different perspectives of the problem.

1.1 Thesis Overview

This thesis is organized as follows:
Chapter 2 contains the traditional single-objective and environmental/economic Unit Commitment problems description.

Chapter 3 gives a mathematical formulation for the single-objective UC problem, which is followed by a discussion on previous approaches. Then, the proposed Biased Random Key Genetic Algorithm (BRKGA) is introduced and explained. Finally, the performed computational experiments are reported and the efficiency and effectiveness of the approach is demonstrated by using benchmark systems with up to 100 units for a 24 hour planning horizon. This chapter is an author version of the paper published in *Journal of Combinatorial Optimization* entitled ”A Hybrid Biased Random Key Genetic Algorithm Approach to the Unit Commitment Problem” [96]. A previous version has been presented at an international conference and published in *Lecture Notes in Computer Science* [95].

Chapter 4 discusses and reviews the bi-objective UC problem before providing its mathematical formulation. It follows the description of how the BRKGA is adapted to the bi-objective optimization UC problem. This chapter also provides a comparative study of the proposed BRKGA method and other multi-objective optimization techniques. Test systems with 10 and to up 100 thermal units have been used. This chapter is also an author version of a paper submitted to an international journal. A preliminary version of the work in this chapter has been presented at an international conference and published in its proceedings [97].

Chapter 5 contains the single-objective UC problem formulated as an optimal control problem (OCP). It proposes a variable time transformation method that converts the mixed integer OCP into an OCP with only real-valued variables and thus, it is possible to find local solutions through NLP solvers.

This chapter is an author version of the article published as chapter 6 of the book, *Dynamics of Information Systems*, Springer, due December 2014 [34]. A preliminary version of this work has been presented at an international conference and published in its proceedings [38].

The final discussions and conclusions are given in Chapter 6, ending with the outline of a future work.

Chapter 2

The Unit Commitment Problem

The electric power sector has been subject to new global challenges and evolutions such as, the growing electricity demand, the security of supply, an environmental sustainability and competitiveness. The liberalization of the electric power markets, which has been happening, promotes further needs for optimization within the power sector.

Energy investment planning, operation, pricing and management activities are performed in an hierarchical and sequential procedure seeking to determine an economic, reliable and environmentally sustainable energy supply. This complex decision-making procedure includes the Unit Commitment which is an important problem for power systems planning and operation in European countries and around the world. The complexity of the UC problems depends on the diversity of the generating unit characteristics and on the size of the energy systems under consideration. The most representative costs are associated to thermal power generation.

The Unit Commitment (UC) problem involves determining which power generating units should be online at each time period and how much power each of these online units should be producing. By optimizing such decisions, power utilities can produce power at a lower cost, while satisfying demand and other operational constraints. The former constraints are used to ensure security and reliability of supply, while the latter are technological and reduce the freedom in the choice of starting-up and shutting-down generating units and the range of power production values.

The Unit Commitment problem context can vary from one market structure to another. For instance, in a regulated, vertically integrated monopoly electric power system, the decision is made centrally by the utility. The objective is the cost minimization subject to demand and spinning reserve satisfaction and other system operating constraints. In a competitive environment, each generating company needs to decide which units should be online, such that its expected profit is maximized, given the demands, costs, and prices, as well as other system operating constraints. Each individual generating company optimises the UC of its generation units, considering the market price.

However, ideal markets, in principle, ensure the same conditions for the scheduling of the generating units as one would encounter in the centralized UC problem. Therefore, the difference between scheduling the generating units in liberalized or traditional markets is not meaningful. Thus, solutions to the UC problem based on cost of the traditional regulated market are still relevant [13, 116].

Many different versions of the UC problem exist and have been studied in the literature. This thesis starts by addressing the classical UC problem and then moves on to the bi-objective UC problem.

2.1 Problem relevance

Historically, the power systems were vertically integrated. Thus, a power company owned the generating plants, the high voltage transmission system, and the distribution lines. Recently, the power industry was restructured and vertically integrated companies needed to separate their assets and services. Therefore, they are now divided into generation companies, transmission companies, and distribution companies. After the separation, the generation companies focused their attention on how to maximize profit and not being too concerned with whether the demand and reserve are completely met. In a competitive market, where the customers buy from whoever provides the energy at lower prices, reducing production costs increases the chances of competing with other suppliers in the regional, national or international markets.

Thus in this thesis we address the deterministic unit commitment problem, where uncertainties are coped through the imposition of spinning reserve as explained in the next section. Two versions of the UC problem are considered. First, we study the deterministic economic UC problem and thus we need to determine the on and off status of the generating units, as well as the power generated by each unit over a time horizon. The choice of such decisions is made such that the total operating costs are minimized, while satisfying a set of system and unit constraints. After, the environmental/economic UC problem is addressed. The environmental aspects lead us to consider the emissions produced by power plants. Conventional electrical systems are highly fossil fuel dependent, being the major contributors to the greenhouse gas emissions. The conflicting objectives of the pollutant emissions and operating costs minimization are considered simultaneously.

2.2 Problem description

As it was already said, the UC problem consists of a set of generation units for which one needs to decide when each unit is on-line and off-line along a predefined time horizon. In addition, for each time period and each on-line unit it is also decided how much it should be producing. Therefore, the problem includes two types of decisions, which are limited by two types of constraints: load constraints and technological ones. Since in principle there are many solutions satisfying the constraints, one must define a performance measure, which typically is the minimization of the total costs incurred with the operation of the generating units and the production of the required power.

The traditional unit commitment problem, the most commonly addressed version, is a deterministic single objective optimization problem. The objective function is the minimization of the total operating costs over the scheduling horizon. The total operating costs can be expressed as the sum of the fuel, shutdown and start-up costs. System uncertainties are addressed through deterministic reserve policies, which enforce reserve to be a certain percentage of the peak load. The stochastic natures of power systems are therefore modeled in the optimization problems. In this thesis, the traditional UC problem is addressed in Chapter 3.

Other UC problem variants including additional complexities, and perhaps more adapted to current energy market conditions, have also been considered in the literature. Among the different issues considered in the literature we chose to address the pollutant emissions issue. The choice of the pollutants issue is mainly due to the fact that, in recent years, environmental concerns have been gaining importance. Several policies regarding pollution have been proposed by the policy makers. For example, taxes on pollutant gases such as CO_2 have been added, pollution allowances have been implemented, an internal market for carbon dioxide allowances has been established by the European Union (where companies can buy or sell pollution allowances).

Chapter 4 addresses a bi-objective UC problem, in which the pollutant emissions are to be minimized, in addition to costs minimization. The pollutants issue has been addressed in the literature in different ways. For example, [63] considers the emission constrained UC problem; [66] considers the fuel constrained UC problem; [15, 118] consider the price-based UC problem. We chose to consider the bi-objective (environment and economic) UC problem because, although the emission of pollutants is restricted, there is a market to buy additional permissions. Thus, with our bi-objective optimization problem, we have the possibility to trade-off between cost and pollution, having in this way a more flexible, and potentially better solution.

2.2.1 Objective function: total operating cost minimization

As it is said before, in the single objective UC problem, the objective function considered is the minimization of the total costs. There are 3 types of costs: generating costs, start-up costs and shut-down costs. These costs are incurred by each generating unit at each time period.

The generating costs, which are mainly due to the fuel consumption, are usually modeled as a quadratic function concerning to the production level (y_t). An illustration of the cost of the generation cost function is provided in Figure 2.1.

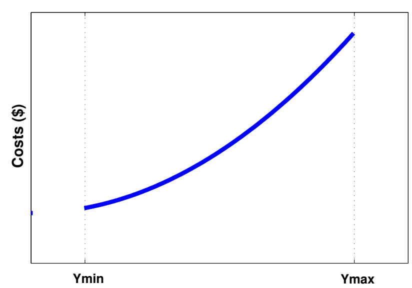

Figure 2.1: Fuel cost function.

Start-up costs are incurred every time a generating unit is started and quite often are considered constant. However, in the case of plants with steam turbine, the start-up costs should not be considered constant since they depend on the time that the unit has been down and also on the state of the boiler, i.e. hot or cold. If the boiler is kept hot during the downtime period (banking) then the start-up costs are generally modeled as a linear function of time,

$$S(t) = b_0 + c_1.T^{off}(t), \tag{2.1}$$

where $T^{off}(t)$ is the number of periods that a unit has been continuously down until time period t, $b_0(\$)$ is the fixed start cost and $c_1(\$/h)$ is the cost coefficient associated to fuel consumption in order to maintain the required temperature.

However, if the boiler is left to cool-down (cooling) then the start-up costs are typically considered as exponentially time dependent as in the following equation,

$$S(t) = b_0 + b_1 . \left(1 - e^{-T^{off}(t)/\alpha}\right), \tag{2.2}$$

where b_1(\$) is the cold start up cost and α is the cooling constant.

In the case of diesel groups, the start-up costs are much harder to model since they may assume intermediate levels of heating and fuel exchanges. In general, simplified cost functions are used and in the literature it is quite frequently a two-step function,

$$S_{(t)} = \begin{cases} S_H, & \texttt{if } T_{min}^{off} \leq T^{off}(t) \leq T_{min}^{off} + T_c, \\ S_C, & \texttt{if } T^{off}(t) > T_{min}^{off} + T_c, \end{cases} \tag{2.3}$$

where T_{min}^{off} is the minimum required downtime of the given unit, S_H and S_C are the costs incurred for a hot and cold start-up , respectively, and T_c is a unit parameter such that $T_{min}^{off} + T_c$ indicates the number of hours that the boiler needs to cool down. A graphical representation of the start up cost function is given in Figure 2.2.

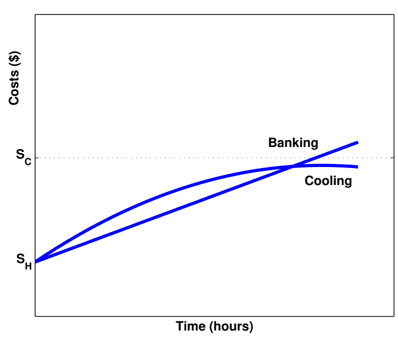

Figure 2.2: Start up cost function in plants with steam turbines.

The shut down costs are associated to the necessary conditions to be able to provide a hot start up (banking). Typically, shut-down costs are considered to be constant and frequently disregarded since they can be included in the start-up costs [4].

2.2.2 Objective function: pollutants emission minimization

In response to environmental concerns, as the global warming trend, the United Nations Framework Convention on Climate Change (UNFCCC), initiated in 1992, and the Kyoto protocol, which began in 1997, provide an international framework where many countries and international organizations undertake to reduce their greenhouse gases emissions [134]. For example, in the European Union, it was required to reduce its greenhouse gas emissions by 8% during the period from 2008 to 2012 compared to 1990 levels [16]. It should be emphasized that power industry is one of the largest pollutant emissions source, nearly 40% of the CO_2 has being emitted by fossil fuel combustion. For instance, in 2005 about 31% of Portuguese CO_2 emissions resulted from public electricity and heat production [33]. There is a need to develop emission policies, technologies, and operations in power systems that help in the reduction of total pollutant emissions [134]. However, since the demand for energy has been growing, it is a very difficult challenge to revert this trend in a short term.

The UC problem, including simultaneous minimization of operating costs and pollutant emissions (bi-objective function), can provide information regarding the trade-off between costs and pollutant emissions. This is very important for decision-makers, since these two aspects are now further related due to the internal market for carbon dioxide allowances established by the European Union (the European Union Emissions Trading Scheme). The electric power production companies can now sell their allowances or buy additional allowances from other companies.

The pollutant emissions are modeled by different functions depending on the generating units state (in operation and starting-up and shutting-down transition states). The atmospheric pollution such as sulphur oxide (SO_x), nitrogen oxide (NO_x) and carbon dioxide (CO_2) caused by burning fossil fuel is usually modelled by a quadratic function,

$$E_j(y_{t,j}) = \alpha_j \cdot (y_{t,j})^2 + \beta_j \cdot y_{t,j} + \gamma_j, \tag{2.4}$$

where $\alpha_j, \beta_j, \gamma_j$ are the emission coefficients of unit j [6, 129, 126, 136].

The start-up emissions may depend on the number of periods that a unit has been down or may be constant. Shutdown emissions are typically represented by a constant or disregarded. For the sake of simplicity, let us consider constant start-up emissions and disregard the shutdown ones. Thus, the total emission of atmospheric pollutants is expressed as

$$E(y,u) = \sum_{t=1}^{T} \left(\sum_{j=1}^{N} \{E_j(y_{t,j}) \cdot u_{t,j} \right. \qquad (2.5)$$
$$\left. + Se_{t,j} \cdot (1 - u_{t-1,j}) \cdot u_{t,j}\} \right),$$

where $Se_{t,j}$ is the start-up atmospheric pollutant emissions of unit j at time period t and $u_{t,j}$ is the satus of unit j at time period t (1 if the unit is on; 0 otherwise).

2.2.3 Constraints

There are two types of constraints. On the one hand, the power system must satisfy the customers and, on the other hand, the generating units are subject to working restrictions.

System constraints

Power systems must satisfy customer demand at all times, thus the power produced must equate demand. Furthermore, since uncertainty is not being explicitly considered, one must ensure the ability of quickly generating additional power. Therefore, load demand and spinning reserve constraints must be imposed on the system.

Load demand constraints: impose that the total power generated at each time period must meet the demand.

Spinning reserve constraints: impose that the generating units leave a certain amount of reserve, i.e., unused production capacity, at each time period. Then, in the case of a demand spike or equipment failure, the generators will not be completely used out and thus will have some capacity to ramp up to produce the extra required power.

Generating unit constraints

The generating units impose some other constraints regarding their characteristics and physical restrictions. These include unit output capacity, output variation, and minimum number of time periods that the unit must be in each state (on or off).

Output range constraints: ensure that production of each generating unit is bounded by the unit minimum and maximum production capacity. Unit limits may be due to either economic or technical reasons.

Ramp up and ramp down constraints: ensure that production variations in consecutive periods are limited. Due to the thermal stress limitations and some mechanical characteristics of the generating units, it is not possible to have fast variations on power production. Therefore, the output generation level variation between two consecutive periods is limited by maximum ramp-up and ramp-down rates.

Minimum uptime and downtime constraints: ensure that units remain at each state during a specified minimum number of time periods. Due to technical reasons, when a unit switches states, i.e., it is switched on or switched off, it must remain in the new state at least a certain number of pre-specified time periods.

Chapter 3

A Genetic Algorithm approach to the Unit Commitment Problem: Single-objective case

In this chapter, a hybrid genetic algorithm is proposed to address the Unit Commitment (UC) problem. It should be reminded that in the UC problem, one wishes to schedule a subset of a given group of electrical power generation units and also to determine their production output in order to meet energy demands at minimum cost. In addition, the solution must satisfy a set of technological and operational constraints.

The algorithm developed is an Hybrid Biased Random Key Genetic Algorithm (Hybrid BRKGA). The biased random key technique was chosen given its reported good performance on many combinatorial optimization problems. In the algorithm, solutions are encoded using random keys, which are represented as vectors of real numbers in the interval $[0,1]$. The GA proposed is a variant of the random key genetic algorithm, since bias is introduced in the parent selection procedure as well as in the crossover strategy. The BRKGA is hybridized with local search in order to intensify the search close to good solutions.

Tests have been performed on benchmark large-scale power systems with up to 100 units for a 24 hours period. The results obtained have shown the proposed methodology to be an effective and efficient tool for finding solutions to large-scale UC problems. Furthermore, from the comparisons made it can be concluded that the results produced improve upon the solutions obtained by reported state-of-the-art methodologies.

3.1 Introduction

As it was already referred the power systems are one of the most important infrastructures of a country since the commodity involved is essential to everyday life, its availability and price are critical to many companies, and it requires continuous balancing [91, 92]. The study and operation of these systems involves solving many different optimization problems [59]. Amongst these problems, the Unit Commitment (UC) problem stands out by playing a key role in planning and operating power systems. Optimal scheduling of the generation units, not only have the potential of saving millions of euros, but also of maintaining system reliability by keeping a proper spinning reserve [135]. The UC problem is an optimization problem where one wishes to determine the on/off status of the generation units at minimum operating costs. In addition, the production of the committed units, which also has to be determined, must be such that it satisfies demand and spinning reserve constraints. Furthermore, a large set of technological constraints are also imposed on generation units. Due to its combinatorial nature, multi-period characteristics, and nonlinearities, this problem is highly computational demanding and, thus, it is a hard optimization task to solve it for real sized systems. The UC problem has been extensively studied in the literature. Several methodologies, based on exact and on approximate algorithms have been reported. Optimal solutions can only be obtained for small sized problem instances, through the solutions of the corresponding Mixed Integer Quadratic Programming (MIQP) model.

Other versions of the UC problem have also been studied, see e.g. [135, 123, 57, 101].

In the past, several traditional heuristic approaches have been proposed, based on exact methods such as Dynamic Programming, Branch and Bound, Lagrangian Relaxation, and other for Mixed-Integer Programming, see e.g. [79, 21, 53, 94, 40, 41, 85]. Most of the recently developed methods are metaheuristics, evolutionary algorithms, and hybrids of the them, see e.g. [133, 107, 22, 56, 64, 95, 48]. These latter types have, in general, lead to better results than the ones obtained with the traditional heuristics.

In this chapter, a Hybrid Biased Random Key Genetic Algorithm (HBRKGA) is proposed to address the UC problem. The HBRKGA proposed here is based on the framework provided by [47], which has been used in other important applications in an effective and efficient way [36, 45, 106, 46, 37, 62]. BRKGAs are a variation of the random key genetic algorithms, first introduced by [9]. A Biased Random Key GA differs from a random key GA in the way parents are selected for mating and also on the probability of inheriting chromosomes from the best parent. In our HBRKGA, we also include repair mechanisms and thus, all the individuals considered for evaluation are feasible. The HBRKGA is capable of finding better solutions than the best currently known ones for most of the benchmark problems solved. Furthermore, the computational time requirements are modest and similar to those of other recent approaches.

The remaining of this chapter is organized as follows. In subsection 3.2, the UC problem mathematical formulation is given. In subsection 3.3, a description of previous methodologies addressing the UC problem is carried out. The solution approach proposed to address the UC problem is explained in subsection 3.5. Due to recent advances in MIQP commercial solvers, such as CPLEX, it is possible to solve UC problems optimally, at least of smaller sizes. Therefore, in subsection 3.6 the UC problem is reformulated as a mixed integer quadratic model. Then, in subsection 3.7 the effectiveness and efficiency of our approach is tested on benchmark systems with up to 100 units for a 24-hour period. In addition, the results obtained are compared to those of the current state of the art approaches reported in the literature and, for small size instances, with the ones obtained by a commercial solver. Finally, some conclusions are drawn in subsection 3.8.

3.2 Unit Commitment Single Objective Problem Formulation

In the Unit Commitment problem the optimal turn-on and turn-off schedules need to be determined over a given time horizon for a group of power generation units under some operational constraints. In addition, the output levels must be decided for each on-line unit at each time period. The model has two types of decision variables. Binary decision variables $u_{t,j}$, which are either set to 1, meaning that unit j is committed at time period t; or otherwise are set to zero. Real valued variables $y_{t,j}$, which indicate the amount of energy produced by unit j at time period t. Such decisions are limited by two types of constraints: load constraints, consisting of demand and spinning reserve constraints; and technological constraints. The objective of the UC problem is the minimization of the total operating costs over the scheduling horizon.

3.2.1 Objective Function

The objective function has three cost components: generation costs, start-up costs, and shut-down costs. The generation costs, also known as the fuel costs, are conventionally given by the following quadratic cost function.

$$F_j(y_{t,j}) = a_j \cdot (y_{t,j})^2 + b_j \cdot y_{t,j} + c_j, \tag{3.1}$$

where a_j, b_j, c_j are the cost coefficients of unit j.

The start-up costs, that depend on the number of time periods during which the unit has been off, are given by

$$S_{t,j} = \begin{cases} S_{H,j}, & \texttt{if } T^{off}_{min,j} \leq T^{off}_{j}(t) \leq T^{off}_{min,j} + T_{c,j}, \\ S_{C,j}, & \texttt{if } T^{off}_{j}(t) > T^{off}_{min,j} + T_{c,j}, \end{cases} \tag{3.2}$$

where $S_{H,j}$ and $S_{C,j}$ are the hot and cold start-up costs of unit j, respectively. The shut-down costs for each unit Sd_j, whenever considered in the literature, are not time dependent.

Therefore, the cost incurred with an optimal scheduling is given by the minimization of the total costs for the whole planning period,

$$\text{Minimize} \quad \sum_{t=1}^{T}\sum_{j=1}^{N} \left(F_j(y_{t,j}) \cdot u_{t,j} + S_{t,j} \cdot (1-u_{t-1,j}) \cdot u_{t,j} + Sd_j \cdot (1-u_{t,j}) \cdot u_{t-1,j}\right). \tag{3.3}$$

3.2.2 Constraints

The constraints are divided into two sets: the demand constraints and the technical constraints. The first set of constraints can be further divided into load requirements and spinning reserve requirements.

1) Load Requirement Constraints: The total power generated must meet the load demand, for each time period.

$$\sum_{j=1}^{N} y_{t,j} \cdot u_{t,j} \geq D_t, t \in \{1,\dots,T\}. \tag{3.4}$$

2) Spinning Reserve Constraints: The spinning reserve is the total amount of real power generation available from on-line units net of their current production level.

$$\sum_{j=1}^{N} Ymax_j \cdot u_{t,j} \geq R_t + D_t, t \in \{1,\dots,T\}. \tag{3.5}$$

The second set of constrains includes limits on the unit output range, on the maximum output variation allowed for each unit (ramp rate constraints), and on the minimum number of time periods that the unit must be continuously in each status (on-line or off-line).

3) Unit Output Range Constraints: Each unit has a maximum and minimum production capacity.

$$Ymin_j \cdot u_{t,j} \leq y_{t,j} \leq Ymax_j \cdot u_{t,j}, \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,...,N\}. \tag{3.6}$$

4) Ramp rate Constraints: Due to the thermal stress limitations and mechanical characteristics the output variation levels of each on-line unit for consecutive periods are restricted by ramp rate limits.

$$-\Delta_j^{dn} \leq y_{t,j} - y_{t-1,j} \leq \Delta_j^{up}, \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,...,N\}. \tag{3.7}$$

5) Minimum Uptime/Downtime Constraints: The unit cannot be turned on or turned off instantaneously once it is committed or decommitted. The minimum uptime/downtime constraints impose a minimum number of time periods that must elapse before the unit can change its status.

$$T_j^{on}(t) \geq T_{min,j}^{on} \text{ and } T_j^{off}(t) \geq T_{min,j}^{off}, \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,...,N\}. \tag{3.8}$$

3.3 Previous methodologies addressing the UC problem

In this subsection, we start by describing several traditional heuristic approaches based on exact methods that, in the past, have been reported in the literature. Then, we describe methods based on metaheuristics, mainly evolutionary algorithms, and hybrids of the them, which more recently have been reported in the literature.

Dynamic Programming (DP) was the earliest optimization-based method to be applied to the UC problem. The advantage of DP is its ability to maintain solution feasibility.

The disadvantage is the curse of dimensionality, which may result in unacceptable computational time and memory requirements. Due to the enumerative nature of the dynamic programming, it suffers from a long processing time that expands exponentially with the size of the problem. Thus, only small sized problems can be solved. Therefore, in practice many heuristic strategies have been introduced to limit the dynamic search for a large system. The most widely used method to reduce the dimension is based on a priority list. The list is typically formed by ranking the units based on their marginal power production cost or average full load cost index [102]. More recently, other approximate methods based on DP have been proposed for the UC problem and its variants. For instance, the authors in [94] have proposed a DP algorithm based on linear relaxation of the on/off status of the units and on sequential commitment of units one by one for the UC in multi-period combined heat and power production planning under the deregulated power market. In [85] a DP technique with a fuzzy and simulated annealing based unit selection procedure has been proposed. The computational requirements are reduced by minimizing the number of prospective solution paths to be stored at each stage of the search procedure through the use of heuristics, such as priority ordering of the units, unit grouping, fast economic dispatch based on priority ordering, and avoidance of repeated economic dispatch. Not many works on the UC problem make used of Branch-and-Bound (BB). In the earlier ones [65, 21], the authors address the UC problem with time-dependent start-up costs, demand and reserve constraints and minimum up and down time constraints. However the authors do not incorporate ramp rate constraints. Furthermore, [21] consider that the fuel consumption is given by a linear cost function, which constitutes another major drawback. In [53] a two-phase procedure is proposed. In the first phase, through the constraint satisfaction techniques, the constraints are propagated as much as possible to reduce the search domain. The second phase fulfills the economic dispatch function on the committed units, obtaining an upper bound. Lagrangian Relaxation (LR) is capable of solving large scale UC problems in a fast manner, however the solutions obtained are, usually, suboptimal. Based on the LR approach, the UC problem can be written in terms of 1) a cost function that is the sum of terms each involving a single unit, 2) a set of constraints involving a single unit, and 3) a set of coupling constraints involving all of the units (the generation and reserve constraints), one for each hour in the study period. An approximate solution to this problem can be obtained by adjoining the coupling constraints onto the cost function by using Lagrange multipliers. The resulting relaxed problem is to minimize the so-called Lagrangian subject to the unit constraints. LR was first applied to solve the UC problem without considering ramp constraints [79]. [8] uses LR to disaggregate the model into separate subproblems, one for each unit. The author tests the method on a 10-units system with exponential start-up costs, see case study 5. Recently, in [40] an effective Lagrangian relaxation approach for the UC problem has been proposed. This approach relies on an exact algorithm for solving the single-unit commitment problem proposed in [39]. More recently, in [32] two Lagrangian relaxation methods are proposed: one based on subgradient optimization and the other based on cutting planes. They were tested on several problem instances generated by the authors with a simpler and linear cost function, but not on the usual benchmark ones. Therefore, no comparisons with alternative methods were possible. From the tests performed, it was concluded that the subgradient method yields better results. By solving the MIQP model, optimal solutions can be found, but the computational time requirements are enormous and, usually, increase exponentially with the problem size, even with the availability of efficient software packages (such as CPLEX and LINDO), as will be seen in the results section. Some authors have tried to improve the performance of the MIQP by reformulating the UC problem as a mixed integer linear programming problem by means of piece-wise linear approximations of the cost function, see e.g., [40, 41, 120].

Regarding methods based on metaheuristics, there is recent literature reporting results on evolutionary programming [58], particle swarm optimization [133], quantum evolutionary algorithms [56, 64], memetic algorithms [117], and genetic algorithms [60, 5, 107, 22, 95]. [58] employs evolutionary programming in which populations of individuals are evolved through random changes, competition, and selection. The UC schedule is coded as a string of symbols and viewed as a candidate for reproduction. Initial populations of such candidates are randomly produced to form the basis of subsequent generations. [133] introduce an improved particle swarm optimization (IPSO) with adoption of the orthogonal design for generating the initial population scattered uniformly over a feasible solution space. This method has been tested on the problems of case study 1 with good results, recently outperformed by [56, 64]. In these works, Quantum-inspired Evolutionary Algorithms (QEAs) are proposed. The QEA is based on the concept and principles of quantum computing, such as quantum bits, quantum gates and superposition of states. QEA employs quantum bit representation, which has better population diversity compared to other representations used in evolutionary algorithms, and uses quantum gates to drive the population towards the best solution. The mechanism of QEA can inherently treat the balance between exploration and exploitation, thus incorporating a sort of local search. [56, 64] divide the UC problem into two subproblems: 1) schedule the on/off status of the units and 2) determine the power output of the committed units. In both works, repair mechanisms are used to accelerate the solution quality and to ensure that unit schedules generated by QEA are feasible. [56] improve the conventional QEA by introducing a simplified rotation gate for updating Q-bits and a decreasing rotation angle approach for determining the magnitude of the rotation angle. The current best known results for problems in case study 1 have been reported in these works, which we are able to improve.

A Memetic Algorithm (MA) and a Genetic Algorithm (GA) using local search combined with Lagrangian relaxation are introduced in [117]. In these algorithms, a local search is integrated as part of the reproductive mechanism. Results show that this approach can yield reasonable schedules at satisfactory computational times. Although it was used to solve problems in case studies 1 and 5, only for the latter it is competitive. GA solutions to the UC problem have been given in [60] with the addition of the problem specific operators. Problem specific operators are defined within windows, thus acting on building blocks rather than bits. Therefore, once a good building block is found it is preserved through the evolution process. [5] propose a GA using a repair mechanism, which was implemented in parallel. Ramp rate limits are always enforced while constructing the solutions, and therefore never violated. However, heuristics are used to enforce load feasibility (enough power is committed) and time feasibility (minimum up/down time). The proposed algorithm has been successfully applied to a real problem with 45 units, see case study 4. [22] also decompose the UC problem into the scheduling and dispatching problems. The former is solved by a GA using a floating-point chromosome representation. Since the encoding and decoding schemes are specific to and based on the load profile type, different problems require different such schemes. The production of each on-line unit is determined by LR. In [107] a real coded GA is proposed. A solution is represented by a real number matrix, representing the generation schedule for each unit at each time period. A repair mechanism is used to guarantee that the generation schedule satisfies system and unit constraints. The method was tested by using the most common benchmark problems (case study 1) and a 38-units problem (case study 2), being only competitive for the latter one. A very recent type of evolutionary algorithm, the Imperialist Competition Algorithm (ICA), has been applied to the UC problem in [48]. In it a population consists of a set of countries, all divided between imperialist countries and colonies, based on the imperialists power, which is inversely proportional to its cost function for a minimization problem. Then the colonies move toward their relevant imperialist and the position of the imperialists is updated if necessary. In the next stage, the imperialistic competition among the empires begins, and through this competition, the weak empires are eliminated. The imperialistic competition will gradually lead to an increase in the power of powerful empires and a decrease in the power of weaker ones, until just one empire remains. The authors tested their methodology on the most commonly used benchmark problems, see case study 1. However, as it can be seen in the results section they only improve upon literature results for the problem instance with 10 units. More details on these methods and other developed applications for the UC problem can be found in the extensive and comprehensive bibliographic surveys published over the years, see e.g. [102, 81, 82, 98].

3.4 Genetic Algorithms

Genetic algorithms is a global search heuristic, which generates solutions to optimization problems using techniques inspired by natural evolution, such as mutation, selection, and crossover [78, 50]. The GA's may be included in a broader class of the evolutionary algorithms (EA). In general, a population of the chromosomes (also called individuals or phenotypes) is taken and must evolve to include the best solutions. During the evolutionary process, each individual can be subject to selection, mutation and crossover genetic operators. Typically, solutions are represented in binary as strings of 0 and 1 values or using floating point representation, or even, as string of real-values [125]. In general, the initial population is randomly generated. In each generation, the fitness of every individual in the population is evaluated. Next, a proportion of the individuals in the current population with better fitness can be copied to the new population, another proportion are stochastically selected from the current population, and each individual genome is recombined and randomly mutated to form a new population. Furthermore, in some instances, it may occur premature convergence. The migration can be used to maintain the diversity of the solutions in the population and, this way, improve the performance of the genetic algorithm [3]. The migration consists of obtaining a specified proportion of the individuals for the new population, through randomly generation, as was the case for the initial population. The new population is then used in the next generation of the algorithm. The most commonly used genetic operators are the **selection**, **mutation** and **crossover**. The crossover allows to combine the parents alleles forming a new chromosome string that inherits characteristics from both parents [60]. The mutation is used to maintain genetic diversity from one generation of a population of individuals to the next.

At each generation: a given proportion of individuals in the current population is selected to reproduce the individuals of the next population, yielding a pool of the individuals. The **selection** of the individuals may be performed by random sampling or based on their fitness value where the best solutions are typically selected. Next, a pair of the parents are chosen from the pool previously selected obtaining a new individual (child) by crossover and mutation. This procedure is repeated until a proportion of the new child individuals is obtained. The **crossover** is a procedure from which more than one parent solutions reproduces a child solution. There are different crossover modalities, such as, the one-point crossover, two-point crossover and uniform crossover. In the single crossover point, the child is obtained in the following way: all allele values from beginning of chromosome to the crossover point are copied from one parent, the remains alleles values are copied from the second parent.

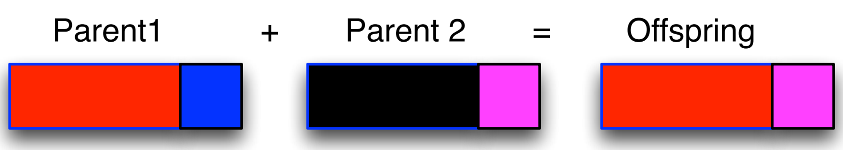

Figure 3.1: Single point crossover.

In the two-point crossover, the allele values from beginning of chromosome to the first crossover point are copied from one parent, the part from the first to the second crossover point is copied from the second parent and the remains alleles values are copied from the first parent.

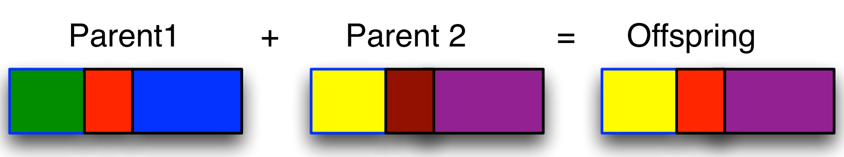

Figure 3.2: Two-point crossover.

In addition, it can be considered the uniform crossover where the child allele values in the string are randomly copied from the first or from the second parent with a fixed probability, typically 0.5.

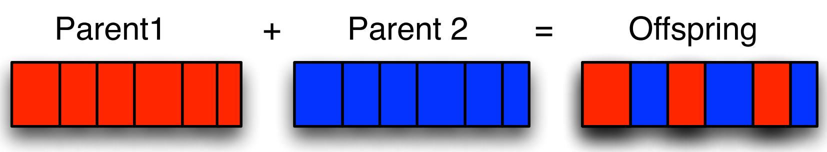

Figure 3.3: Uniform crossover.

In the **Mutation**, the chromosome is modified in one or more alleles values comparatively to its initial configuration. The correspondent problem solution may change abruptly from the previous solution. The mutation is repeatedly performed during evolution taking into consideration a specified mutation probability, which typically should be set low.

Example:

1 0 **0** 1 0 1 0 0

↓

1 0 **1** 1 0 1 0 0

Figure 3.4: Bit string mutation example.

The stopping criterium of the GA's is usually the maximum number of generations to be reached and previously specified.

In summary, the genetic algorithm comprises a genetic representation of the solution and their a fitness function, which allows to evaluate the solution. After to define genetic representation and the fitness function, it can be performed the initialization of the GA where the individuals are randomly generated to construct an initial population. Repetitively, in each generation, it is performed the application of the selection, mutation and crossover (genetic operators), improving the solutions in the population.

3.5 The proposed methodology

In recent years many heuristic optimization approaches have been developed, one of the most popular being GAs. Typically, GAs evolve a population of solutions as the result of selection, competition, and recombination. Crossover and mutation are used to maintain a diversity of the evolving population and thus, escape from local optima. Several GAs have been proposed for the UC problem, see e.g. [60, 5, 107, 22, 2]. As already said, GAs are a powerful stochastic global search technique as the search is performed by exploiting information sampled from different regions of the solution space [93]. Nevertheless, GAs usually do not perform well in fine-tuning near local optimal solutions because they use minimum a priori knowledge and fail to exploit local information. Local Search algorithms start with an initial solution and try to reach an optimal solution by means of small perturbations to the current solution, that is, the search is done within a pre-specified neighborhood. The inclusion of a Local Search procedure into a GA often leads to substantial improvement since the "local" improvement capabilities of the former are being combined with the "global" nature of the GA. GAs with random keys were first introduced by [9], for solving sequencing problems. In biased random key GAs, the bias is introduced at two different stages. On the one hand, when parents are selected, good solutions have a higher chance of being chosen, since one of the parents is always taken from a subset including the best solutions. On the other hand, the crossover strategy is more likely to choose alleles from the best parent to be inherited by offspring.

In this chapter, we propose a Hybrid Random Key Genetic Algorithm (HBRKGA), which is an improvement of the work in [95], based on the framework proposed by [47]. In here, we use improved decoding and repair mechanisms. The main reasons for using repair mechanisms are 1) to work on bounded search spaces (consisting of only feasible solutions) and 2) to avoid the problem of choosing penalties of different nature for each of the violated constraints [77]. In addition, and to intensify the search around good solutions, we have incorporated a local search procedure that, as it can be seen in the results section, has lead to better solutions. Chromosomes are represented as vectors of randomly generated real numbers in the interval $[0,1]$. The vector size N is given by the number of generating units. Each component of the vector corresponds to a priority that is to be assigned to each generation unit. The initial population consists of p vectors of N random keys, which are used by the decoder to generate feasible solutions, details are provided in subsection 3.5.1. Then, each solution is evaluated according to its corresponding total cost. Based on this cost, the population is divided into two subsets: the elite set, consisting of the best solutions, and the non-elite set, consisting of the remaining solutions. Solutions in the elite set are copied onto the next generation, which also consists of two other groups of solutions: solutions generated by crossover and new randomly generated solutions. Regarding the former they are obtained by reproduction between a parent taken from the elite solution set and a parent taken from the remaining solutions. Furthermore, the probability of inheriting alleles from the elite parent is higher than that of the other parent. The HBRKGA framework is illustrated in Figure 3.5, an adaptation from [47].

Specific to our problem are the decoding procedure, as well as the feasibility handling procedures. The decoding procedure, that is how solutions are constructed once a population of chromosomes is given, is performed in two main steps, as it can be seen in Figure 3.6. Firstly, a solution satisfying unit output range and ramp rate limits for each period is obtained. In this solution, the units are turned on-line according to their priority, which is given by the associated random key value. Furthermore, unit production is also set by random key value. The production values are chosen such that the ramp rate constraints and the output range constraints are satisfied. Then, these solutions are checked for the remaining constraints and repaired whenever necessary.

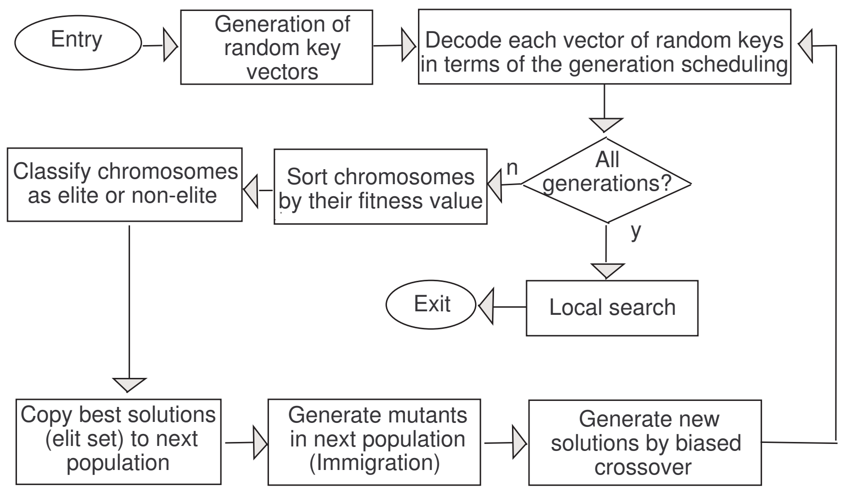

Figure 3.5: The HBRKGA adapted framework..

3.5.1 Decoding Procedure

The decoding procedure proposed here is based on that of [95]. The output generation levels are obtained based on the vector of random keys. Their values are computed such that the capacity limits and ramp rate limits are ensured during the decoding phase.

Given a vector of numbers in the interval $[0,1]$, say $RK = (r_1, r_2, ..., r_N)$, a rank vector $O = (O_1, O_2, ..., O_N)$ is computed. Each i^{th} component O_i is defined taking into account the descending order of the RK value, i.e., $O_i = \sum_{j=1}^{N} \delta(r_j - r_i)$, with

$$\delta(x) = \begin{cases} 1 & \texttt{if } x \geq 0, \\ 0 & \texttt{if } x < 0. \end{cases}$$

Then an output generation matrix Y is obtained, where each element $y_{t,j}$ gives the production level of unit j such that $O_j = i, i = 1, ..., N$ at time period $t = 1, ..., T$. This amount, which is proportional to the random key value, is guaranteed to be in the range defined by minimum and maximum allowed output limits and ramp rate limits, as follows:

$$y_{t,j} = Y_{t,j}^{min} + r_j . \left(Y_{t,j}^{max} - Y_{t,j}^{min} \right), \tag{3.9}$$

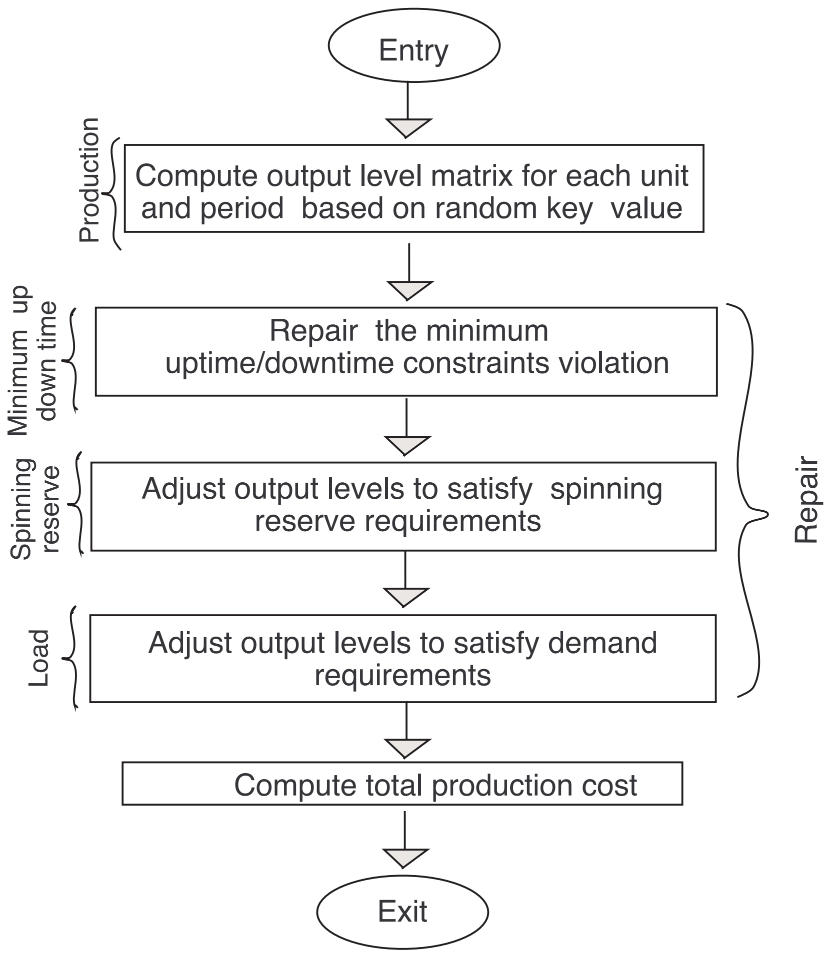

Figure 3.6: Decoder flow chart.

where $Y_{1,j}^{max} = Ymax_j$, $Y_{1,j}^{min} = Ymin_j$. These limits are defined considering the unit output generation level limits and the ramp rate limits. The procedure is given in Algorithm 1.

At the same time that the ramp rate constraints are ensured for a specific time period t, new output limits ($Y_{t,j}^{max}$ and $Y_{t,j}^{min}$ upper and lower limits, respectively) must be imposed, for the following period $t+1$, since their value depends not only on the unit output limits but also on the output level of the current period t. Equation (3.10) show

Algorithm 1 Initial matrix generation output

$i = 1$

$d = D_t + R_t$

while $i \leq N$ and $d > 0$ **do**

  Find j such that $O_j = i$

  $y_{t,j} = Y_{t,j}^{min} + r_j.\left(Y_{t,j}^{max} - Y_{t,j}^{min}\right)$

  $d = d - Y_{t,j}^{max}$

  Next i

end while

how these values are obtained.

$$Y_{t,j}^{max} = min\left\{Ymax_j, y_{t-1,j} + \Delta_j^{up}\right\},$$
$$Y_{t,j}^{min} = max\left\{Ymin_j, y_{t-1,j} - \Delta_j^{dn}\right\}. \tag{3.10}$$

After computing the output generation matrix Y, with the production level of each unit j for each time period t, the generation schedule may not be admissible and therefore, the solution obtained may be infeasible. Hence, the decoding procedure also incorporates a repair mechanism. These repair mechanisms are described in the next section.

3.5.2 Handling infeasibilities

Since BRKGA is a generic search method, in the application of this GA to the constrained UC optimization problem we also include a mechanism to repair UC solutions. The repairing procedure transforms what would be an infeasible solution into a feasible solution. The main reason for using a repair technique in Genetic Algorithms is the reduction of the search space to feasible solutions. Although penalty functions are the simplest and most commonly used methods for handling constraints in Evolutionary Algorithms (EAs), they have some limitations. The main drawback is that penalty factors which determine the severity of the penalization, must be set by the user and their values are problem dependent [77]. Extensive experimentation is needed to define appropriate parameters [76]. So, in the implementation of the BRKGA algorithm we use an efficient solution repair technique. However, in subsection 3.7.2 we study the effect of using penalty functions in the BRKGA implementation for the UC problem solution. The fitness function used in BRKGA with penalty factors has two terms, the spinning reserve and minimum up/down time constraint violation penalties. The first term is computed from the objective function normalized, which in turn is composed by the total production costs

$$f(x) = \sum_{t=1}^{T}\sum_{j=1}^{N}\left(F_j(y_{t,j})\cdot u_{t,j} + S_{t,j}\cdot(1-u_{t-1,j})\cdot u_{t,j} + Sd_j\cdot(1-u_{t,j})\cdot u_{t-1,j}\right),$$

(see equation 3.3), while the constraint violations are penalized by second term. The main steps in calculation the fitness measures beginning with the maximum and minimum values of the objective function in the population, $f_{max} = max_x f(x)$, $f_{min} = min_x f(x)$, from which, are obtained the normalized objective function for each individual (random key),

$$\tilde{f}(x) = \frac{f(x) - f_{min}}{f_{max} - f_{min}}.$$

The constraint violations $\tilde{g}_1(x)$ and $\tilde{g}_2(x)$ of individual x are calculated as the summation of the normalized violations of spinning reserve and minimum up/down time constraints, respectively:

$$\tilde{g}_i(x) = \frac{c_i(x) - c^i_{min}}{c^i_{max} - c^i_{min}}$$

where $c^i_{min} = max_x c_i(x)$ and $c^i_{max} = min_x c_i(x)$ represents the maximum and minimum values of each constraint violation in the population, respectively. Here, the spinning reserve constraint violation is given by: $c_1(x) = \sum_{t=1}^{T} c_{1,t}(x)$ with

$$c_{1,t}(x) = max\left\{0, D_t + R_t - \sum_{j=1}^{N} y_{t,j}\cdot u_{t,j}\right\},$$

while the minimum up/down time constraint is obtained as follows: $c_2(x) = \sum_{j=1}^{N} c_{2,j}(x)$ where

$$c_{2,j}(x) = \sum_{t=1}^{T} max\left\{0, -\left(T_j^{on}(t-1) - T_{min,j}^{on}\right) \cdot (u_{t-1,j} - u_{t,j})\right\} +$$
$$\sum_{t=1}^{T} max\left\{0, -\left(T_j^{off}(t-1) - T_{min,j}^{off}\right) \cdot (u_{t,j} - u_{t-1,j})\right\},$$

is the amount of the minimum up/down time constraint violations of the j^{th} unit. Although the constraints are expressed in different units, the normalization prevents any sort of bias toward of the constraints violation. Thus, no penalty factors are required.

Using the approach proposed by [23], each individual is evaluated as in equation 3.11,

$$Fitness(x) = \begin{cases} \tilde{f}(x) & \texttt{if feasible} \\ \tilde{f}_{worst} + \tilde{g}_1(x) + \tilde{g}_2(x) & \texttt{otherwise} \end{cases}, \tag{3.11}$$

where $\tilde{f}_{worst}$ is the objective function value of the worst feasible solution in the population. If all solutions are infeasible in the population, then $\tilde{f}_{worst}$ is set to 1 as in [86].

The repair mechanism starts by ensuring that minimum up/down time constraints are satisfied. The adjustment of the unit status is obtained using the repair mechanism illustrated in Figure 3.7. As it can be seen, for two consecutive periods the unit status can only be changed if the $T_{min}^{on/off}$ is already satisfied, for a previously turned on or turned off unit, respectively.

For each period, it may happen that the spinning reserve requirements are not satisfied. If the number of on-line units is not enough, some off-line units are turned on, one at the time, until the cumulative capacity matches or is larger than $D_t + R_t$, as shown in Figure 3.8. In doing so, units are considered in descending order of priority, i.e., random key value. After ensuring the spinning reserve satisfaction, it may happen that we end up with excessive spinning reserve. Since this is not desirable due to the

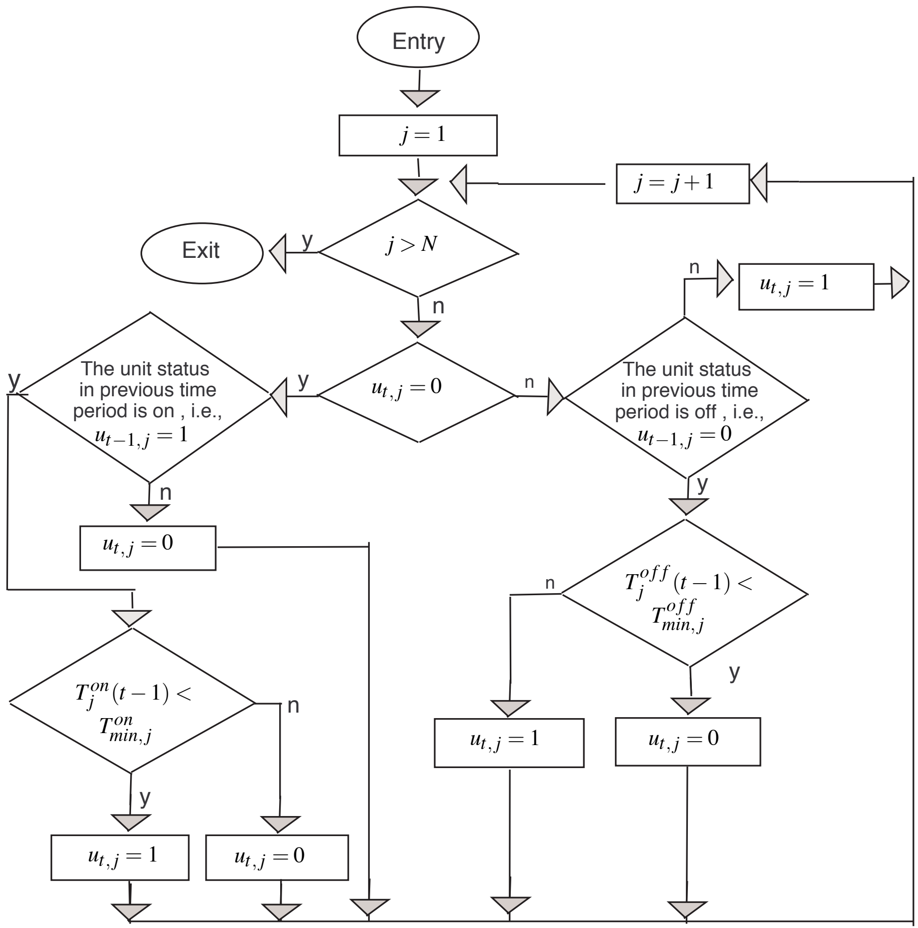

Figure 3.7: Flowchart of Minimum up down time repair algorithm.

additional operational costs involved, we look for units that can be decommited. Units are considered for turning off-line in ascending order of priority. At the end of this procedure we have found the U matrix, specifying which units are being operated at each time period, and the Y matrix, which indicates how much each on-line unit is producing. All constraints are satisfied except, may be, the load demand. Nevertheless, the maximum and minimum allowed production limits can be directly inferred from matrix Y. Therefore, we must adjust the total production to satisfy load demand for each time period. Firstly, for all on-line units the production is set to its minimum

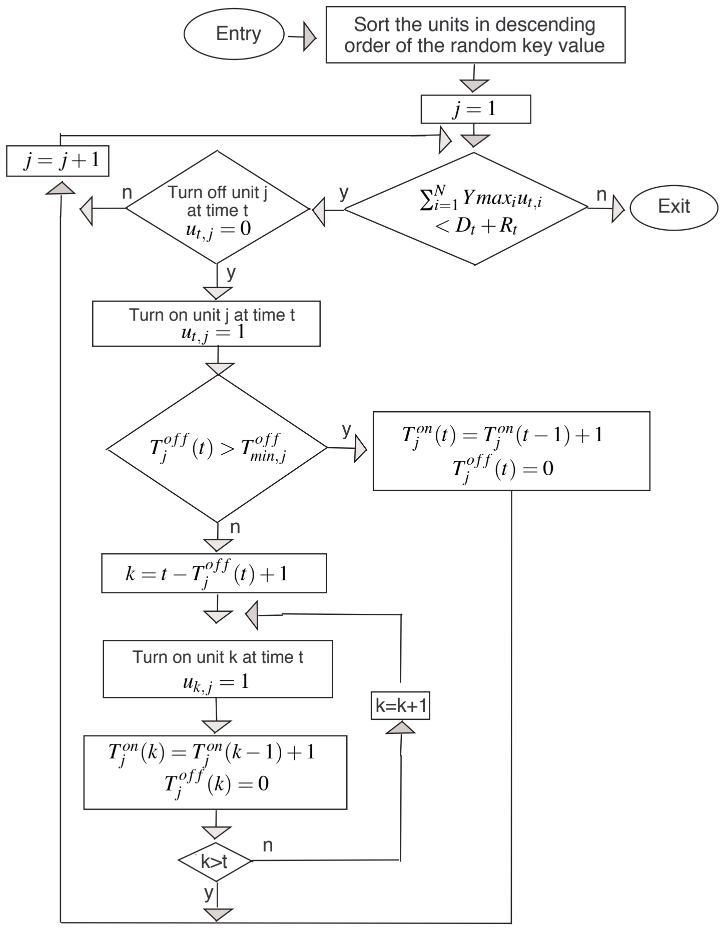

Figure 3.8: Handling spinning reserve constraint.

allowed value. Next, for each time period, each unit is set to its maximum allowed production, one at the time, until the production reaches the load demand value. In doing so, units are considered in descending order of priority. This is repeated no more than N times. It should be noticed that by changing production at time period t the production limits at time period $t+1$ change, and hence these new values, which are obtained as in equation (3.10), must be satisfied. Once these repairing procedures have been performed, the feasible solution obtained is evaluated through its respective total cost.

3.5.3 GA Configuration

To obtain a new population of solutions, we join 3 subsets of solutions obtained as follows:

- Copied Solutions: 20% of the best solutions of the population of the current generation (elite set) are copied onto the next generation;
- Mutants: 20% of the solutions of the population of the next generation are obtained by randomly generating new solutions.
- Offspring Solutions: 60% of the solutions of the population of the next generation are obtained by biased reproduction, which is achieved by using both a biased parent selection and a biased crossover probability.

As said before, the biased reproduction is accomplished by using both a biased parent selection and a biased crossover. Biased parent selection is performed by randomly choosing one of the parents from the elite set and the other parent from the remaining solutions. This way, elite solutions are given a higher chance of mating, and therefore of passing on their characteristics to future populations. Regarding the biased crossover, we consider a biased coin which is tossed to decide on which parent to take the gene from. Since the coin is biased, the offspring inherits the genes from the elite parent with higher probability (0.7 in our case).

3.5.4 Local Search

Another improvement to our previous work [95] is the inclusion of a local search procedure. At the end of HBRKGA we use a local search procedure to try to improve the solutions in the final elite set. This mechanism, which is illustrated in Figure 3.9, is a 2-swap procedure, where an on-line generation unit is replaced by an off-line generation unit, if the swap is feasible and leads to a lower cost. Given a solution (in the elite set) we build two sets of generation units. On the one hand, we build a set S_{on} containing the on-line generation units that can be turned off; on the other hand, we build a set S_{off} containing the off-line units that can be turned on.

For each time period, we pick-up a pair of units, one from each of the sets built, and analyze the feasibility of the swap. If the swap is feasible, we compare the total cost of the new solution with that of the current solution. If an improvement can be achieved, the swap is performed resulting in a better solution; otherwise the swap is discarded. In both cases, we move on and try the next swap using the previously built sets, i.e. no update to the sets S_{on} and S_{off} is performed. The 2-swap strategy is repeatedly performed until all swaps have been tried. The procedure is applied to all solutions in the elite set. The contribution of the local search to the global solution quality can be seen in the results provided Section 3.7.

3.6 Mixed integer quadratic programming

The UC problem can be casted as a mixed-integer nonlinear program (MINLP). Despite the ever-increasing availability of cheap computing power and the advances in off-the-shelf software for MINLP, solving (UC) by general-purpose software, even using the most advanced approaches available, is not feasible when the number of units and/or the length of the time horizon becomes large, [40].

Here and for comparison purposes we formulate the UC problem as a Mixed Integer Quadratic Programming (MIQP) problem and solve it using the commercial software CPLEX. To do so, we simply pass the MIQP formulation given below to CPLEX.

In order to formulate the UC problem as a MIQP model we need to introduce the following auxiliary binary variables:

$l_{t,j}$: indicates whether unit j has been started-up or not at time period t (1 if it has been started-up; 0 otherwise);

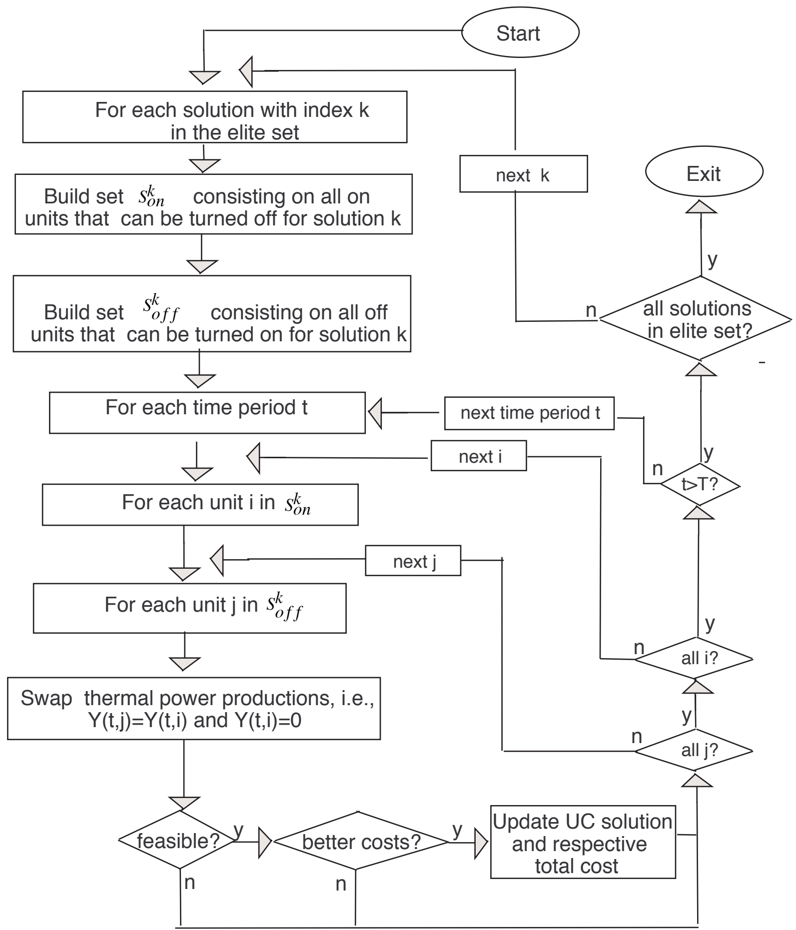

Figure 3.9: Flow chart of local search.

$h_{t,j}$: indicates the cold status of the off-line unit j at time t (1 if the unit is cold; 0 otherwise);

$v_{t,j}$: indicates wether unit j has had a cold start-up or not at time period t (1 if it had; 0 otherwise).

The objective function is now rewritten as

$$\text{Minimize} \quad \sum_{t=1}^{T}\left(\sum_{j=1}^{N}\left\{a_j \cdot (y_{t,j})^2 + b_j \cdot y_{t,j} + c_j \cdot u_{t,j} + S_{H,j} \cdot l_{t,j} + (S_{C,j} - S_{H,j}) \cdot v_{t,j}\right\}\right).$$

As before several constraints must be satisfied. The power balance, the spinning reserve, the minimum and maximum production capacity and the ramp rate constraints are express as before, see equations (3.4) to (3.7) in Subsection 3.2.

The minimum up time constraints are nonlinear and thus are reformulated as given in equation (3.12).

$$\sum_{k=t}^{t_{max,j}} u_{k,j} \geq (u_{t,j} - u_{t-1,j}) \cdot t_{s,j}, \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,2,...,N\}. \tag{3.12}$$

where

$$t_{s,j} = \begin{cases} min\left\{T^{on}_{min,j}, T-t+1\right\}, & \texttt{if } t>1 \texttt{ or } (t=1 \texttt{ and } I_0(j)<0), \\ max\left\{0, T^{on}_{min,j} - I_0(j)\right\}, & \texttt{if } t=1 \texttt{ and } I_0(j)>0, \end{cases}$$

$$t_{max,j} = \begin{cases} min\left\{t+t_{s,j}-1, T\right\}, & \texttt{if } t_{s,j}>0, \\ T, & otherwise. \end{cases}$$

and $I_0(j)$ is the initial status of the unit j. The minimum down time constraints are also nonlinear and thus are reformulated as given in equation (3.13).

$$\sum_{k=t}^{t_{max,j}} \left(1-u_{k,j}\right) \geq (u_{t-1,j} - u_{t,j}) \cdot t_{s,j}, \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,2,...,N\}. \tag{3.13}$$

where

$$t_{s,j} = \begin{cases} min\left\{T^{off}_{min,j}, T-t+1\right\}, & \texttt{if } t>1 \texttt{ or } (t=1 \texttt{ and } I_0(j)>0), \\ max\left\{0, T^{off}_{min,j} + I_0(j)\right\}, & \texttt{if } t=1 \texttt{ and } I_0(j)<0, \end{cases}$$

$$t_{max,j} = \begin{cases} min\left\{t+t_{s,j}-1, T\right\}, & \texttt{if } t_{s,j}>0, \\ T, & otherwise. \end{cases}$$

Given the newly defined variables, we need to define the following new constrains:

$$l_{t,j} \geq u_{t,j} - u_{t-1,j}, \text{ for } t \in \{1,2,...,T\} \text{ and } j \in \{1,2,...,N\}. \tag{3.14}$$

$$v_{t,j} \geq l_{t,j} + h_{t-1,j} - 1, \text{ for } t \in \{1,2,...,T\} \text{ and } j \in \{1,2,...,N\}. \tag{3.15}$$

with

$$h_{0,j} = \begin{cases} 1, & \texttt{if } I_0(j) < 0 \text{ and } T^{off}_{min,j} + T_{c,j} < -I_0(j), \\ 0, & \texttt{if } I_0(j) > 0 \text{ or } T^{off}_{min,j} + T_{c,j} \geq -I_0(j) \end{cases}$$

$$h_{t,j} \geq 1 - \sum_{k=t_{min,j}}^{t} u_{k,j} - \delta(ta_j - t).\delta(ta_j + I_0(j) - t), \text{ for } t \in \{1,2,...,T\} \text{ and } j \in \{1,2,...,N\}. \tag{3.16}$$

where

$$\delta(x) = \begin{cases} 1, & \texttt{if } x > 0 \\ 0, & \texttt{if } x \leq 0, \end{cases}$$

and

$$t_{min,j} = \begin{cases} t - ta_j, & \texttt{if } t > ta_j \\ 1, & \texttt{otherwise} , \end{cases}$$

with $ta_j = T^{off}_{min,j} + T_{c,j} + 1$.

Constraints (3.14) guarantee that unit j has been started at time t only if it is on at time t and has been off at time $t-1$. In equation (3.15) it is assured that the cold start costs are only paid if unit j is cold and has been just started. Finally, constraints (3.16) state that unit j is cold at time t if and only if it has not been started for at least $T^{off}_{min,j}$ time periods.

CPLEX can be attractive in many situations since in addition to its robustness, it also allows for the incorporation of other constraints [40]. However, since for solving problems with integer variables CPLEX uses a Branch-and-Cut algorithm, it ends up solving a series of continuous subproblems. To these subproblems cuts must be added, on fractional-valued variables in the solution to the subproblems, in order to generate new subproblems with more restrictive bounds on the branching variables. Thus, a single mixed integer problem is decomposed into many subproblems. Therefore, even small sized problems require significant amounts of time and physical memory to be solved. Furthermore, CPLEX cannot cope with more general cost functions, such as, for example, exponential start-up costs, as is the case of the problems in case study 5, first proposed by [115] and [8].

3.7 Numerical Results

In this subsection, we report on the results obtained with the proposed HBRKGA. In addition, we also report on the results obtained without using the local search, here referred to as BRKGA. It should be noticed that the parameters are the same for both algorithms. Due to the stochastic nature of the methods proposed each problem was solved 20 times. Both GAs were implemented in Matlab. The proposed approaches have been tested on 5 different benchmark UC case studies. Some of the case studies include several problem instances, while others include only one. Amongst the case studies considered, we single out case study 1, since the problems in it are the ones that have been consistently considered in the literature and thus solved by many different methods and authors.

3.7.1 GA parameters setting

The present state-of-the-art theory on GAs does not provide information on how to configure the parameters involved in the algorithm. Therefore, the values used in our computational experiments have been taken from the guidelines provided in [30, 47], as well as, from past experience [95].

Computational experiments with different values for the crossover probability, the number of generations, and the population size were conducted on the problem with 40

Table 3.1: Average cost for the 40-unit system (case study 1) for different crossover probability values.

P_c	Best	Average	Worst	$\frac{Average-Best}{Best}\%$	$\frac{Worst-Best}{Best}\%$
0.5	2244466	2245409	2246047	0.04	0.07
0.6	2244312	2245388	2247529	0.05	0.14
0.7	2244345	2245350	2245775	0.04	0.06
0.8	2244347	2245432	2246957	0.05	0.12
0.9	2244354	2245476	2246566	0.05	0.10

generations units given in case study 1.

The biased crossover probability was tested on the range $0.5 \leq P_c \leq 0.9$ with a step size of 0.1, as suggested in previous work [95]. These 5 values were tried for 5 different populations sizes ($N_p = N$, $2N$, $3N$, $4N$, and $5N$).

Regarding the number of generations, it was set, for test purposes, to a sufficiently large number ($N_{Gers} = 20N$), which soon became apparent to be too large, and thus reduced to $10N$, see Figure 3.12. From these combinations we realized that a good compromise would be achieved for a $P_c = 0.7$ and $N_p = 2N$, as it can be seen in figures 3.10 and 3.11.

To illustrate the algorithm behavior in Table 3.1 we give the results obtained for varying P_c values with $N_p = 2N$ and $N_{Gers} = 10N$. We chose the value 0.7 since to this value corresponds the best performances with lower variability. (Note that a better best solution was found using 0.6.)

Regarding the population size N_p, as it can be seen from Figure 3.13, the solution quality is continuously and marginally improved with the population size while the

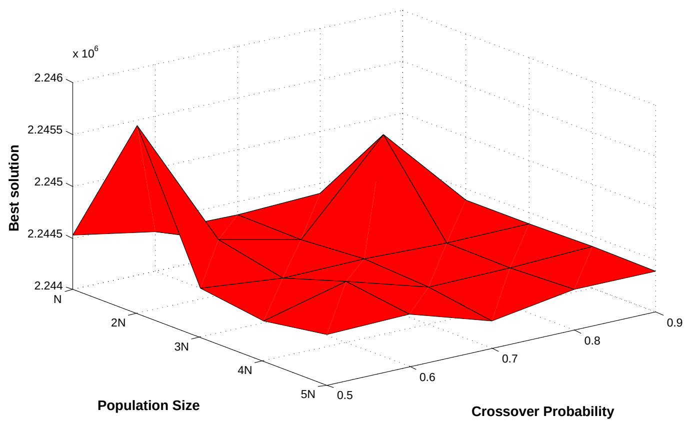

Figure 3.10: Best cost for the 40-unit system (case study 1) by crossover probability and population size.

computational time increase is almost linear. A trade-off analysis between the solution quality and the computation time lead us to set $N_p = 2N$.

In summary, we have set the number of generations to $10N$, the crossover probability to 0.7, and the population size to $2N$.

In the subsections 3.7.3 to 3.7.7 we compare the results obtained by BRKGA and HBRKGA with the best results reported in the literature. Furthermore, we have used CPLEX (version 12.1) to obtain an optimal solution and thus find out how close our results are to the optimum. Nevertheless, such comparisons are only possible for

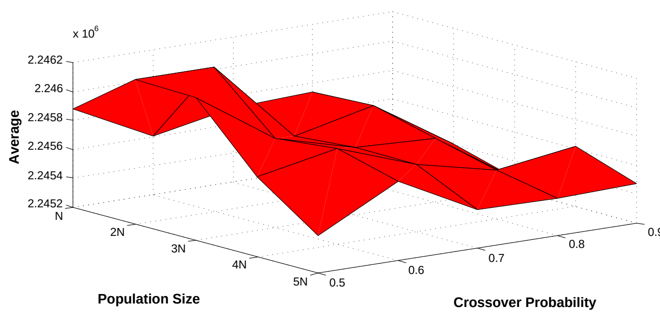

Figure 3.11: Average cost for the 40-unit system (case study 1) by crossover probability and population size.

small sized problems, since CPLEX is unable to solve larger problems due to the huge memory requirements. In addition, CPLEX cannot handle problems in case study 5, since the start-up costs are an exponential function of the number of hours that unit has been down.

3.7.2 Comparison of BRKGA with and without penalty function

As already said, the BRKGA algorithm incorporates a repair procedure to ensure that the population consists feasible solutions only. However, as discussed in Section 3.5.2,

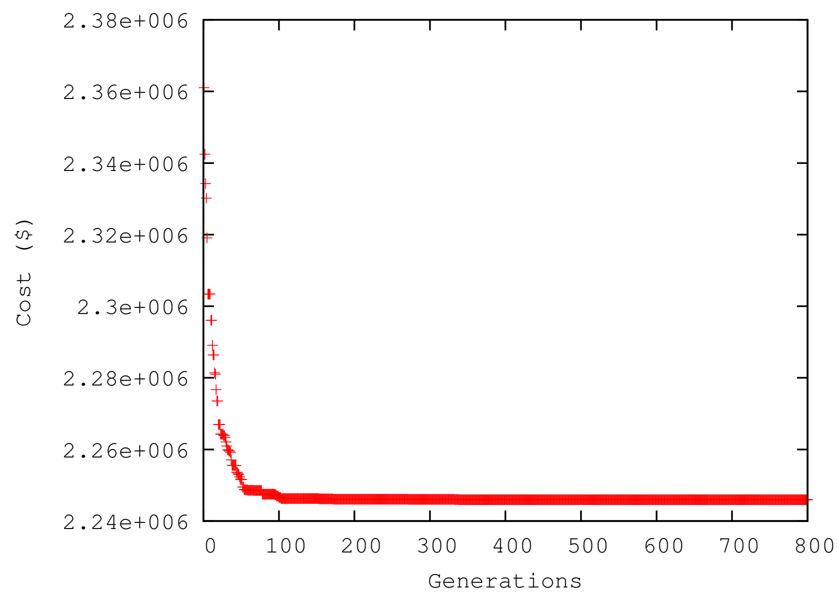

Figure 3.12: Average cost for the 40-unit system (case study 1) by generation.

many authors incorporate penalties into the fitness function to lead the search away from infeasible solutions; see [76, 107]. Thus, in this section we provide computational evidence regarding the better performance achieved by using the aforementioned repair mechanism.

Let us consider five variations of the GA proposed in this chapter, all considering penalty functions to address some of the possible violations. (See Section 3.5.2 for details on how to incorporate the penalty terms into the objective function.) In the first version $BRKGA_{P1}$ penalty functions are associated used both with the spinning reserve and minimum up/down time constraint violations, while the load constraints are ensured. Two other versions have been tested, $BRKGA_{P2}$ and $BRKGA_{P3}$ considering a penalty function regarding either the minimum up/down time or the spinning reserve violations, respectively. The remaining constraints are ensured by the corresponding repair mechanisms. For these 3 versions, the penalty strategy has been used for all generations. Finally, the other 2 versions, $BRKGA_{P4}$ and $BRKGA_{P5}$ are defined in the same manner as $BRKGA_{P2}$ and $BRKGA_{P3}$. The only difference being that for the later

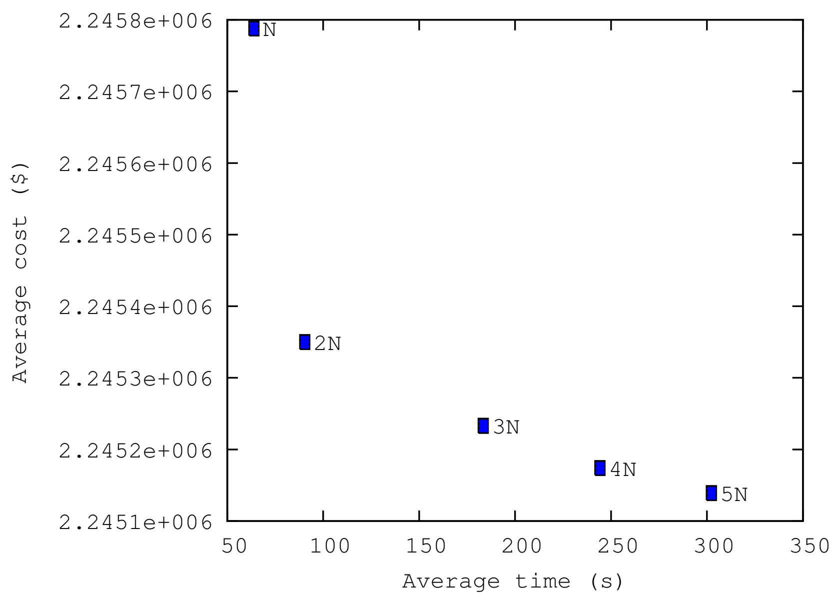

Figure 3.13: Average cost and computational time for the 40-unit system (case study 1) for different population sizes.

the penalty strategy is only applied in the first $\frac{N_{Gers}}{2}$ generations. For the remaining iterations, feasibility is enforced by applying the full repair procedure.

The simulation given in Appendix A. In Table 3.2 the best cost (if there are feasible solutions) or minimum number of violated constraints (if all solutions are unfeasible) and the execution time of the different BRKGA versions are reported. The results show that in all cases the BRKGA with repair procedure and without penalties in fitness function produces feasible solutions and better results for operating costs. This shows the BRKGA method superiority over the other possible versions using penalty function approach.

3.7.3 Case study 1

The HBRKGA and BRKGA have been tested on a set of often used benchmark problems, involving systems with 10 up to 100 generation units and considering, in each

Table 3.2: BRKGA performance using penalty functions.

	$BRKGA_{P1}$			$BRKGA_{P2}$			$BRKGA_{P3}$		
	Operation cost($)	Minimum number of violated constraints	Execution time (sec)	Operation cost($)	Minimum number of violated constraints	Execution time (sec)	Operation cost($)	Minimum number of violated constraints	Execution time (sec)
10	–	2	2.0	–	2	2.0	564248	0	2.1
20	–	4	9.5	–	4	9.5	1124664	0	17.8
40	–	9	79.1	–	9	83.4	2261312	0	90
60	–	8	153.9	–	8	184.7	3379080	0	188.9
80	–	14	259.8	–	14	301.1	4526756	0	321.5
100	–	15	929.9	–	15	1023.4	5662325	0	1050.3
	$BRKGA_{P4}$			$BRKGA_{P5}$			BRKGA		
	Operation cost($)	Minimum number of violated constraints	Execution time (sec)	Operation cost($)	Minimum number of violated constraints	Execution time (sec)	Operation cost($)	Minimum number of violated constraints	Execution time (sec)
10	564248	0	2.2	564248	0	2.2	564248	0	2.2
20	1169174	0	9.5	1169174	0	9.5	1124664	0	12.8
40	2245894	0	97.8	2261356	0	91.4	2244492	0	86.8
60	3365632	0	199.8	3379196	0	193.6	3365026	0	276.4
80	4487183	0	341.4	4522936	0	324.5	4486833	0	630.8
100	5607858	0	1094.8	5658642	0	1088.2	5607288	0	1258.7

case, a scheduling horizon of 24 hours. The 10 generation unit system, the base case, was originally proposed by [60].

Problem instances involving 20, 40, 60, 80 and 100 units are obtained by replicating the base case system and the load demands are adjusted in proportion to the system size. In all cases the spinning reserve is kept at 10% of the hourly demand. The start up costs have one of two possible values depending on the number of time periods the unit has been off, as given in equation (3.2). These values are different for each generation unit. The shut down costs are disregarded. Details of how these benchmark problems were constructed and on the system and demand data can be found in [60].

For the problems in this case study, CPLEX was able to find an optimal solution to

Table 3.3: Comparison of the best results obtained by the BRKGA and the HBRKGA with the best ones reported in literature, for problems of case study 1.

							CPLEX	HBRKGA		
Size	IPSO	IQEA	QEA	ICA	BRKGA	HBRKGA	MIQP	Rank	Gap(opt)	Gap(best)
10	563954	563977	***563938***	***563938***	564248	**563938**	563938	1st	0	0
20	1125279	1123890	***1123607***	1124274	1124664	1123955	1123297	3rd	0.06	0.03
40	2248163	*2245151*	2245557	2247078	2244492	**2244345***	2242634	1st	0.08	-0.04
60	3370979	*3365003*	3366676	3371722	3365026	**3363804**	–	1st	–	-0.04
80	4495032	*4486963*	4488470	4497919	4486833	**4485197**	–	1st	–	-0.04
100	5619284	*5606022*	5609550	5617913	5607288	**5605933**	–	1st	–	-0.002

* Recall that this is the best known solution, although it may not be an optimal solution.

systems involving 10 and 20 units. For problems with 40 units, we report on the best solution found by CPLEX before it crashed due to the excessive memory requirements. However, although the solution is not optimal, it is the best solution found so far. In tables 3.3 to 3.5 we compare the results obtained (best, average, and worst) with the best former results (in italic) obtained amongst the many publish methods. The best current solution, excluding the CPLEX one, is given in bold, for each of the problems. In the last column, we report on the gap between the HBRKGA solution and the previously best known solution. It should be noticed that whenever the HBRKGA produces a solution which is better than the best currently known solution the gap is negative. In Table 3.3 we also report on the optimality gap for the smaller problem instances, since for these we have the optimal solution value (provided by CPLEX). The results used for comparison purposes have been reported in: IPSO - [133]; IQEA - [56]; QEA - [64]; ICA - [48].

As it can be seen in Table 3.3, for all problem instances, except one, our best results improve upon the best previously known results. Moreover, for the problem instances for which an optimal solution has been found by CPLEX it can be seen that the HBRKGA has been able to find an optimal solution in one case, while in the other case the solution found is within 0.06% of optimality. By comparing the HBRKGA with the BRKGA,

Table 3.4: Comparison of the average results obtained by the BRKGA and the HBRKGA with the best average ones reported in literature, for problems in case study 1.

Size	IQEA	QEA	BRKGA	HBRKGA	HBRKGA Rank	HBRKGA Gap(%)
10	563977	***563969***	564445	564062	2nd	0.02
20	*1124320*	1124689	1124846	**1124213**	1st	-0.01
40	*2246026*	2246728	2245820	**2245350**	1st	-0.03
60	*3365667*	3368220	3366053	**3365201**	1st	-0.02
80	*4487985*	4490128	4488303	**4487620**	1st	-0.01
100	*5607561*	5611797	5607902	**5607024**	1st	-0.01

Table 3.5: Comparison of the worst results obtained by the BRKGA and the HBRKGA with the best worst ones reported in literature, for problems in case study 1.

Size	IPSO	IQEA	QEA	BRKGA	HBRKGA	HBRKGA Rank	HBRKGA Gap(%)
10	564579	***563977***	564672	565689	564737	4th	0.135
20	1127643	***1124504***	1125715	1126273	1125048	2nd	0.048
40	2252117	*2246701*	2248296	2246797	**2245775**	1st	-0.04
60	3379125	***3366223***	3372007	3367777	3366773	2nd	0.016
80	4508943	*4489286*	4492839	4489663	**4488962**	1st	-0.01
100	5633021	***5608525***	5613220	5609537	5608559	2nd	0.001

which already improves upon some of the previously known best solutions, we can see that the local search is always effective since the HBRKGA is always better than the BRKGA. And the improvement ranges from 0.007% to 0.063%. Although these values are small their impact may be very relevant given that they refer to a multi-million dollar industry.

Regarding the average results we have also improved on all but one of the problem

Table 3.6: Analysis of the variability of the solution quality for problems in case study 1.

Size	$\frac{Average-Best}{Best}\%$			$\frac{Worst-Best}{Best}\%$				St. deviation(%)		
	HBRKGA	IQEA	QEA	HBRKGA	IPSO	IQEA	QEA	HBRKGA	IQEA	QEA
10	0.02	0.0	0.005	0.14	0.11	0.0	0.13	0.03	0.0	0.02
20	0.02	0.04	0.09	0.1	0.21	0.05	0.19	0.03	0.01	0.06
40	0.04	0.04	0.05	0.06	0.18	0.07	0.12	0.02	0.02	0.02
60	0.04	0.02	0.05	0.09	0.24	0.04	0.16	0.02	0.01	0.03
80	0.05	0.02	0.04	0.08	0.31	0.05	0.1	0.02	0.01	0.02
100	0.02	0.03	0.04	0.05	0.24	0.04	0.07	0.01	0.01	0.02

instances solved, when compared with the best previously known results, see Table 3.4. In Table 3.5 similar results are reported for the worst solutions. Again, we improved upon the best previous results. The results reported in these tables also show that the local search incorporation is effective, since the HBRKGA improves upon the BRKGA.

Another important feature of the proposed algorithm is that, as it can be seen in Table 3.6, the variability of the results is quite small. The difference between the worst and best solutions found for each problem is always below 0.14%, while if the best and the average solutions are compared this difference is never larger than 0.05%. This allows for inferring the robustness of the approach, which is very important since the industry is reluctant to use methods with high variability as this may lead to poor solutions being used. When compared to the robustness of the alternative methods, it can be seen that it is better than that of the IPSO and QEA and almost the same as that of the IQEA.

Regarding the computational time, no exact comparisons may be done since, on the one hand, the values are obtained on different hardware; on the other hand, the HBRKGA reported time is real time and not CPU time and thus it is not directly comparable with others reported in the literature. Our computational experiments were performed on

Table 3.7: Analysis of the execution time, for problems in case study 1.

Size	IPSO	IQEA	QEA	ICA	BRKGA	HBRKGA	CPLEX
10	142	15	19	48	2	2	45
20	357	42	28	63	13	14	401
40	1100	132	43	151	87	90	1489
60	2020	273	54	366	276	301	–
80	3600	453	66	994	631	712	–
100	5800	710	80	1376	1259	1503	–

a Xeon X5450, 3.0 GHz and 4.0 GB RAM. This is a shared machine and therefore several jobs are usually running in parallel. Nevertheless, in Table 3.7 we report on our computational time requirements as well as on the ones of the works used for comparison purposes. It should be noticed that the results reported for the IPSO may not be accurate, since the authors only provide them in a graphical form. These results are also provided graphically in Figure 3.14. As it can be seen, the IPSO has computational time requirements much larger than the other methods. On the contrary, the QEA is the fastest method. The other three methods have a similar behavior in what concerns computational requirements. Therefore, the HBRKGA has an intermediate performance, regarding computational time, which does not seem to be a big price to pay for the increased solution quality. Recall that, as seen in Table 3.3, the HBRKGA provides the best solution for all but one of the problems analyzed in this case study.

When we analyze the computational time in a logarithmic scale, see the graph in Figure 3.15, a favorable conclusion regarding our algorithms can be drawn. The growth of all the other algorithms is closer to a line in the log scale, meaning that the time increase with problem size is closer to an exponential growth. In contrast, our algorithms have concave growth, in the log scale, meaning that the time increase is subexponential.

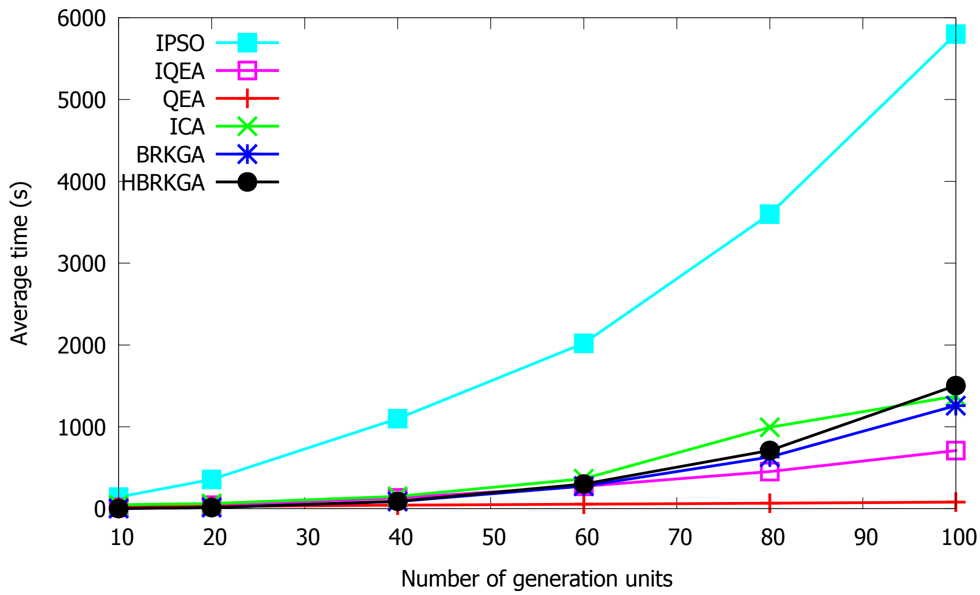

Figure 3.14: Computational time requirements for the methods being compared, for problems in case study 1.

3.7.4 Case study 2

Case study 2 consists of a single real problem instance, the Taipower system, which comprises the scheduling of 38 units for a time horizon of 24 hours. This problem was first proposed by [53]. The start up costs are constant, not necessarily different for all units, while the shut down costs are disregarded. The spinning reserve is set to 11% of hourly load and ramp rate constraints are also taken into consideration. The characteristics of the thermal units, the load demand, and the specific conditions of the problem are given in [53]. This specific problem has not been considered by many authors doing research of the UC problem. Thus, our approach is compared with the four approaches proposed in [53] which are based on dynamic programming (DP), Lagrangian Relaxation (LR), Simulated Annealing (SA), and Constraint Logic Programming (CLP) and also with a GA (MRCGA) recently proposed by [107]. In addition, we also compare our solutions to the solution obtained by CPLEX. This

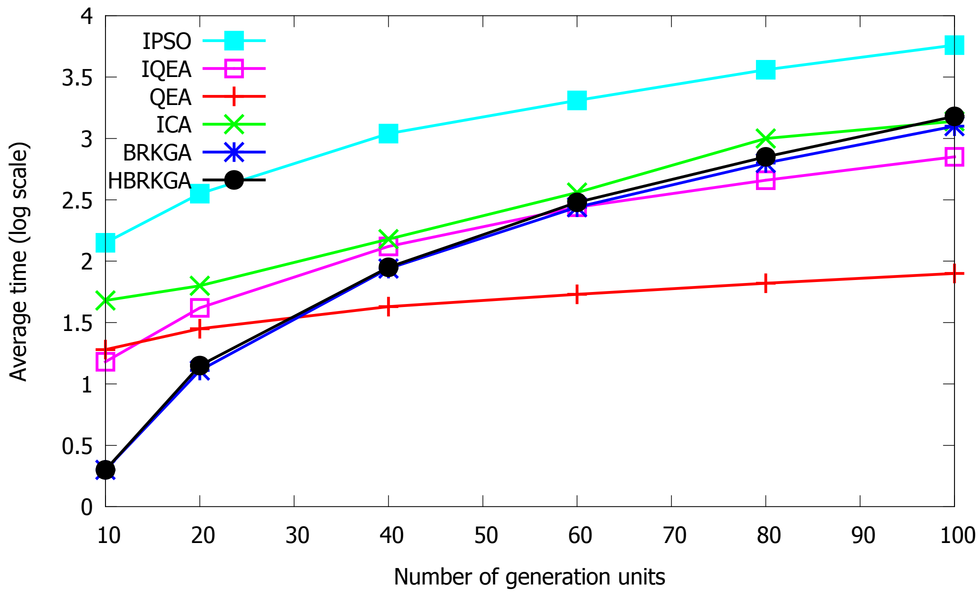

Figure 3.15: Computational time requirements for the methods being compared, for problems in case study 1.

solution may not be optimal since CPLEX has not ran to the end due to the excessive memory requirement. However, we report on the best solution found, before it crashed, and on the time it took to find such a solution for the first time.

Both the BRKGA ad the HBRKGA improve upon the best known solutions, for all cases (best, average, and worst). Again the local search has proved to be effective since in all cases the HBRKGA obtains better solutions than the BRKGA. The computational times are not a concern since the method that takes longer (the DP by [53]) takes just over 3 minutes.

3.7.5 Case study 3

Case study 3 also consists of a single real problem. This problem is a 26-generator system which has to be scheduled for a 24-hour period. Only start-up costs are considered and they are constant, not necessarily the same for all units. The spinning reserve

Table 3.8: Comparison of the results obtained by the BRKGA and the HBRKGA with the best ones reported in literature, for problems in case study 2.

Size	DP	LR	CLP	MRCGA	BRKGA	HBRKGA	CPLEX
Best	215.2	214.5	213.8	*206.7*	206.0	**205.3**	203.6
Average	–	–	–	*207.4*	206.5	**206.1**	–
Worst	–	–	–	*208.0*	207.1	**206.7**	–
Gap(%)	5.7	5.4	5.0	1.5	1.2	0.83	–
St.deviation(%)	–	–	–	–	0.19	0.22	–
Av.Time(s)	199	29	17	45.6	84.7	102.6	1963.9

requirement is set at 400MW for each time period. The system and demand data can be found in [119], as well as, the conditions used in the computational experiments. We compare the solution quality obtained by the BRKGA and by the HBRKGA with that a fuzzy mixed integer Linear Programming proposed in [119]. For this problem, we were able to find an optimal solution by using CPLEX. As it can be seen in Table 3.9, both the BRKGA and the HBRKGA improve on the previously best known results. Again the use of the local search allowed for obtaining an improved solution. Furthermore, the best solution obtained by the HBRKGA is very close to an optimal solution. Thus, CPLEX cannot be considered a better alternative when compared with the HBRKGA since, the latter obtains a solution within 0.26% of optimality, being about 21 times faster.

3.7.6 Case study 4

The problem addressed in this case study comprises 45 units over a planning horizon of 24 hours. The system data and the load demand can be found in [5]. The spinning reserve is set to 10% of the load demand at every hour. Both the start-up and the shut-down costs are constant, not necessarily the same for all units. Table 3.10 shows the best solutions know so far, obtained in [5] from three versions of a GA: global

Table 3.9: Comparison of the results obtained by the BRKGA and the HBRKGA with the best ones reported in literature, for problems in case study 3.

Size	FMILP	BRKGA	HBRKGA	CPLEX
Best	*722388*	722260	**721197**	719314
Average	–	722283	**721202**	–
Worst	–	722410	**721212**	–
Gap (%)	0.43	0.41	0.26	–
St.deviation(%)	–	0.01	0.01	–
Av.Time(s)	25.5	24.3	29.6	642

parallelization (GP), which uses a parallel implementation of the repair algorithm, coarse-grained parallel genetic algorithm (CGPGA), which evolves several populations independently, one on each processor, and hybrid parallel genetic algorithm (HPGA), which combines both previously parallelizations. In addition, our best, average and worst solutions, for both the BRKGA and the HBRKGA, are reported. The methods here proposed improve on the best known solution by 0.22%. For this problem the local search was not effective, since the cost of the best, average, and worst solutions are the same for HBRKGA and the BRKGA. It should be notice that CPLEX was unable to provide any solution for this problem due to its size.

Table 3.10: Comparison of the results obtained by the BRKGA and the HBRKGA with the best ones reported in literature, for problems in case study 4.

Size	GP	CGPGA	HPGA	BRKGA	HBRKGA
Best	1034472374	1032472928	*1032415327*	**1030145017**	**1030145017**
Average	–	–	–	**1030722315**	**1030722315**
Worst	–	–	–	**1034934856**	**1034934856**
Gap (%)	0.42	0.23	0.22	0	–
St.deviation(%)	–			0.14	0.14
Av.Time(s)	80.6	847.1	658.4	115.6	147.3

Regarding the computational time, although our approaches have not been implemented in parallel, they are faster than the approach producing the best former results.

3.7.7 Case study 5

Case study 5 consists of two different problems both considering exponential start-up costs. This type of cost are more realistic and although several authors mention this fact, most end up using constant cost or otherwise approximating them by a piecewise linear function.

Both problems in this case study involve the scheduling of 10 units over a 24-hour time horizon. In both cases the shut-down costs are disregarded.

In the first problem, the spinning reserve is set to 10% of the hourly load demand. All problem data in given in [115], where it has been first proposed. The start-up costs are computed as:

$$S_{t,j} = b_0 . \left(1 - b_1 . e^{-b_2 t}\right) . \tag{3.17}$$

This problem has been addressed in [117], where an optimal solution has been found by using dynamic programming. The authors also propose approximate methods to address this problem: a Lagrangian Relaxation (LR), a genetic algorithm (GA), a memetic algorithm (MA), and a method combining both the LR and MA (LRMA).

In Table 3.11, we report the results published in [117], as well as, the results obtained by our approaches. As it can be seen, we are able to obtain a good solution (with a 0.51% optimality gap), which is better than that of the GA, the MA, and the LRMA proposed in [117]. However, the LR was able to find a better solution. Regarding computational time, our methodologies are much better, being up to 53 times faster. For this problem, it happens again that the local search does not help in finding a better solution.

Table 3.11: Comparison of the results obtained by the BRKGA and the HBRKGA with the best ones reported in literature, for the first problem of case study 5 problem 1.

Size	DP[117]	LR [117]	GA [117]	MA[117]	LRMA [117]	BRKGA	IBRKGA
Best	59478	***59485***	59882	59788	59892	59779	59779
Average	–	***59486***	60364	60271	59936	59836	59834
Worst	–	***59491***	60977	60838	60100	60102	60091
Gap (%)	–	0.01	0.68	0.52	0.7	0.51	0.51
St.deviation(%)	–	0.004	0.74	0.65	0.123	0.11	0.11
Av.Time(s)	207	55	209	161	128	3.9	4.7

The second problem in this case study has been proposed in [8], where the problem data can be found. The spinning reserve requirements are specified for each time period and vary between 6.47% and 11.35%. Regarding the start-up costs, they are exponentially dependent on the number of time periods during which the unit has been off. The data is given in tables A.19 and A.20 in subsection A.5.2.

The start-up costs are given as follows:

$$S_{t,j} = b_0 . \left(1 - e^{-\frac{max(0, -T_j^{off}(t))}{b_2}}\right) + b_1 . \tag{3.18}$$

More recently, other authors have addressed this problem. In Table 3.12, we compare our results with the ones obtained by the LR due to [8], and the recently proposed heuristics: DP - [117]; MA - [117]; FPGA - [22].

As it can be seen, neither the more recently proposed heuristics nor our algorithms were able to improve on the best known results, found by the LR due to [8]. Regarding the quality of the average and of the worst solutions, the HBRKGA is the method that provides the best results. It should be noticed that the BRKGA also presents better average and worst results than the other heuristics. Therefore, the BRKGA and the HBRKGA methods present solutions with the lowest variability. Moreover, the BRKGA and HBRKGA average execution times are much shorter than those of the

Table 3.12: Comparison of the results obtained by the BRKGA and the HBRKGA with the best ones reported in literature, for the second problem in case study 5.

Size	DP	LR	MA	FPGA	BRKGA	HBRKGA
Best	540904	***540895***	541108	541182	542068	541918
Average	–	–	545591	542911	542508	**542372**
Worst	–	–	549290	545572	543377	**543301**
Gap (%)	0.002	–	0.04	0.05	0.21	0.19
St.deviation(%)	–	–	0.61	0.27	0.1	0.11
Av.Time(s)	255	59	101	–	5.9	7.3

other methods, the HBRKGA being up to 43 times faster than the DP heuristic. For this problem the local search is effective, since the HBRKGA solution quality is better for all solution types.

3.8 Conclusions

Biased Random Key GAs have been developed for and applied to several combinatorial optimization problems with interesting results. Given this empirical evidence, see [47], we previously proposed such an algorithm for the unit commitment problem [95]. The results obtained suggested that such an approach would worth while of further investigation. Therefore, in this chapter, we propose a Biased Random Key Genetic Algorithm with Local Search to address the unit commitment problem. In addition, we have improved the decoding and repair procedures used within the GA.

The new algorithm has been tested on a set of UC benchmark problems commonly used and other UC problems found in the literature. The results reported here, show that the proposed method outperforms the current state-of-the-art methods available. For all problem instances, but two, we have been able to find better results then the best results found so far. In addition, these better solutions have been found with computational time requirements, typically, smaller or of the same magnitude than that alternative methods. Furthermore, the results show a further very important feature, lower variability. It should be noticed that the difference between the best and the worst solutions is always below 0.14%, while the difference between the best and the average solutions is always below 0.05%, for the most commonly used problems (case study 1). This is very important since the methods to be used in industrial applications are required to be robust, since otherwise they may lead to poor solutions being used.

Chapter 4

A Genetic Algorithm approach to the Unit Commitment Problem: the multi-objective case

Given the increasing public awareness of environmental impacts, governments have made regulation on pollutants more stringent. Since fossil-fuelled power plants are one of the main contributors to the emission of greenhouse gases to the atmosphere, such concerns are having a significant impact on the operation of power systems. Therefore, the Unit Commitment Problem (UCP), which traditionally minimizes the total production costs, needs to consider the pollutants emissions as another objective in order to address this concern. This way, the UCP becomes a multiobjective problem with two competing objectives. The approach proposed to address this problem combines a Biased Random Key Genetic Algorithm (BRKGA) with a non-dominated sorting procedure. The BRKGA encodes solutions by using random keys, which are represented as vectors of real numbers in the interval $[0,1]$. The non-dominated sorting procedure is then employed to approximate the set of Pareto solutions through an evolutionary optimization process. Computational experiments have been carried out on benchmark systems with 10 up to 100 generation units for a 24 hours scheduling horizon. The results obtained show the effectiveness and efficiency of the proposed BRKGA to find good solutions to the multiobjective UCP. The diversity and well-distribution characteristics of the non-dominated solutions obtained are demonstrated. Furthermore, from the comparison with alternative multiobjective methods it is shown that the method proposed obtains better results in most cases.

4.1 Introduction

During the last few decades the rapid growth in the use of fossil fuels has led to the emission of a large amount of atmospheric pollutants, that are continuously released into the environment. The increased public awareness regarding the harmful effects of atmospheric pollutants on the environment, as well as the tightening of environmental regulations have forced power utilities to search for different operational strategies. These new strategies must lead to a reduction in pollution and environmental emissions. Thus, power utilities look for solutions that in addition to be cost effective must also be pollution concerned. The power system generation scheduling is composed of two tasks [111, 117]: On the one hand, one must determine the scheduling of the turn-on and turn-off of the thermal generating units; on the other hand, one must also determine the amount of power that should be produced by each on-line unit (the latter is also known as economic dispatch) for a specific time horizon. In the Unit Commitment Problem (UCP), such decisions are made simultaneously and in order to minimize the total operating costs. Here, however, and due to the aforementioned environmental concerns, one also wants to minimize the pollutant emissions originated by such production. The Combined economic-environmental UCP, addressed here, considers both optimization problems simultaneously, and optimizes both the cost and the pollutant emissions, resulting in what is known as a multiobjective optimization problem.

Several methods have been reported in the literature over the years to address Environmental/Economic Dispatch (EDD) problem, but not so much to the environmental/economic unit commitment (EEUC) problem. For the latter problem, very few multiobjective approaches, not converting the problem into a single objective one, have been reported. The majority of the studies concerning emission constraints are on the economic dispatch problem, deciding only the power contribution of each thermal unit, but not deciding on which units should be committed for generation at each hour. This problem has been addressed for many years, one of the first papers being that of [42]. For the sake of completion, here are provided several of the most recently proposed heuristics (for the EED), see e.g., [28, 131] for genetic algorithms, [72, 80] for Harmony Search Algorithms, [80, 55] for Differential Evolution Algorithms, [103] for Gravitational search Algorithms, [20, 132, 49] for particle swarm optimization Algorithms, and [84, 83] for bacterial foraging Algorithms.

However, to obtain an optimal solution, it is important to consider not only the output generation level of each generating unit but also and simultaneously the turn on/off schedule, due to start-up costs/emissions that have significant influence in the problem solution. The account of environmental factors in the unit commitment problem did not receive as much attention as in the economic dispatch problem. However, the recent advent of carbon dioxide trading in the European Union has renewed interest in the environmentally constrained unit commitment problem. The environmental concerns have been incorporated into the unit commitment problem in two ways, namely: as a constraint and as an objective. In the latter case, some authors still treat the problem as a single objective problem by combining the two objectives into one, while others address it as a bi-objective problem and thus look for non-dominated solutions.

In some studies, see e.g., [124, 130], the UCP is addressed considering emission constraints. In the aforementioned works, Lagrangian relaxation based algorithms have been proposed. The authors in [124] propose a augmented Lagrangian relaxation, where the system constraints, e.g., load demand, spinning reserve, transmission capacity and environmental constraints, are relaxed by using Lagrangian multipliers, and quadratic penalty terms associated with system load demand balance are added to the Lagrangian objective function. At each iteration, the quadratic penalty terms are linearized, around the solution obtained at the previous iteration, and the resulting problem is decomposed into N subproblems. The corresponding unit scheduling subproblems are solved by dynamic programming, and the economic dispatch is solved by a network flow algorithm.

The authors in [128] provide a series of mixed-integer programming models for the EEUC problem. The models incorporate the costs and emissions in different ways: minimize emissions only, minimize emissions subject to cost limit constraints, minimize costs subject to emission constraints, minimize costs including emissions allowance value with and without emission limits. These models are then linearized by resorting to piecewise linear functions the use of binary variables. The resulting models are solved by using a branch-and-bound MILP solver developed by Zhejiang University. In this type of methods, one objective is optimized, while the remaining objectives are constrained to some limit. One advantage of this type of methods is that it is possible to achieve efficient solutions in a non-convex Pareto-front by varying the limits imposed; this is also a drawback since it involves the choice of appropriate bounds for the constraints. The computational time requirements tend to be too large since many runs must be performed and in addition, the UCP is a NP-hard problem.

The UCP considering emissions as a second objective function but combined with the main objective function (operating costs) has been addressed by several authors and approaches. In [63] the authors combine the objectives functions using a weighting factor and use a Lagrangian-relaxation-based algorithm. The authors in [87] use a price penalty factor, defined as the ratio between maximum fuel cost and maximum emission of corresponding generator, to blend the emission with fuel costs. Since the solution procedure proposed relies on an exhaustive enumeration (generates all possible possible combinations of the generator units status), it guarantees the optimality of the solution. However, it is only feasible for small sized problem instances (it has been tested on a 5 units system). This problem is also addressed in [88], where the authors propose several techniques, namely: genetic algorithms, evolutionary programming, particle swarm optimization, and differential evolution. Although the authors compare the results obtained with the four techniques, it was not possible to draw any strong conclusions about regarding the techniques efficiency and effectiveness since only two problem instances have been solved. In [18] the UCP with three conflicting functions such as fuel cost, emission and reliability level of the system is considered. These functions are formulated as a single objective function using the fuzzy set theory. A binary real coded steps Artificial Bee Colony algorithm is proposed, where the binary coded ABC is used to determine the generation units status and the real coded ABC is used to determine the production of the on-line units. The disadvantage of such approaches is that they do not allow for obtaining a set of solutions with a tradeoff between costs and emissions, since an apriori compromise is defined. In the [129], an approach based on the convex combination of the objective functions, the weighting factor are then varied between 0 and 1. The problem version address only considers constraints on load, spinning reserve, and output limits. The solution procedure is based on the decommitment approach, i.e., it starts by that all units are turned on and then it decommits units one at the time, based on cost savings and on emissions reduction. A single problem instance with 10-units has been solved. This type of approaches has several disadvantages: a uniform spread of weight parameters, in general, does not produce a uniform spread of points on the Pareto-front; Non-convex parts of the Pareto set cannot be reached by minimizing convex combinations of the objective functions; Implies a considerable computational burden since several runs are needed, as many times as the number of desired optimal solutions. Other authors have combined the last two strategies, i.e. combining the two objective functions and imposing constraints on the achievable values for one or both objectives, in order to try to overcome their drawbacks. Catalão et al. [16, 17] address the multi-objective unit commitment problem considering cost and emission objective functions. The authors propose an approach based on Lagrangean relaxation, which combines the weighted sum method, using a convex combination of the objective functions, with the ε-constraining method, constraining the objectives to be within pre-specified threshold levels. The approach was tested on a case study with 11 thermal units and a scheduling time horizon of 168 hours and the results reported demonstrated it to be fast and efficient. This approach has then extended , in [15] to the profit-based unit commitment problem also considering environmental concerns. The main difference between these two problems is that in the former rather than minimizing costs one is interested in minimizing the difference between the costs and the profit. A ratio of change parameter, previously introduced in [14], is computed in order to find the best compromise solution amongst the Pareto-optimal set. This ratio allows infer on relation between the percentage amount of decrease in profit and the corresponding percentage amount of decrease in total emission. The corresponding gradient angle, which is also computed, indicates wether the percentage decrease in the total emissions is small for a significant percentage decrease in total profit (small gradient values) or vice versa.

Current research is directed to handle both objectives simultaneously as competing objectives rather than somehow convert the multiobjective problem into a single objective problem. Despite that multi-objective evolutionary algorithms (MOEAs) can be efficiently used to eliminate most of the difficulties of classical methods, as far as the authors are aware of, only three such methods have been applied to the environmental/economic unit commitment (EEUC) problem. The author in [113] propose a method combining Non-dominated Sorting Genetic Algorithm-II (NSGA-II) with problem specific crossover and mutation operators. The initial population is obtained by randomly generating the units status (binary matrices) except for one solutions that is obtained through a Priority list. The power dispatch is obtained by using the lambda-iteration method. Parents are randomly chosen from a pool, formed using binary tournament, and the offspring is obtained by applying window crossover. Mutation is applied using swap window and window operators. Then the NSGA-II principle is used to form the next generator. The authors have one problem instance with 60 generating units. This work has then been improved in [112], since problem specific binary genetic operators are used for the unit status matrix (commitment matrix) and real genetic operators are used for the power matrix thus exploring the binary and real spaces separately. The authors also use two different crossover procedures, one to evolve the commitment matrix and another to evolve the power matrix. The way feasibility is handled is also different. Solution feasibility is regarding power demand is ensured through a repair mechanism. The violation of other constraints results in a violation penalty, that if below a certain threshold is ignored. The same problem was solved and in the latter work the Pareto-front obtained has many more solutions. In [71] a memetic evolutionary algorithm is proposed. This algorithm is an extension of the well know NSGA-II: non-dominated sorting genetic algorithm-II [27], since it incorporates a local search procedure. The algorithm comprises a two-stages: a multi-objective EA (MOEA) for the generation scheduling problem and the weighted-sum lambda-iteration algorithm proposed in [127] for the power dispatch. The local search operator is applied, at the last iteration, to shut down or turn on some of the units located at the boundaries of the schedules, i.e. when units change status. Computational experiments have been run on systems composed of 10 and 100 generation units for a 24-hour demand horizon. The auhors concluded that the local search procedure is effective since they were able to find some solutions with better trade off, with respect to cost and emission, than those found the pure NSGA-II. However, in none of this works the quality and diversity of the non-dominated solutions found have been measured and assessed quantitatively.

In this chaper, the BRKGA algorithm is combined with a nondominated sorted procedure. The BRKGA approach includes a ranking selection method to evaluate the population and divide it into different Pareto fronts by assigning to each solution a rank equal to its non-domination level (in rank 1 are the non-dominated solutions, in rank 2 are the solutions only dominated by rank 1 solutions, and so on). A crowdcomparison procedure is used to maintain population diversity. The BRKGA developed is based on the framework proposed in [46] and on a previous version developed for the single objective UC problem [95]. Our algorithm is tested on two standard 24-hour test systems, introduced in [126] and [100], each considering several cases involving from 10 up to 100 generating units. Following on the idea presented in [1], we develop a comparative study of our method and other MOEA methods to demonstrate the efficiency and effectiveness of our approach.

4.2 Unit Commitment multi-objective Problem Formulation

In the multiobjective UC problem one needs to determine an optimal schedule and power production, which involves determining the turn-on and turn-off schedule of power units, represented by binary variables u, as well as determining the amount of power produced by each unit, represented by continuous variables y.

The objectives are to minimize the production cost $F(y,u)$ and emission of atmospheric pollutants $E(y,u)$ over the scheduled time horizon subject to system and operational constraints.

$$\text{Minimize} \qquad [F(y,u), E(y,u)] \tag{4.1}$$

4.2.1 Objective Functions

As already said, in the multi-objective problem formulation, two important objectives in electrical thermal power systems are considered.

On the one hand, the first objective is to minimize the system operational costs composed of generation and start-up costs. The generation costs, i.e. the fuel costs, are conventionally given by a quadratic cost function as in equation (4.2),

$$F_j(y_{t,j}) = a_j \cdot (y_{t,j})^2 + b_j \cdot y_{t,j} + c_j, \tag{4.2}$$

where a_j, b_j, c_j are the cost coefficients of unit j, and $y_{t,j}$ is the amount of power to be produced by unit j at time period t.

Therefore, the total operational costs for the whole planning period are given by

$$F(y,u) = \sum_{t=1}^{T} \sum_{j=1}^{N} \left(F_j(y_{t,j}) \cdot u_{t,j} + S_{t,j} \cdot (1 - u_{t-1,j}) \cdot u_{t,j} + Sd_j \cdot (1 - u_{t,j}) \cdot u_{t-1,j} \right), \tag{4.3}$$

where $S_{t,j}$ and Sd_j are the start-up and shut-down costs of unit j at time period t, respectively. The binary variable $u_{t,j}$ is the status of unit j at time period t.

On the other hand, the second objective is to minimize the total quantity of atmospheric pollutant emissions such as NO_x and CO_2. The emissions are generally expressed as a quadratic function:

$$E_j(y_{t,j}) = \alpha_j \cdot (y_{t,j})^2 + \beta_j \cdot y_{t,j} + \gamma_j, \tag{4.4}$$

where $\alpha_j, \beta_j, \gamma_j$ are the emission coefficients of unit j.

So, the total emission of atmospheric pollutants is expressed as follows:

$$\begin{aligned} E(y,u) = \sum_{t=1}^{T} \Bigg(\sum_{j=1}^{N} \{ & E_j(y_{t,j}) \cdot u_{t,j} \\ & + Se_{t,j} \cdot (1 - u_{t-1,j}) \cdot u_{t,j} \} \Bigg), \end{aligned} \tag{4.5}$$

where $Se_{t,j}$ is the start-up atmospheric pollutant emissions of unit j at time period t. In the literature $Se_{t,j} = Se_j$ is generally considered constant.

4.2.2 Constraints

The constraints considered are the same as the ones in the single objective, i.e. equations 3.4 to 3.8 in Section 3.2.

4.3 Multiobjective UC optimization

4.3.1 Decoding procedure

The decoding procedure used in all four multiobjective optimization algorithms is the one described in the previous chapter. For each chromosome, the corresponding solution is performed in two main stages. Firstly, the output generation level matrix for each unit and period is computed using the random key values. In this solution, the units production is proportional to their priority, which is given by the random key value. By doing so, each element of the output generation matrix, $y_{t,j}$ is given as the product of the percentage vectors by the periods demand D_t, i.e., $y_{t,j} = D_t \frac{RK_j}{\sum_{i=1}^{N} RK_i}$. Here each component of the percentage vectors are given by corresponding random key entry divided by the sum of the all random key values as illustrated in algorithm 1 in the previous chapter. Then, these solutions are checked for constraints satisfaction and whenever a constraint is not satisfied the solution is modified by the repair algorithm.

4.3.2 Repair algorithm

The idea of this technique is to convert any infeasible individuals to a feasible solution by repairing the sequential possible violations constraints in the UC problem. The repair algorithm is composed by several steps. Firstly, the output levels are adjusted in order to satisfy the output range constraints. Next, we have the adjustment of output levels to satisfy ramp rate limits. It follows the repairing of the minimum uptime/downtime constraints violation. Afterwards, the output levels are adjusted in order to satisfy spinning reserve requirements. Finally, the output levels are adjusted for demand requirements satisfaction at each time period. Further details of the repair procedure are described in the previous chapter.

4.3.3 Pareto Dominance

The concept of domination is used in the most multi-objective optimization algorithm. The solutions with multiple objectives are compared on the basis of whether one dominates the other solution or not [25]. Let us consider the following multi-objective optimization problem: minimize (maximize) the M components $f_m, m = 1, ..., M$, of a vector $f(x)$ simultaneously, where x is the decision variable in the search space Ω and $f(x) = (f_1(x), ..., f_M(x))$.

Since objectives can be conflicting, instead of searching for a single best solution, the optimization task focuses on finding a set of good compromise solutions [35]. The MOEA techniques used in this chapter are based on concept of Pareto dominance. Assuming a minimization problem, dominance is defined as follows:

Pareto Dominance:

Given the vector of objective functions $f = (f_1, \ldots, f_M)$ is said that candidate x^1 dominates x^2 (for minimizing), written as $x^1 \preceq x^2$, if

$$\begin{aligned} & f_m(x^1) \leq f_m(x^2), \;\; \forall m \in \{1, \ldots, M\} \quad \text{and} \\ & \exists m \in \{1, \ldots, M\} : f_m(x^1) < f_m(x^2). \end{aligned} \tag{4.6}$$

Pareto Optimality:

For a Multi-objective Optimization Problem (MOP), a given solution x^* is Pareto optimal if and only if there is no vector $x \in \Omega$, so that

$$\begin{aligned} &f_m(x) \leq f_m(x^*), \ \forall m \in \{1,\dots,M\} \quad \text{and} \\ &f_m(x) < f_m(x^*) \ \text{ for at least one objective function.} \end{aligned} \tag{4.7}$$

Pareto Optimal Set:

For a MOP, the Pareto Optimal Set ($\mathcal{P}^*$) is defined as

$$\mathcal{P}^* := \{x \in \Omega | \neg \exists x' \in \Omega, f(x\prime) \preceq f(x)\}. \tag{4.8}$$

Pareto Front:

For a MOP and Pareto Optimal Set ($\mathcal{P}^*$), the Pareto Front ($\mathcal{PF}^*$) is defined as

$$\mathcal{PF}^* := \{f(x) = (f_1(x), f_2(x), \cdots, f_M(x)) | x \in \mathcal{P}^*\}. \tag{4.9}$$

In extending the ideas of single-objective EAs to multi-objective cases, three major problems must be addressed:

1. How to accomplish **fitness assignment** and selection in order to guide the search towards the Pareto optimal set;
2. How to maintain a **diverse population** in order to prevent premature convergence and achieve a well distributed, wide spread trade-off front;
3. How to prevent, during the successive generations, that some good solutions are lost.

In general, it is not possible to compute the true Pareto set owing to the complexity of the search space. Therefore, the approaches aim to obtain approximations of the Pareto front and also the correspondent Pareto set.

4.3.4 NSGA II

NSGA II is a fast and elitist non-dominated sorted genetic algorithm [27], which allows to approximate the set of Pareto solution. In this approach, the ranking selection method is used to focus on nondominated solutions while the crowding distance is computed to ensure diversity along the nondominated front. The population of size N_p is used for selection, crossover, and mutation to create a new offspring population of equal size. The rank procedure is employed by different levels of domination until all individuals in the intermediate combined population, of size $2N_p$, are ranked. Firstly, the nondominated solutions are assigned with same rank value and thereafter the crowding distance is computed. The nondominated solutions must be emphasized more than any other solution. In order to find individuals of the next front, the solutions of the first front are temporarily ignored, and the above procedure is repeated to find subsequent fronts. The individuals of the new population are selected from the intermediate population using subsequent nondominated fronts in the order of their ranking. To choose exactly the population members, the solutions of the last front are sorted considering the crowding distance by descending order. The NSGA-II approach proposed by [27] was implemented as follows:

- Generate random initial population of size N_p, decoding the individuals and evaluate the solutions;
- Sort the initial population using non-domination-sort. For each individual assign rank and crowding distance;
- For each generation the following steps are performed: Select the parents, which are fit for reproduction by using the binary tournament selection based on the rank and crowding distance; the genetic operators intermediate crossover and Gaussian mutation are applied under selected parents to create the offspring population of size N_p ; the offspring population is combined with parent population (the size of intermediate population is the double); after non-dominated sorting of

the combined population, only the best N_p individuals are selected based on its rank and crowding distance; a new generation is then obtained maintaining the population size fixed; the stop criterium is a maximum number of generations previously established.

4.3.5 NPGA

A Niched Pareto genetic algorithm was presented in [51]. This technique involves the addition of two specialized genetic operators: Pareto domination tournaments and fitness sharing. These operators allow for selection based on partial ordering of the population, as well as, to preserve diversity in the population.

Tournament selection is used to adjust selection pressure by changing the tournament size. Two candidates are chosen at random from the current population. A comparison set of N_c individuals is also chosen randomly. The sample size N_c gives us control over domination pressure. Each of the candidates are compared to each individual in the comparison set. If a candidate is dominated by the comparison set, and other is not, the former loses the competition. If there are tournament ties, i.e. neither or both candidates are dominated by the comparison set, the selection is based on the fitness sharing of individuals, using niche counts as computed for the objective space in [51]; see equation 4.10. Each candidate niche count is computed in the objective space, using its evaluated objective values. The candidate with lowest niche count wins the tournament. Tournaments are held until the next generation is filled. The niche count for candidate i is given by:

$$m_i = \sum_{j \in Pop} Sh(d_{i,j}), \tag{4.10}$$

where $d_{i,j}$ is the Euclidean distance between competitor i and other individual j and Sh is the fitness sharing function expressed as follows:

$$Sh(d) = \begin{cases} 1 - \frac{d}{\sigma_{share}} & \texttt{if } d < \sigma_{share} \\ 0 & \texttt{if } d >= \sigma_{share} \end{cases}. \tag{4.11}$$

Here σ_{share} is the niche radius, i.e. the specified distance. The winner of the tied tournament is the competitor with the lowest niche count. As in [51], the fitness sharing is updated continuously, once the niche counts are calculated using individuals in the partially filled population of the next generation, rather than that of the current generation. Then crossover and mutation operators are applied to the new population. NPGA approach was implemented following the steps:

- Step 1. Randomly generate an initial population
- Step 2. For each generation, decode the individuals in the population; evaluate the solutions and create the empty offspring population;
- Step 3. Randomly choose two individuals from of population; perform the tournament selection and fitness sharing procedures and find the winner;
- Step 4. Repeat step 3 obtaining another winner (parent);
- Step 5. Perform the crossover and mutation operators with the winner individuals of the step 3 and 4 obtaining the offspring individual and update the offspring population Q;
- Step 6. If $|Q| < N_p$ go to step 3; otherwise, increment the generation counter and if the number of generations is less or equal to G_{max} repeat from Step 2. The algorithm stops when the number of generations is higher than G_{max}.

4.3.6 SPEA 2

The Strength Pareto Evolutionary Algorithm (SPEA) was introduced in [139] and an improved version, known as SPEA2 is given in [138]. In this algorithm, nondominated solutions are stored in an external set. The individuals are assigned according to the Pareto dominance concept. When the nondominated solutions exceed a previously fixed size for the external set, the number of individuals in the external set is reduced by means of a truncation technique, as in [138]. If the number of nondominated individuals is less than the predefined external set size, the external set is filled up by dominated individuals. The fitness assignment occurs in two different stages. The individuals are assigned by the strengths of its dominators in both the external set and the population. Strength represents the number of individuals in the population and in the external set covered by the individual considered. The fitness of each individual is given by the sum of the strengths of its dominators in the external set and in the population. If more than one individual have the same fitness value, the density estimation technique is used as given in SPEA2 [138]. This technique results from an adaptation of the $k-th$ nearest neighbor method. The basic idea of the truncation procedure is to remove the individual which has the minimum distance to another individual. If there are several individuals with minimum distance, the individuals with second smallest distances to another individual are removed and so on. The SPEA-II approach proposed by [138] implements the following steps:

- **Step 1.** Generate the initial population decoding the individuals and evaluate the solutions and create the empty external Pareto-optimal;

- **Step 2.** Compute fitness values of individuals in the population and in the external set;

- **Step 3.** Copy nondominated individuals of the population to the external set;

- **Step 4.** Update the external set keeping only the nondominated solutions. When the number of nondominated solutions is higher than the specified size for the external set, it is reduced by applying the truncation technique. If the number of nondominated individuals is less than the external set size, the external set is filled up with dominated individuals;

- **Step 5.** The algorithm stops when the maximum number of generations is reached;

- **Step 6.** The mating pool is filled using the binary tournament selection with replacement on the updated external set;

- **Step 7.** After the recombination of the mating pool, the genetic operators simulated binary crossover and polinomial mutation are applied and a new population is created. Increment the generation counter and repeat from Step 2.

4.4 BRKGA adapted to multiobjective UC optimization

We also use the ranking selection method for ordering the nondominated solutions according to the Pareto domination concept, while the crowding distance is used to break the ties by choosing the best individuals to be included in new population. The BRKGA has already been described in the previous chapter. The initial population, with size N_p, is created by generating the random keys. Given a population of chromosomes (random keys) the decoding procedure is applied such that to each chromosome corresponds a feasible UC solution. A feasible solution consists of a generation level matrix and the corresponding unit status matrix, both satisfying the UC constraints. The fitness function used to evaluate the solutions includes both the total operational costs and CO_2 or NO_x pollutant emissions. We have adopted a fitness procedure similar to that of NSGA-II, given in [27]. Therefore, the population is sorted based on the nondomination concept. Each solution is assigned a fitness (rank) equal to its nondomination level. The biased selection and biased crossover operators and the introduction of mutants are used to create an offspring population, also of size N_p. On the one hand, the biased selection ensures that one of the parents used for mating comes from a subset containing the best solutions of the current population. On the other hand, the biased crossover chooses with higher probability an allele from the best parent. Mutants are generated in the same way as the initially population and are introduced directly on the next generation.

We start by combining the current population with the newly obtained one. The combined population size is the double ($2N_p$) of the current population and it is sorted by the nondomination criterium (Fast Nondominated Sorting Approach).

The nondomination criterium leads to several levels of nondominated fronts. The first level includes all nondominated individuals of the combined population. Second level, corresponds to a front containing individuals only dominated by the individuals of the first level front. All other levels are defined in a similar way, that is, in each level a front containing individuals dominated by all previous nondominated fronts is obtained. In order to obtain the new population we go through the generated fronts, in ascending order of level, and include all its individuals until we reach N_p. At the last nondominated front level to be included if only some of the individuals are to be chosen, the descending order of crowding distance is used as a selection criterium.

The multiobjective BRKGA flowchart is illustrated in Figure 4.1.

4.4.1 Genetic operators in BRKGA

Biased Selection: a pair of parents are selected from the current population. This population is divided into two sets: The elite set, comprising the best individuals, and the non-elite set, comprising the remaining individuals. One parent is selected from the elite set, while the other parent is chosen from the remaining, non-elite, individuals.

Biased Crossover: Given two parents and a specified probability of crossover, the crossover interchanges the genes or alleles to produce a new individual. As already mentioned, genes are chosen by using a biased uniform crossover, that is, for each gene a biased coin is tossed to decide on which parent the gene is taken from. This way, the offspring inherits the genes from the elite parent with higher probability (0.7 in our case).

Mutants: To ensure diversity and to avoid premature convergence, we introduce a

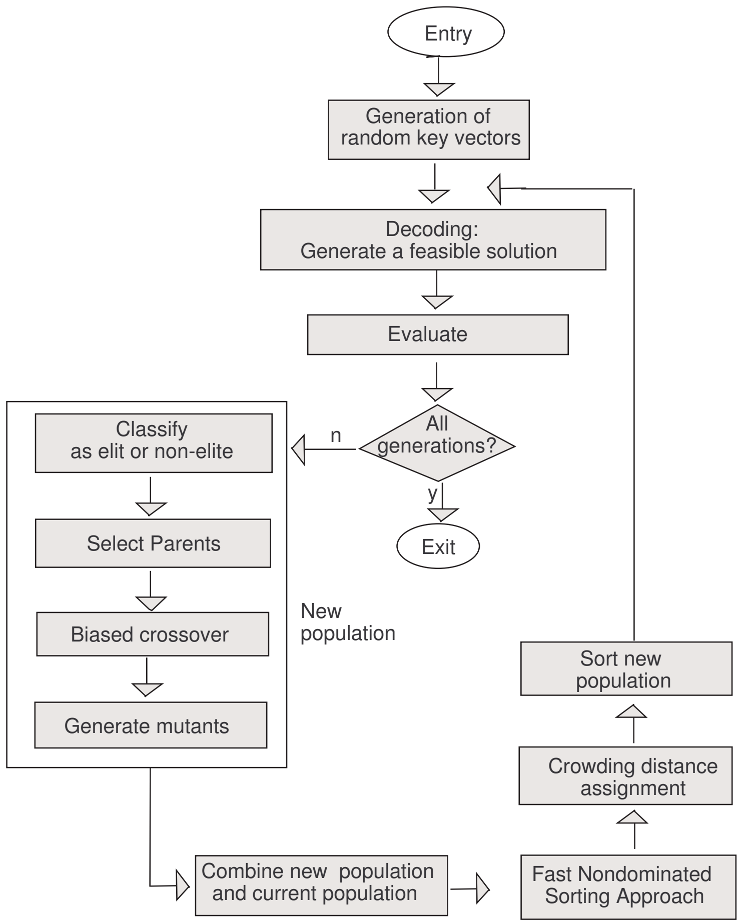

Figure 4.1: Flowchart of BRKGA multiobjective algorithm.

percentage of new individuals, called mutants, in the population. These individuals are randomly generated, as was the case for the initial population.

4.4.2 Performance metrics

In this chapter four different performance measures are used considering the distinct goals of convergence to the Pareto optimal front and the uniformity of distribution in terms of dispersion and extension. We compare the convergence performance of different MOEA using the set coverage metric measure, the contribution measure, the extent indicator measure and the spacing measure. The set coverage measure [140] take under consideration a pair of nondominated sets comparing the fraction of each set that is covered by the other set. This metric is defined as

$$Cov(PF_1, PF_2) = \frac{|\{b \in PF_2; \exists a \in PF_1 : a \ \texttt{cover} \ b\}|}{|PF_2|}, \tag{4.12}$$

where $|.|$ represents the size (cardinality) of a set. When $Cov(PF_1, PF_2) = 0$ means that none of the points in PF_2 are covered by the set PF_1. If $Cov(PF_1, PF_2) = 1$ means that all points in PF_2 are dominated by or equal to points in PF_1. It should be noticed that $Cov(PF_1, PF_2)$ is not necessarily equal to $1 - Cov(PF_2, PF_1)$.

The contribution measure [75] $Con(PF_1, PF_2)$ of an approximation Pareto front PF_1 relatively to another approximation Pareto front PF_2 gives the percentage of the solutions of the nondominated set of $PF_1 \cup PF_2$. Thus, this metric value has to be greater than 0.5 to indicate that PF_1 is better than PF_2 in terms of convergence of the Pareto front. Let PF be the set of solutions in $PF_1 \cap PF_2$, PF^* the set of Pareto solutions of $PF_1 \cup PF_2$. Let D_1 (D_2) be the set of solutions in PF_1 (PF_2) that dominate some solutions of PF_2 (PF_1) and let also N_1 (N_2) be the noncomparable solutions of PF_1 (PF_2). So, the contribution measure is given by:

$$Con(PF_1, PF_2) = \frac{\frac{|PF|}{2} + |D_1| + |N_1|}{|PF^*|}, \tag{4.13}$$

where $|PF^*| = |PF| + |D_1| + |N_1| + |D_2| + |N_2|$. It should be noticed that $Con(PF_2, PF_1) = 1 - Con(PF_1, PF_2)$.

The extent indicator measure is computed as given in [137]

$$E(PF_1) = \sqrt{\sum_{i=1}^{n} max\{\|a_i - b_i\|; a, b \in PF_1\}}, \tag{4.14}$$

where $\|.\|$ is the Euclidean norm. The function E use the maximum extent in each dimension to determine the range to which the front spreads out. In the case of two objectives, this corresponds to distance of the two outer solutions, i.e. gives the distance between the best cost solution and the best emission solution.

Other diversity performance metric is the spacing measure [24]. This measure gives the standard deviation of different distance of solution values in the solution space and is defined as:

$$S = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^{n} \left(d_i - \bar{d}\right)^2}, \tag{4.15}$$

where $d_i = min_j \ d(i, j)$, $\bar{d}$ is the mean distance and $d(i, j)$ is the Euclidean distance between the individual i and j. $S = 0$ means that all members in nondominated set are equidistantly spaced. Moreover, if the nondominated solutions tends to be uniformly distributed the distance will be small. So, smaller spacing measure value means better dispersion of the nondominated solutions.

4.5 Computational Experiments and Results

4.5.1 BRKGA parameters

The BRKGA final parameter values were decided upon after some empirical experiments have been performed. The experimented values were chosen using the guidelines provided by [27, 47], as well as, the computational experiments in the previous chapter.

The current population of solutions is evolved by the GA operators onto a new population as follows: the elite set is formed by 20% of best solutions; 40% of the new population is obtained by introducing mutants; and finally, the remaining 60% of the population is obtained by biased reproduction, which is accomplished by having both a biased selection and a biased crossover. We set the number of generations to $10N$ and the population size to $2N$. Tables 4.1 and 4.2 report the average coverage measure (in percentage) obtained over 10 optimization runs for both instance problems of the 60 units concerning the case studies addressed in subsections 4.5.3 and 4.5.4. Initially the maximum number of generations was considered sufficiently large, $G_{max} = 20.N$. The crossover probability was tried for values selected between $pc = 0.6$ and $pc = 0.9$, in steps of 0.1 and the population size range between N and $5N$, in steps of N. In general, the best coverage performance was obtained for 0.7, as it can be seen in tables 4.1 and 4.2. In addition it should be mentioned that no major differences in terms of the extent and dispersion were found for BRKGA with different crossover probability values.

Table 4.1: Percentage of Nondominated Solutions of set B covered by those in set A, for case study 1.

$N_p = N$				
set A / set B	$BRKGA_{pc=0.6}$	$BRKGA_{pc=0.7}$	$BRKGA_{pc=0.8}$	$BRKGA_{pc=0.9}$
$BRKGA_{pc=0.6}$		27.6	56.6	75.2
$BRKGA_{pc=0.7}$	57.8		66.2	79.7
$BRKGA_{pc=0.8}$	31.8	20.3		62.9
$BRKGA_{pc=0.9}$	17.3	15.4	30.8	
$N_p = 2N$				
$BRKGA_{pc=0.6}$		26.2	34.3	40.7
$BRKGA_{pc=0.7}$	43.7		49.3	62.9
$BRKGA_{pc=0.8}$	41.6	25.8		51.3
$BRKGA_{pc=0.9}$	36.7	16.3	32.8	
$N_p = 3N$				
$BRKGA_{pc=0.6}$		23.0	44.6	43.2
$BRKGA_{pc=0.7}$	37.4		74.3	51.2
$BRKGA_{pc=0.8}$	18.9	11.4		39.4
$BRKGA_{pc=0.9}$	24.1	29.0	38.5	
$N_p = 4N$				
$BRKGA_{pc=0.6}$		30.9	39.8	56.4
$BRKGA_{pc=0.7}$	52.4		59.1	80.8
$BRKGA_{pc=0.8}$	30.0	20.5		55.3
$BRKGA_{pc=0.9}$	24.8	9.5	22.5	
$N_p = 5N$				
$BRKGA_{pc=0.6}$		16.5	46.9	54.7
$BRKGA_{pc=0.7}$	71.4		77.2	88.5
$BRKGA_{pc=0.8}$	24.3	8.2		54.9
$BRKGA_{pc=0.9}$	27.9	5.2	25.4	

The crossover probability was set to $pc = 0.7$. Regarding the population size N_p, its choice must take into account both the coverage performance (in percentage) and

Table 4.2: Percentage of Nondominated Solutions of set B covered by those in set A for case study 2.

$N_p = N$				
set A / set B	$BRKGA_{pc=0.6}$	$BRKGA_{pc=0.7}$	$BRKGA_{pc=0.8}$	$BRKGA_{pc=0.9}$
$BRKGA_{pc=0.6}$		3.1	25.0	4.6
$BRKGA_{pc=0.7}$	91.4		59.6	56.7
$BRKGA_{pc=0.8}$	73.6	32.2		36.5
$BRKGA_{pc=0.9}$	96.9	34.2	60.3	
$N_p = 2N$				
$BRKGA_{pc=0.6}$		24.2	35.5	52.2
$BRKGA_{pc=0.7}$	75.1		70.8	93.5
$BRKGA_{pc=0.8}$	56.1	26.4		76.0
$BRKGA_{pc=0.9}$	38.7	0.7	22.4	
$N_p = 3N$				
$BRKGA_{pc=0.6}$		48.4	48.9	48.2
$BRKGA_{pc=0.7}$	44.1		51.6	38.6
$BRKGA_{pc=0.8}$	56.5	47.5		53.9
$BRKGA_{pc=0.9}$	50.6	50.4	40.1	
$N_p = 4N$				
$BRKGA_{pc=0.6}$		4.9	41.1	20.0
$BRKGA_{pc=0.7}$	89.1		85.8	83.2
$BRKGA_{pc=0.8}$	44.6	17.6		18.3
$BRKGA_{pc=0.9}$	56.4	10.2	71.2	
$N_p = 5N$				
$BRKGA_{pc=0.6}$		30.9	59.6	51.9
$BRKGA_{pc=0.7}$	43.8		65.4	54.8
$BRKGA_{pc=0.8}$	26.7	29.8		22.5
$BRKGA_{pc=0.9}$	37.1	37.6	50.9	

the execution time. Obviously, the coverage performance improves with population size. However, larger population size may render BRKGA impracticable for large thermal system instances. Tables 4.3 and 4.4 show that the *BRKGA* implemented with population size $N_p = 2N$ and $pc = 0.7$ allows to obtain a reasonable execution time.

Ten trials were performed considering different number of generations: $5N$, $10N$, $15N$

Table 4.3: Percentage of Nondominated Solutions of set B covered by those in set A for case study 1 with $pc = 0.7$.

60 units					
set A / set B	$BRKGA_{N_p=N}$	$BRKGA_{N_p=2N}$	$BRKGA_{N_p=3N}$	$BRKGA_{N_p=4N}$	$BRKGA_{N_p=5N}$
$BRKGA_{N_p=N}$		20.1	19.8	8.3	2.1
$BRKGA_{N_p=2N}$	32.4		21.4	18.1	15.2
$BRKGA_{N_p=3N}$	37.8	23.5		21.1	17.4
$BRKGA_{N_p=4N}$	41.3	34.2	31.0		19.9
$BRKGA_{N_p=5N}$	45.5	38.2	35.1	31.7	
Execution time (s)	288.2	664.5	1143.3	1588.4	2140.6

Table 4.4: Percentage of Nondominated Solutions of set B covered by those in set A for case study 1 with $pc = 0.7$.

60 units					
set A / set B	$BRKGA_{N_p=N}$	$BRKGA_{N_p=2N}$	$BRKGA_{N_p=3N}$	$BRKGA_{N_p=4N}$	$BRKGA_{N_p=5N}$
$BRKGA_{N_p=N}$		25.9	23.8	5.6	4.3
$BRKGA_{N_p=2N}$	44.6		35.9	25.0	21.9
$BRKGA_{N_p=3N}$	56.3	37.4		26.1	23.7
$BRKGA_{N_p=4N}$	65	40.2	36.5		28.8
$BRKGA_{N_p=5N}$	71.5	44.5	38.8	34.9	
Execution time (s)	254.2	576.7	935.9	1430.8	1942.1

and $20N$. For each number of generations considered, the population size and crossover probability were fixed to be $2.N$ and 0.7, respectively. Again, a balance between solution quality and computational time must be achieved. A good compromise is obtained with the number of generations being set to $G_{max} = 10N$, as shown in tables 4.5 and 4.6.

Table 4.5: Percentage of Nondominated Solutions of set B covered by those in set A for case study 1 with $pc = 0.7$ and $N_p = 2N$.

60 units				
set A / set B	$BRKGA_{G_{max}=5N}$	$BRKGA_{G_{max}=10N}$	$BRKGA_{G_{max}=15N}$	$BRKGA_{G_{max}=20N}$
$BRKGA_{G_{max}=5N}$		18.2	16.0	13.1
$BRKGA_{G_{max}=10N}$	51.7		41.2	37.6
$BRKGA_{G_{max}=15N}$	53.4	42.4		38.7
$BRKGA_{G_{max}=20N}$	54.0	42.1	40.8	
Execution time (s)	166.8	334.7	477.9	663.8

Table 4.6: Percentage of Nondominated Solutions of set B covered by those in set A for case study 1 with $pc = 0.7$ and $N_p = 2N$.

60 units				
set A / set B	$BRKGA_{G_{max}=5N}$	$BRKGA_{G_{max}=10N}$	$BRKGA_{G_{max}=15N}$	$BRKGA_{G_{max}=20N}$
$BRKGA_{G_{max}=5N}$		23.9	19.7	16.5
$BRKGA_{G_{max}=10N}$	45.7		22.9	18.5
$BRKGA_{G_{max}=15N}$	46.0	25.5		19.9
$BRKGA_{G_{max}=20N}$	50.1	31.6	24.6	
Execution time (s)	139.1	290.3	421.8	556.1

4.5.2 SPEA, NSGA, and NPGA Configurations

The algorithms have been implemented according to their description in the literature. The other operators (recombination, mutation, sampling) remain identical. To ensure the same conditions of application of the BRKGA identical population size, $2N$, and number of generations, $10N$, are used for each algorithm.

The NPGA, NSGA II, and SPEA2 parameters values are chosen using the guidelines proposed in [27]. Some complementary computational experiments are performed, where other appropriate values of the GA parameters are arrived at based on the satisfactory performance of trials conducted for this application with different range of values. For NPGA, the niche radius is $\sigma_{share} = \frac{1}{N}$ as chosen in [51]. Several computational experiments were made in order to choose the size of the comparison set N_c. In the tests this value varied in the interval $[5\%, 30\%]$ with a 5% step. The results obtained have shown a favorable value of N_c to be 10%.

For NPGA and NSGA II real coding an intermediate crossover similar to Matlab crossover operator has been employed. The children are obtained as

$$Child_1 = Parent_1 + rand.ratio.(Parent_2 - Parent_1)$$

and

$$Child_2 = Parent_2 - rand.ratio.(Parent_2 - Parent_1),$$

where *rand* is a random number in the interval $[0,1]$, the ratio crossover was set 1.2 and the crossover probability to 0.8. The Gaussian mutation is used as in Matlab Toolbox Optimization with $scale = 0.1$, $shrink = 0.5$. The mutation rates, has been set to 0.2.

For SPEA2, we use a population of size $2.N$ and an external population of size $2.N$, so that overall population size becomes $4N$. The uniform crossover and simulated binary crossover operators are applied with probability 0.7 and 0.9, respectively. For real-coded crossover, the probability distribution used in the simulated binary crossover operator has been set up distribution index η_c of 5 as in [24]. Like in [26], we use the polynomial mutation described as follows: if x_i is the decision variable selected for mutation with a probability p_m, the result of the mutation is the new value $x_i^{'}$ obtained by a polynomial probability distribution $P(\delta) = \frac{1}{2}.(\eta_m + 1)(1 - |\delta|)$. x_i^L and x_i^U are the lower and upper bound of x_i, respectively, and r_i is a random number in the interval $[0,1]$. Hence, we have

$$x_i^{'} = x_i + \left(x_i^U - x_i^L\right).\delta_i,$$

with

$$\delta_i = \begin{cases} (2r_i)^{\frac{1}{\eta_m+1}} - 1 & \texttt{if } r_i < 0.5, \\ 1 - |2(1-r_i)|^{\frac{1}{\eta_m+1}} & \texttt{if } r_i >= 0.5. \end{cases} \tag{4.16}$$

The distribution index η_m was set to 15 and the mutation probability to 0.1 as recommended by [24]. Table 4.7 has the population size, the crossover and mutation probabilities, and the number of generations used in each approach.

Table 4.7: GA Parameters.

	BRKGA	NSGAII	NPGA	SPEA2
Population size	2N	2N	2N	2N
Crossover probability	0.7	0.8	0.8	0.9
Mutation probability		0.2	0.2	0.1
N. Generations	10N	10N	10N	10N

4.5.3 Case 1 results

The BRKGA and other three multiobjective optimization techniques were tested on a set of benchmark problems, involving system with 10 up to 100 generation units and considering, in each case, a horizon of 24 hours. The 10 generation unit system problem, the base case, was originally proposed by [6, 129] and the system data is provided in Appendix A. Details are given in tables B.2, B.4, B.3 and B.1. Subsequentially, the 20, 40, 60, 80 and 100 generators systems are obtained by duplicating the base case system (i.e. the 10 generators system) and the load demands are adjusted in proportion to the system size. In all cases the spinning reserve is kept at 10% of the hourly demand.

In Figure 4.2 we have plotted the nondominated solutions, i.e. the Pareto front obtained with the four methods. As it can be seen, the BRKGA has the most widely spread front. Therefore, it seems that BRKGA preserves the diversity of the nondominated solutions and have better diversity characteristics and well-distributed over the Pareto-optimal front than other three algorithms.

The average values, over 10 optimization runs of each algorithm, of the four measures is given in tables 4.8 to 4.11. Since the set coverage measure indicates the fraction of each nondominated set that is covered by the other nondominated set, it can be concluded that the nondominated solutions of our method covers relatively higher percentages of the other solutions.

For instance, in the problem with 10 units, on the one hand, as can be seen in Table 4.8, on average the nondominated set achieved by BRKGA dominates about 66.5 % of the nondominated solutions found by NSGA II. However, the front obtained by NSGA II only dominates in less than 11.4 % of the nondominated solutions produced by BRKGA. On the other hand, with regard to NPGA, a BRKGA front dominates on average 91.5% of the corresponding NPGA front, while the nondominated set produced by NPGA only dominates 1.3% the front obtained by BRKGA. Finally, the nondominated set achieved by BRKGA dominates about 55% of the nondominated solutions

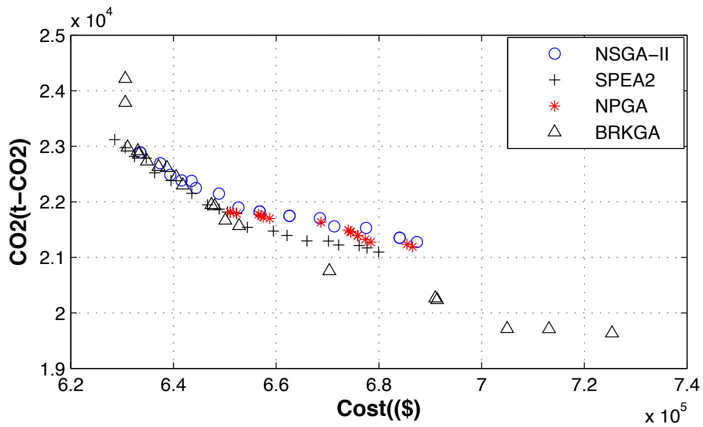

Figure 4.2: Pareto-optimal fronts obtained from different algorithms in a single run for 10 units.

found by SPEA2 while the front obtained by SPEA2 dominates only in less than 26%. Even if we look at the most relative performance of the BRKGA, which occurs for the problem with 80 generation units, it can be seen that the BRKGA dominates in about 59%, 34.6% and 21.1% of the nondominated solutions found by NSGA II, NPGA and SPEA2, respectively. However, the front obtained by BRKGA is dominated only about 1.1%, 0.6% and 16.1% of the NSGA II, NPGA and SPEA2 nondominated solutions, respectively.

Regarding the contribution measure, as said before, it indicates the percentage of the solutions of the nondominated set of $PF_1 \cup PF_2$ that are provided by PF_1. As already said, if $Con(PF_1, PF_2) > 0.5$ means that PF_1 is better than PF_2 in terms of convergence of the Pareto front. Thus, the values reported in Table 4.11 allow in the conclusion that the BRKGA outperforms the other three techniques in terms of convergence.

The spacing measure, which is reported in Table 4.9, reflects how uniformly spread the solutions obtained are. As it can be seen the BRKGA has larger values. Therefore, the nondominated solutions found by it are not as uniformly spread as the ones produced

Table 4.8: Percentage of Nondominated Solutions of set B covered by those in set A.

10 units				
set A / set B	BRKGA	NSGA II	NPGA	SPEA2
BRKGA		66.5	91.5	55
NSGA II	11.4		32	4
NPGA	1.3	18		0.5
SPEA2	26	61.3	91.5	
20 units				
BRKGA		70.3	97.3	69
NSGA II	13.9		44.8	1.8
NPGA	0.9	16.3		0.5
SPEA2	17.8	75.5	91.8	
40 units				
BRKGA		72.1	86.1	43.4
NSGA II	4.7		52.8	0
NPGA	2.4	19.9		0
SPEA2	26.8	90	94.6	
60 units				
BRKGA		68.3	66	60.9
NSGA II	3.4		66	0
NPGA	2.5	21.3		0
SPEA2	10.5	98.8	99.3	
80 units				
BRKGA		59	34.6	21.1
NSGA II	1.1		58.4	0
NPGA	0.6	20.8		0
SPEA2	16.1	97.5	88.9	
100 units				
BRKGA		82.4	55.9	36.7
NSGA II	0.4		46.5	0
NPGA	0.04	37.2		0
SPEA2	13.9	98.2	99.8	

by other methods. Nevertheless, this doesn't seem to be a drawback since the BRKGA is the method that provides the larger extent of nondominated solutions, see Figure 4.2. Finally, the average of extent measure of the nondominated solutions, over 10

Table 4.9: Spacing average measures over 10 optimization runs.

	10	20	40	60	80	100
BRKGA	7279.4	8781.2	11822	12729	12218	12773
NSGA II	6793	7796.1	9414.7	9155.4	10262	8335.7
NPGA	5350.5	5489.9	6790.7	10194	13144	10201
SPEA2	4938.7	7194.7	6151.1	9817.8	7608.2	7282.4

Table 4.10: Extent average measures over 10 optimization runs.

	10	20	40	60	80	100
BRKGA	1140.8	1620	2273.9	2779.1	3202.8	3581.9
NSGA II	1127.1	1586.2	2234.7	2732.3	3143.5	3526.7
NPGA	1103.5	1560.7	2224.2	2672.6	3140.3	3511.9
SPEA2	1124.4	1585.5	2229.5	2731.1	3142.4	3514.2

optimization runs, is given in Table 4.10. When looking at the results for the extent measure, we can infer the distance between the outer nondominated solutions of each technique. It can be seen that the nondominated solutions obtained by the proposed BRKGA span over the entire Pareto-optimal front. Thus, given that the BRKGA has larger values, it can be concluded that it outperforms the other three approaches.

4.5.4 Case 2 results

The second set of benchmark problems also incorporate a system with 10 up to 100 generation units for time horizon of 24 hours. The base case of the 10 generation unit system problem was originally proposed by [126] and the system data is provided in Appendix B. For problem details see tables B.5 to B.7 in Appendix B and the reference therein. Using a similar procedure to the case study 1 systems with 20, 40, 60, 80 and 100 generators are obtained. Here, in all cases the spinning reserve is also kept at 10% of the hourly demand. In Figure 4.3, we have plotted the nondominated solutions for all four methods. As it can be seen, the NPGA is clearly dominated by the other three methods. Regarding the remaining methods, from Figure 4.3 it can be seen that

Table 4.11: Contribution measure percentages.

10 units				
cont(A,B)	BRKGA	NSGA II	NPGA	SPEA2
BRKGA		87.1	98.5	76
NSGA II	12.9		78.8	22
NPGA	1.5	21.2		4.5
SPEA2	24	78	95.5	
20 units				
BRKGA		81.1	99	82.9
NSGA II	18.9		81.9	20.1
NPGA	1	18.1		3.2
SPEA2	17.1	79.9	96.8	
40 units				
BRKGA		87	95.3	66.5
NSGA II	3		88.9	8.6
NPGA	4.7	11.1		2.1
SPEA2	33.5	91.4	97.9	
60 units				
BRKGA		84.1	92.5	76.7
NSGA II	15.9		97.4	1.2
NPGA	7.5	2.6		0.3
SPEA2	23.3	98.8	99.7	
80 units				
BRKGA		79.3	88.1	58.7
NSGA II	20.7		99.1	2.3
NPGA	11.9	0.9		0.5
SPEA2	41.3	97.7	99.5	
100 units				
BRKGA		91.1	94.4	63.3
NSGA II	8.9		91.2	1.7
NPGA	5.6	8.8		0.1
SPEA2	36.7	98.3	99.9	

the nondominated solutions of the NSGA are almost always dominated by the ones obtained by the BRKGA and SPEA2.

In fact these two latter methods are the ones of interest, see tables 4.12 to 4.15. From

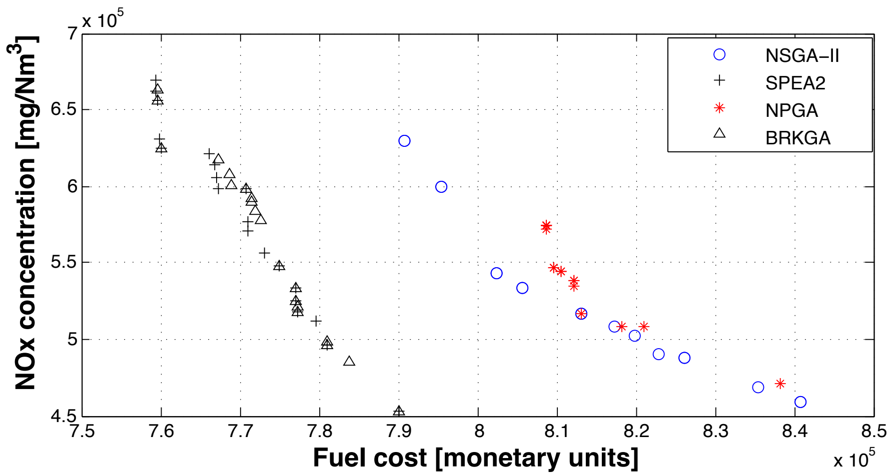

Figure 4.3: Pareto-optimal fronts obtained from different algorithms in a single run for 10 units.

looking these into the results reported for the four measures considered it can be concluded that the nondominated solutions of SPEA2 covers relatively higher percentages of the other solutions. In addition, BRKGA is the second best algorithm in terms of coverage performance. Although the BRKGA front often dominates higher percentages of the corresponding NPGA and NSGA-II fronts, BRKGA nondominated solutions rarely covers SPEA2 solutions. Nevertheless, this is not always the case since, for example, considering the problem with 100 thermal units, we can observe in Table 4.12 that, on average, BRKGA front dominates on average 35.5 % of the corresponding SPEA2 front while the nondominated set produced by SPEA2 dominates 16.3% of the nondominated BRKGA solutions. Moreover, the nondominated set achieved by BRKGA dominates 82.6% of the nondominated solutions found by NSGA II, while the front obtained by NSGA II dominates less than 0.4 % of the nondominated solutions produced by BRKGA. Finally, the BRKGA front dominates on average 57.9% of the corresponding NPGA front while the nondominated set produced by NPGA do not cover any solutions produced by BRKGA.

Table 4.12: Percentage of Nondominated Solutions coverages of set B covered by those in set A.

10 units				
set A / set B	BRKGA	NSGA II	NPGA	SPEA2
BRKGA		75.5	69.5	31.5
NSGA II	12.7		44.5	0
NPGA	23.8	38.5		2
SPEA2	54.4	97	90	
20 units				
BRKGA		46.3	50.5	46.5
NSGA II	34.6		56.8	53.5
NPGA	28.6	33.3		35.5
SPEA2	48.1	29.5	42.3	
40 units				
BRKGA		75.8	62.5	64.8
NSGA II	3.9		38.1	16.3
NPGA	4.8	56.1		27.4
SPEA2	13.6	76.6	56.1	
60 units				
BRKGA		75.2	55.6	24.3
NSGA II	0.6		54.6	0
NPGA	0.15	37.8		5.7
SPEA2	35.7	100	92.6	
80 units				
BRKGA		80.3	77	0
NSGA II	0		64.6	0
NPGA	0	28.1		0
SPEA2	99.4	100	100	
100 units				
BRKGA		82.6	57.9	35.5
NSGA II	0.4		50.2	0
NPGA	0	36.8		0
SPEA2	16.3	98.2	99.7	

Table 4.13: Spacing average measures over 10 optimization runs.

	10	20	40	60	80	100
BRKGA	25748	24719	24289	33963	28341	13350
NSGA II	31245	22638	32695	31628	32111	8329.6
NPGA	14780	29055	35476	43853	42637	9987.9
SPEA2	34167	25959	42436	25413	41035	7282.4

Table 4.14: Extent average measures over 10 optimization runs.

	10	20	40	60	80	100
BRKGA	710.2	1124.3	1659.7	2063.8	2417.7	3580.6
NSGA II	692.9	1061.1	1536.9	1927.6	2232.7	3525.8
NPGA	683.3	1050.3	1505.2	1910.2	2231.8	3512.6
SPEA2	658.9	1089.6	1512.9	1928.7	2261.1	3514.2

The convergence performances of different algorithms are also emphasized in Table 4.15 where we can see that the most of the nondominated solutions obtained by SPEA2 are closer to the true Pareto-optimal solutions since their contribution relatively to another approximation approach is, in general, greater than 50%. This also the case for the BRKGA, except when compared with the SPEA2. However, BRKGA outperforms the other three techniques in terms of the diversity and extent indicators. As it can be seen in Table 4.13, in general, the average spacing measure values, over 10 optimization runs, are smaller than NSGA-II, NPGA and SPEA2 spacing measure values, which means that the BRKGA nondominated solutions are more uniformly distributed than other nondominated solutions obtained by NSGA-II, NPGA and SPEA2. Moreover, Table 4.14 shows that BRKGA has largest extent in all cases. It should be referred that all GAs were implemented on Matlab and executed on a 2 processors Xeon X5450, 3.0 GHz and 4.0 GB RAM. This is a server machine and therefore several jobs are usually running in parallel.

Table 4.15: Contribution measure percentages .

10 units				
cont(A,B)	BRKGA	NSGA II	NPGA	SPEA2
BRKGA		83.2	95.4	38.9
NSGA II	16.8		58.7	2.7
NPGA	4.6	41.3		3.8
SPEA2	61.1	97.3	96.2	
20 units				
BRKGA		49.4	68.2	43.3
NSGA II	50.6		79.1	75.2
NPGA	31.8	20.9		36.9
SPEA2	56.7	24.8	63.1	
40 units				
BRKGA		78.3	79.8	60.6
NSGA II	21.7		59.1	24.6
NPGA	20.2	40.9		18.9
SPEA2	39.4	75.4	81.1	
60 units				
BRKGA		69.3	83.1	26.5
NSGA II	30.7		80.8	0
NPGA	16.9	19.2		2.8
SPEA2	73.5	100	97.2	
80 units				
BRKGA		71.4	75.4	0.3
NSGA II	28.6		71.2	0
NPGA	24.6	28.8		0
SPEA2	99.7	100	100	
100 units				
BRKGA		91.2	95.4	62.7
NSGA II	8.8		92.6	1.7
NPGA	4.6	7.4		0.1
SPEA2	37.3	98.3	99.9	

4.6 Conclusions

This chapter proposes a new approach to find Pareto sets for the multiobjective unit commitment problem. The proposed algorithm combines the biased selection and biased crossover of the BRKGA approach with nondominated sorting procedure and crowded comparison operator used in NSGA II technique.

The algorithm maintains a finite-sized archive of nondominated solutions which gets iteratively updated in the presence of new solutions based on the concept of Pareto dominance.

The proposed approach has been assessed through a comparative study, for two case study problems, with the other state of the art multiobjective optimization techniques. The convergence and diversity performances are evaluated. The best results are obtained for BRKGA and SPEA2 approaches with respect to most of multiobjective performance metrics. Comparatively to the SPEA2, the BRKGA algorithm has best coverage performance but worst diversity performance in first case study, while it has worst performance coverage but best diversity performance in second case study. The results shows that BRKGA can be an effective method for producing tradeoff curves. Tradeoff curves such as those presented here may give decision makers the capability of making better decisions. Moreover, the best diversity performance of the BRKGA in second case study allows the decision maker to have more choices in the selection of solution. Given that the approaches have similar decode procedures, the improvement in performance is most likely due to elitism. Elitism also guarantees that no good solutions are lost.

Chapter 5

Optimal Control Formulations for the Unit Commitment Problem

5.1 Introduction

In this chapter, we address the Unit Commitment (UC) problem using optimal control methodologies. Despite being a highly researched problem with dynamical and multi-period characteristics, it appears that it has not been addressed by optimal control methods before, except in [38] and [34].

A problem that must be solved frequently by a power utility is to economically determine a schedule of which units are to be used and how much each unit should produce in order to meet the forecasted demand while satisfying operational and technological constraints, over a short time horizon [91, 92]. As it was already said good solutions are of most importance since they not only may provide substantial savings (tens to hundreds of millions of euros) in operational and fuel costs but also maintain system reliability by keeping a proper spinning reserve[135]. Due to its combinatorial nature, multi-period characteristics, and nonlinearities, this problem is highly computationally demanding and, thus, solving the UC problem for real-sized systems is a hard optimization task: it is an NP-hard problem [114]. The UC problem has been extensively studied in the literature. Several numerical optimization techniques, based both on exact and on approximate algorithms, have been reported.

Several approaches based on exact methods have been used, such as dynamic programming, mixed-integer programming, benders decomposition, Lagrangian relaxation, and branch-and-bound methods; see, e.g., [67, 21, 109, 7]. The main drawbacks of these traditional techniques are the large computational time and memory requirements for large complexity and dimensionality problems. Dynamic programming [67, 81] is a powerful and flexible methodology; however it suffers from the dimensionality problem, not only in computational time but also in storage requirements. Recently a stochastic dynamic programming approach to schedule power plants was proposed [90]. In [7], a solution using Lagrangian relaxation is proposed. However, the problem becomes too complex as the number of units increases and there are some difficulties in obtaining feasible solutions. Takriti [109] addresses the unit commitment problem by using mixed-integer programming which is a very hard task when the number of units increases since it requires large memory and leads to large computational time requirements. Other authors have proposed the use of mixed-integer linear programming to solve the linearized versions of the problem; see, e.g., [41, 120]. The branch-and-bound method proposed in [21] uses a linear function to represent the fuel consumption and a time-dependent start-up cost, but has an exponential growth in the computational time with problem dimension.

More recently, several metaheuristic methods such as evolutionary algorithms and their hybrids have been proposed; see, e.g., [117, 29, 105, 19, 2]. These approaches have, in general, better performances than the traditional heuristics. The most commonly used metaheuristic methods are simulated annealing [74, 105], evolutionary programming [58, 89], memetic algorithms [117], particle swarm optimization [133], tabu search [73, 121], and genetic algorithms [60, 108, 22, 95]. For further discussion and comparison of these methodologies, with special focus on metaheuristic methods, and other issues related to the unit commitment problem, see the very recent review by Saravanan et al [99].

Although the UC problem is a highly researched problem with dynamical and multi-period characteristics, it appears that it has not been addressed before by optimal control methods, except in [38] and [34] as mentioned previously. In [38], the authors have formulated the UC problem as a discrete mixed-integer optimal control problem, which has then been converted into one with only real-valued controls. Here, we discuss formulations of the UC problem as an Optimal Control (OC) model and propose a new optimal control modeling approach. The model derived is a continuous one and only involves real-valued decision variables (controls).

The main contributions of the proposed modeling approach are twofold. Firstly, since it allows decisions to be taken at any time moment, and not only at specific points in time (usually, hourly), it may render better solutions. It should be noticed that the proposed approach allows for decisions about unit commitment/decommitment and about power production variation at any moment in time. Secondly, it no longer forces utilities to treat demand variations as instantaneous, i.e., time steps. In addition, if one chooses to use the approximated hourly data, as usual in the literature, the solution strategies (both regarding unit commitment/decommitment and power production) of the proposed model will approximate the discrete-time solutions since actions are only required to be taken hourly.

The remaining of this chapter is organized as follows. In Section 5.2, the UC problem is described and its mathematical programming formulation is given. The mixed-integer optimal control formulation and the variable time transformation that allows for rewriting it with only real-valued controls are given in Section 5.3. Section 5.4 provides a detailed description of the continuous-time optimal control model including only real-valued controls, which is proposed here for the first time. Finally, Section 5.5

draws some conclusions and discusses future work.

5.2 The Unit Commitment Problem

The unit commitment problem involves both the scheduling of power units (i.e., the decision when each unit is turned on or turned off along a predefined time horizon) and the economic dispatch problem (the problem of deciding how much each unit that is on should produce). The scheduling of the units is an integer programming problem and the economic dispatch problem is a nonlinear (real-valued) programming problem. The UC problem is then a nonlinear, nonconvex, and mixed-integer optimization problem [22]. The objective of the UC problem is the minimization of the total operating costs over the scheduling horizon while satisfying the system demand, the spinning reserve requirements, and other generation constraints such as capacity limits, ramp rate limits, and minimum uptime/downtimes.

The objective function is expressed as the sum of the fuel, start-up, and shutdown costs.

5.2.1 Mixed-Integer Mathematical Programming Model

The model has two types of decision variables: the binary decision variables $u_j(t)$, which are either set to 1, meaning that unit j is committed at time t, or otherwise are set to zero; the real-valued variables $y_j(t)$, which indicate the amount of power produced by unit j at time t. For the sake of simplicity, we also define the auxiliary variables $T_j^{on/off}(t)$, which represent the number of time periods for which unit j has been continuously online/off-line until time t.

Objective Function:

For benefict of the reading we remind that the objective function has three cost components: generation costs, start-up costs, and shutdown costs. The generation costs, also known as the fuel costs, are conventionally given by the following quadratic cost function:

$$F_j(y_j(t)) = a_j \cdot (y_j(t))^2 + b_j \cdot y_j(t) + c_j, \tag{5.1}$$

where a_j, b_j, c_j are the cost coefficients of unit j.

The start-up costs, that depend on the number of time periods during which the unit has been off, are given by

$$S_j(t) = \begin{cases} S_{H,j}, & \texttt{if } T^{off}_{min,j} \leq T^{off}_j(t) \leq T^{off}_{min,j} + T_{c,j}, \\ S_{C,j}, & \texttt{if } T^{off}_j(t) > T^{off}_{min,j} + T_{c,j}, \end{cases} \tag{5.2}$$

where $S_{H,j}$ and $S_{C,j}$ are, respectively, the hot and cold start-up costs of unit j and $T^{on/off}_{min,j}$ is the minimum uptime/downtime of unit j. The shutdown costs Sd_j for each unit, whenever considered in the literature, are constant.

Therefore, the cost incurred with an optimal scheduling is given by the minimization of the total costs for the whole planning period.
Minimize

$$\sum_{t=1}^{T} \sum_{j=1}^{N} \left(F_j(y_j(t)) \cdot u_j(t) + S_j(t) \cdot (1 - u_j(t-1)) \cdot u_j(t) + Sd_j \cdot (1 - u_j(t)) \cdot u_j(t-1)\right). \tag{5.3}$$

Constraints:
As said before, there are two types of constraints: the operational constraints and the demand constraints. The first set of constraints can be further divided into unit output range limit (equation (5.4)), maximum output variation, i.e., ramp rate constraints (equation (5.5)), and minimum number of time periods that a unit must be continuous in each status (online or off-line) (equations (5.6) and (5.7)), while the second set of constraints can be divided into load requirements (equation (5.8)) and spinning reserve requirements (equation (5.9)).

$$Ymin_j \cdot u_j(t) \leq y_j(t) \leq Ymax_j \cdot u_j(t), \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,...,N\}. \tag{5.4}$$

$$-\Delta_j^{dn} \leq y_j(t) - y_j(t-1) \leq \Delta_j^{up}, \text{ for } t \in \{1,...,T\} \text{ and } j \in \{1,...,N\}. \tag{5.5}$$

$$T_j^{on}(t) \geq T_{min,j}^{on}, \text{ for each time } t \text{ in which unit } j \text{ is turned off and } j \in \{1,...,N\}. \tag{5.6}$$

$$T_j^{off}(t) \geq T_{min,j}^{off}, \text{ for each time } t \text{ in which unit } j \text{ is turned on and } j \in \{1,...,N\}. \tag{5.7}$$

$$\sum_{j=1}^{N} y_j(t) \cdot u_j(t) \geq D(t), t \in \{1,...,T\}. \tag{5.8}$$

$$\sum_{j=1}^{N} Ymax_j \cdot u_j(t) \geq R(t) + D(t), t \in \{1,...,T\}. \tag{5.9}$$

The parameters used in the above equations are defined as follows:

T: Number of time periods (hours) of the scheduling time horizon

N: Number of generation units

$\mathbf{R(t)}$: System spinning reserve requirements at time t, in $[MW]$

$\mathbf{D(t)}$: Load demand at time t, in $[MW]$

$\mathbf{Ymin_j}$: Minimum generation limit of unit j, in $[MW]$

$\mathbf{Ymax_j}$: Maximum generation limit of unit j, in $[MW]$

$\mathbf{T_{c,j}}$: Cold start time of unit j, in $[hours]$

$\mathbf{T_{min,j}^{on/off}}$: Minimum uptime/downtime of unit j, in $[hours]$

$\mathbf{T_{0,j}^{on}}$: Initial state of unit j at time 0, time since the last status switch off/on, in $[hours]$

$\mathbf{T^{off}_{0,j}}$: Initial state of unit j at time 0, time since the last status switch on/off, in $[hours]$

$\Delta_j^{\mathbf{dn/up}}$: Maximum allowed output level decrease/increase in consecutive periods for unit j, in $[MW]$

5.3 Discrete-Time Optimal Control Approach

This section describes the work in [34], where a mixed-integer optimal control model (OCM) is proposed to the UC problem. Although it is possible to address optimal control problems (OCPs) with discrete control sets (see, e.g., [61, 43]), it is computationally demanding. Thus, it was proposed to convert this model into another OCM with only real-valued controls. The conversion process requires the use of a novel variable time transformation that is able to address adequately several discrete-valued control variables arising in the original problem formulation. Finally, the transformed real OCM was transcribed into a nonlinear programming problem to be solved by a nonlinear optimization solver.

5.3.1 Discrete-Time Mixed-Integer Optimal Control Model

The mixed-integer optimal control model has two types of decision/control variables: on the one hand, binary control variables $u_j(t)$, which are either set to 1, meaning that unit j is committed at time t, or otherwise set to zero and on the other hand, real-valued variables $\Delta_j(t)$, which enable to control, by increasing or decreasing, the power produced by unit j at time t. We consider two types of state variables: variables $y_j(t)$, which represent the power generated by unit j at time t and variables $T_j^{on/off}(t)$, which represent the number of time periods for which unit j has been continuously online/offline until time t. For convenience, let us also define the index sets: $\mathcal{T} := \{1,\ldots,T\}$ and $\mathcal{J} := \{1,2,\ldots,N\}$. The parameters related to the problem data are as defined in the previous section. The UC problem can now be formulated as a mixed-integer optimal control model.

Objective Function:

Minimize

$$\sum_{t=1}^{T}\sum_{j=1}^{N}\left(F_j(y_j(t))u_j(t)+S_j(t)(1-u_j(t-1))u_j(t)+Sd_j\cdot(1-u_j(t))\cdot u_j(t-1)\right) \tag{5.10}$$

where the costs are as before.

The state dynamics:

The state dynamics in this model are as follows:

The production of each unit, at time t, depends on the amount produced in the previous time period and is limited by the maximum allowed decrease and increase of the output that can occur during one time period:

$$y_j(t) = \left[y_j(t-1)+\Delta_j(t)\right].u_j(t), \text{ for } t\in\mathcal{T} \text{ and } j\in\mathcal{J}. \tag{5.11}$$

The number of time periods for which unit j has been continuously online until time t is given by

$$T_j^{on}(t) = \left[T_j^{on}(t-1)+1\right].u_j(t), \text{ for } t\in\mathcal{T} \text{ and } j\in\mathcal{J}. \tag{5.12}$$

The number of time periods for which unit j has been continuously off-line until time t is given by

$$T_j^{off}(t) = \left[T_j^{off}(t-1)+1\right].(1-u_j(t)), \text{ for } t\in\mathcal{T} \text{ and } j\in\mathcal{J}. \tag{5.13}$$

Pathwise Constraints:

The constraints are as before, except for the ramp rate constraints, and thus they are given by equation (5.4) and equations (5.6) to (5.9). The ramp rate constraints, which were given by equation (5.5), are now handled by the control constraints:

$$\Delta_j(t) \in \left[-\Delta_j^{dn}, \Delta_j^{up}\right], \text{ for } t \in \mathcal{T} \text{ and } j \in \mathcal{J}. \tag{5.14}$$

5.3.2 The Variable Time Transformation Method

The idea here is to develop a variable time transformation in order to convert the mixed-integer OCM into an OCM with only real-valued controls. The transformation of a mixed-integer optimal control problem into a problem with only real-valued controls is not new nor is the general idea of a variable time transformation method. See the classical reference [54] and also [110, 69, 70, 104, 68]. See also the recent work [44] for a discussion on several variable time transformation methods.

Consider, for each unit j, a non-decreasing real-valued function $t \mapsto \tau_j(t)$. Consider also a set of values $\bar{\tau}_1, \bar{\tau}_2, \ldots$ such that when $\tau_j(t) = \bar{\tau}_k$ for odd k we have a transition from off to on for unit j and when $\tau_j(t) = \bar{\tau}_k$ for even k we have a transition from on to off. So, we consider that unit j is

- on if $\tau_j(t) \in [\bar{\tau}_1, \bar{\tau}_2) \cup [\bar{\tau}_3, \bar{\tau}_4) \cup \ldots \cup [\bar{\tau}_{2k-1}, \bar{\tau}_{2k}) \cup \ldots$
- off if $\tau_j(t) \in [0, \bar{\tau}_1) \cup [\bar{\tau}_2, \bar{\tau}_3) \cup \ldots \cup [\bar{\tau}_{2k}, \bar{\tau}_{2k+1}) \cup \ldots$

It might help to interpret τ_j to be a transformed time scale and the values of $\bar{\tau}_1, \bar{\tau}_2, \ldots$ as switching "times" in the transformed time scale. It can be considered, without loss of generality, that the values $\bar{\tau}_k$ are equidistant. Nevertheless, in real time t, the distance between the two events $\bar{\tau}_k$ and $\bar{\tau}_{k+1}$ can be stretched or shrunk to any nonnegative value, including zero, depending on the shape of the function $t \mapsto \tau_j(t)$.

To simplify the exposition, and without loss of generality, let us consider that $\bar{\tau}_k - \bar{\tau}_{k-1}$ is constant and equal to 1, for all $k = 1, 2, \ldots$. In such case, unit j is

- on if $\tau_j(t) \in [1,2) \cup [3,4) \cup \ldots \cup [2k-1,2k) \cup \ldots$
- off if $\tau_j(t) \in [0,1) \cup [2,3) \cup \ldots \cup [2k,2k+1) \cup \ldots$

Now, consider the controls

$$w(t) \in [0,1], \quad t = 0,1,\ldots,T-1,$$

that represent the increment from $\tau(t)$ to $\tau(t+1)$ such that

$$\tau(t) = \tau_0 + \sum_{k=0}^{t-1} w(k)$$

or

$$w(t) = \tau(t+1) - \tau(t), \quad \text{with } \tau(0) = \tau_0.$$

5.3.3 The Optimal Control Model with real-valued controls

Recall the index set $\mathcal{J}$ and redefine $\mathcal{T}$ to be more consistent with usual discrete-time control formulations.

$\mathcal{T} := \{0,\ldots,T-1\}$ and $\mathcal{J} := \{1,2,\ldots,N\}$.

In the same spirit, we redefine the control $\Delta_j(t)$ for $t \in \{0,\ldots,T-1\}$ to be the amount of power generation incremented or decremented for the next time period (rather than comparatively to the previous period).

Note that the controls are all real-valued and comprise
$\Delta_j(t) \in \left[-\Delta_j^{dn}, \Delta_j^{up}\right]$,
$w_j(t) \in [0,1]$.

Define the sets of time periods:

$$\begin{aligned}
I_j^{on} &:= \{t \in \mathcal{T} : \tau_j(t) \in [2k-1, 2k), k \geq 1\},\\
I_j^{off} &:= \mathcal{T} \setminus I_j^{on},\\
I_j^{off>on} &:= \{t \in \mathcal{T} : \tau_j(t) \geq 2k+1, \tau_j(t-1) < 2k+1, k \geq 0\},\\
I_j^{on>off} &:= \{t \in \mathcal{T} : \tau_j(t) \geq 2k, \tau_j(t-1) < 2k, k \geq 1\}\}.
\end{aligned}$$

Finally, the unit commitment problem can be formulated as an optimal control model, as follows:

Minimize

$$\sum_{j=1}^{N} \left(\sum_{t \in I_j^{on}} F_j(y_j(t)) + \sum_{t \in I_j^{off>on}} S_j(t) + \sum_{t \in I_j^{on>off}} Sd_j \right), \tag{5.15}$$

subject to the dynamic constraints

$$\tau_j(t+1) = \tau_j(t) + w_j(t) \quad j \in \mathcal{J},\ t \in \mathcal{T}, \tag{5.16}$$

$$T_j^{on}(t+1) = \begin{cases} T_j^{on}(t) + 1 & j \in \mathcal{J},\ t \in I_j^{on}, \\ 0 & j \in \mathcal{J},\ t \in I_j^{off}, \end{cases} \tag{5.17}$$

$$T_j^{off}(t+1) = \begin{cases} T_j^{off}(t) + 1 & j \in \mathcal{J},\ t \in I_j^{off}, \\ 0 & j \in \mathcal{J},\ t \in I_j^{on}, \end{cases} \tag{5.18}$$

$$y_j(t+1) = \begin{cases} y_j(t) + \Delta_j(t) & j \in \mathcal{J},\ t \in I_j^{on}, \\ 0 & j \in \mathcal{J},\ t \in I_j^{off}, \end{cases} \tag{5.19}$$

the initial state constraints

$$T_j^{on}(0) = T_{0,j}^{on} \quad \text{(given)}, \tag{5.20}$$

$$T_j^{off}(0) = T_{0,j}^{off} \quad \text{(given)}, \tag{5.21}$$

$$\tau_j(0) = \begin{cases} 0 & \text{if } T_{0,j}^{on} = 0 \\ 1 & \text{if } T_{0,j}^{on} > 0, \end{cases} \tag{5.22}$$

$$y_j(0) = \begin{cases} 0 & \text{if } T_{0,j}^{on} = 0 \\ y_{0,j} \in [Ymin_j, Ymax_j] & \text{if } T_{0,j}^{on} > 0, \end{cases} \tag{5.23}$$

the control constraints

$$\Delta_j(t) \in \left[-\Delta_j^{dn}, \Delta_j^{up}\right], \tag{5.24}$$

$$w_j(t) \in [0,1], \tag{5.25}$$

and the pathwise state constraints

$$y_j(t) \in [Ymin_j, Ymax_j] \quad j \in \mathcal{J},\ t \in I_j^{on}, \tag{5.26}$$

$$\sum_{j \in \mathcal{J}} y_j(t) \geq D_t \quad t = 1, 2, \ldots, T, \tag{5.27}$$

$$\sum_{j \in \mathcal{J}} Ymax_j(t) \geq R_t + D_t \quad t = 1, 2, \ldots, T, \tag{5.28}$$

where $Ymax_j(t) = Ymax_j$ if $t \in I_j^{on}, Ymax_j(t) = 0$ otherwise

$$y_j(t) \in [Y_{min_j}, \max\{Y_{min_j}, \Delta_j^{up}\}] \quad j \in \mathcal{J}, t \in I_j^{off>on}, \tag{5.29}$$

$$T_j^{on}(t-1) \geq T_{min,j}^{on} \quad j \in \mathcal{J},\ t \in I_j^{on>off}, \tag{5.30}$$

$$T_j^{off}(t-1) \geq T_{min,j}^{off} \quad j \in \mathcal{J},\ t \in I_j^{off>on}. \tag{5.31}$$

5.3.4 Conversion into a Nonlinear Programming Problem

To construct the nonlinear programming problem (NLP), let us start by defining the optimization variable x containing both the control and state variables. That is

$$x = [\Delta, w, \tau, T^{on}, T^{off}, y]$$

with dimension $(6T+1)\times N)$.

(We could have considered just the controls Δ, w together with the free initial state $y(0)$. An option which, despite having the advantage of a lower dimensional decision variable, is known to frequently have robustness problems, specially in optimal control problems with pathwise state constraints such as ours. For further discussion, see, e.g., Betts [10].)

The objective function should be rewritten in terms of x: Minimize $J(x)$ over x.

To facilitate the optimization algorithm, we separate the constraints that are simple variable bounds, linear equalities, linear inequalities, and the remaining:

- upper/lower bounds: equations (5.24)-(5.26);
- linear equalities: equation (5.16);
- linear inequalities: equation (5.27);
- nonlinear equalities: equations (5.17)-(5.19); and
- nonlinear inequalities: equations (5.28)-(5.31).

Note that equations (5.20)-(5.23) are not implemented as constraints since the initial values of these state variables are considered as parameters and not variables.

With these considerations the problem is formulated as the following NLP:

$$
\begin{aligned}
&\text{Minimize}_{x\in\mathbb{R}^{(6T+1)\times N}} J(x)\\
&\text{subject to}\\
&\quad LB \leq x \leq UB\\
&\quad A_{eq}x = b_{eq}\\
&\quad A_{ineq}x \leq b_{ineq}\\
&\quad g(x) = 0\\
&\quad h(x) \leq 0.
\end{aligned}
$$

More specifically

Minimize over x

$$J(x) = \sum_{j=1}^{N}\left(\sum_{t\in I_j^{on}} F_j(y_j(t)) + \sum_{t\in I_j^{off>on}} S_j(t) + \sum_{t\in I_j^{on>off}} S_{dj}(t)\right),$$

Subject to

- lower bounds:

  $\Delta_j(t) \geq -\Delta_j^{dn}$, for $t\in\mathcal{T}$ and $j\in\mathcal{J}$,

  $w_j(t) \geq 0, \quad j\in\mathcal{J},\ t\in\mathcal{T};$

  $\tau_j(t) \geq 0, \quad j\in\mathcal{J},\ t\in\mathcal{T},$

  $T_j^{on}(t) \geq 0, \quad j\in\mathcal{J},\ t\in\mathcal{T},$

  $T_j^{off}(t) \geq 0, \quad j\in\mathcal{J},\ t\in\mathcal{T},$

  $y_j(t) \geq 0, \quad j\in\mathcal{J},\ t\in\mathcal{T};$

- upper bounds:

  $\Delta_j(t) \leq \Delta_j^{up}, \quad j\in\mathcal{J},\ t\in\mathcal{T},$

  $w_j(t) \leq 1, \quad j\in\mathcal{J},\ t\in\mathcal{T};$

$\tau_j(t) \leq T, \quad j \in \mathcal{J},\ t \in \mathcal{T},$

$T_j^{on}(t) \leq 2T, \quad j \in \mathcal{J},\ t \in \mathcal{T},$

$T_j^{off}(t) \leq 2T, \quad j \in \mathcal{J},\ t \in \mathcal{T},$

$y_j(t) \leq Ymax_j, \quad j \in \mathcal{J},\ t \in \mathcal{T};$

- linear equalities:

$\tau_j(t+1) - \tau_j(t) - w_j(t) = 0 \quad j \in \mathcal{J},\ t \in \mathcal{T};$

- linear inequalities:

$\sum_{j \in \mathcal{J}} y_j(t) - D(t) \geq 0 \quad t \in \mathcal{T};$

- nonlinear equalities:

$$T_j^{on}(t+1) = \begin{cases} T_j^{on}(t) + 1 & \text{if } j \in \mathcal{J},\ t \in I_j^{on}, \\ 0 & \text{if } j \in \mathcal{J},\ t \in I_j^{off}, \end{cases}$$

$$T_j^{off}(t+1) = \begin{cases} T_j^{off}(t) + 1 & \text{if } j \in \mathcal{J},\ t \in I_j^{off}, \\ 0 & \text{if } j \in \mathcal{J},\ t \in I_j^{on}, \end{cases}$$

$$y_j(t+1) = \begin{cases} y_j(t) + \Delta_j(t) & \text{if } j \in \mathcal{J},\ t \in I_j^{on}, \\ 0 & \text{if } j \in \mathcal{J},\ t \in I_j^{off}, \end{cases}$$

and

- nonlinear inequalities:

$y_j(t) \geq Ymin_j \quad j \in \mathcal{J},\ t \in I_j^{on},$

$\sum_{j \in \mathcal{J}} Ymax_j(t) - R(t) - D(t) \geq 0 \quad t \in \mathcal{T},$

$y_j(t) - Y_{min_j} \geq 0 \quad j \in \mathcal{J}, t \in I_j^{off>on},$

$y_j(t) - \max\{Y_{min_j}, \Delta_j^{up}\} \leq 0 \quad j \in \mathcal{J}, t \in I_j^{off>on},$

$T_j^{on}(t-1) - T_{min,j}^{on} \geq 0 \quad j \in \mathcal{J},\ t \in I_j^{on>off},$

$$T_j^{off}(t-1) - T_{min,j}^{off} \geq 0 \quad j \in \mathcal{J},\ t \in I_j^{off>on}.$$

Of course, since this (real-valued) NLP is a problem that originally was a MI-NLP, it is still a very hard problem. Namely, it is a nonconvex problem and standard NLP solvers will find just a local, not necessarily global, optimum. Nevertheless, this is very useful since it can be embedded, as a local search optimizer, into a global search heuristic method.

5.4 Continuous-Time Optimal Control Approach

This section presents a continuous-time optimal control formulation for the unit commitment problem that uses only real-valued decision variables.

To introduce the ideas and concepts used in this formulation let us start by analyzing a specific and simple situation.

Consider a generation unit for which the minimum time it must be consecutively on is 2 hours ($T_{min}^{on} = 2$) and the minimum time it must be consecutively off is 3 hours ($T_{min}^{off} = 3$). Furthermore, consider also the unit to be initially off-line. Let the unit be turned off and turned on as soon as the elapsed time reaches T_{min}^{on} and T_{min}^{off}, respectively. Such a strategy corresponds to the unit having the maximum number of status switches. Thus, for a 24h period, the profile given in Fig. 5.1 would be obtained.

For the example just described, the times at which status switching occurs are given by

$$t_{i+1} = \begin{cases} t_i + T_{min,j}^{on}, & \text{if } i \text{ is odd,} \\ t_i + T_{min,j}^{off}, & \text{if } i \text{ is even.} \end{cases}$$

All other feasible status switching strategies can be obtained from the one just described by stretching any number of time intervals $[t_i, t_{i+1})$ with $i = 1, \ldots, S$, where S _ the

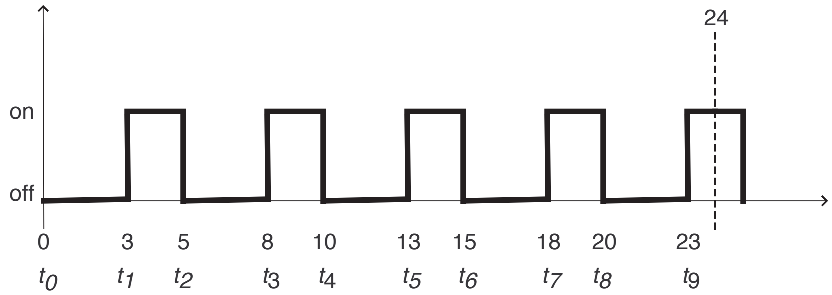

Figure 5.1: Unit status, when the status switching strategy is as often as possible.

maximum number of status switches that can occur within the 24-hours scheduling period _ is given by

$$S = 1 + 2 * \left(24 \text{ DIV } (T^{on}_{min,j} + T^{off}_{min,j})\right),$$

where DIV denotes integer division.

The stretching magnitude α_i in the time interval $[t_i, t_{i+1})$ is bounded from below by 1, since the interval is initially defined as small as possible, and from above by $[1, (24 - t_i)/T_{min}]$, where T_{min} is set to $T^{on}_{min,j}$ or $T^{off}_{min,j}$ depending on whether i is odd or even, respectively, which allows for reaching the end of the scheduling period. It should be noticed that all switches occur at times $t_i \leq 24 - T_{min}$ with T_{min} as defined.

Using a convenient selection of the α_i's any admissible switching profile can be generated. For example, choosing $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_9] = [1, 2, 1, 2, 1, 2, 1, 1, 1, 1]$ leads to the profile given in Fig. 5.2.

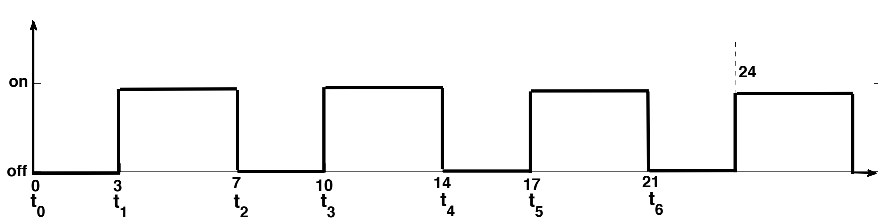

Figure 5.2: Status of unit obtained with $\alpha = [\alpha_0, \alpha_1, \ldots, \alpha_9] = [1, 2, 1, 2, 1, 2, 1, 1, 1, 1]$.

Therefore, in any situation, the computation of the switching times is given by

$$t_{i+1} = \begin{cases} t_i + \alpha_i T_{min,j}^{on} & \text{if } i \text{ is odd,} \\ t_i + \alpha_i T_{min,j}^{off} & \text{if } i \text{ is even.} \end{cases}$$

5.4.1 Formulation

Let us define some parameters before introducing the formulation. When considering several units, the maximum number of switches is not the same for all units since they may have different limits on the number of periods that must elapse before a switch is possible. The same is true for the maximum magnitude of the stretch. Therefore and in order to have one single value for these parameters, we compute upper bounds rather than their true value. By defining

$$T_{min}^{on+off} = \min_j \{T_{min,j}^{on} + T_{min,j}^{off}\},$$

we obtain a limit for the maximum number of switches as

$$S = 1 + 2 * 24 \text{ DIV } \left(T_{min}^{on+off}\right)$$

and the maximum magnitude of the stretch of an interval as

$$s_{max} = 24 / \min_j \{T_{min,j}^{on}, T_{min,j}^{off}\}.$$

For convenience, let us also define the index sets:

$$\mathcal{I} := \{0, 1, \ldots, S\} \text{ \_ switching times indexes,}$$
$$\mathcal{J} := \{1, 2, \ldots, N\} \text{ \_ generation unit indexes,}$$

and the time horizon

$$\mathcal{T} := [0, 24] \text{ \_ time horizon interval.}$$

Decision/Control Variables:

The model has two types of control variables, since two types of decisions are taken. On the one hand, one has to decide for how much time each unit is in each status, that is the magnitude of stretch applied to each time interval for each unit, $\alpha_{i,j}$. On the other hand, one also must decide on the amount of power production for each unit at each time instant. In our case, this is done by deciding on the variation of the production at each time instant $\delta_j(t)$.

$\alpha_{i,j}$: Stretch magnitude applied to the time interval $[t_i, t_{i+1})$ for unit j. These are real-valued variables in the range $[1, s_{max}]$.

$\delta_j(t)$: Rate of change (increase or decrease) for the production of unit j at instant t. These variables are also real-valued and must be within $[-\Delta_j^{dn}, \Delta_j^{up}]$.

State Variables:

The state variables characterize the system and are as follows:

$t_{i,j}$: i-th switching time of unit j;
$u_{i,j}$: Status of unit j in the interval $[t_i, t_{i+1})$, (1 if the unit is on; 0 otherwise);

$u_j(t)$: Status of unit j at instant t, (1 if the unit is on; 0 otherwise);
$y_j(t)$: Power generation of unit j at instant t, in $[MW]$.

Objective Function:

The objective of the UC problem is the minimization of the total costs for the whole planning period, in which the total costs are expressed as the sum of fuel costs and start-up and shutdown costs of the generating units. Therefore, the objective function is as follows:

Minimize

$$\sum_{j\in\mathcal{J}}\int_0^T \left(F_j\left(y_j(t)\right)u_j(t)+S_j(t)\left(1-u_j(t-1)\right)u_j(t)+Sd_j\cdot\left(1-u_j(t)\right)\cdot u_j(t-1)\right)dt.$$

Dynamic Constraints:

We must define the unit status during each time interval. Unit j must have its status switched at the beginning of each interval $[t_i, t_{i+1})$. Thus if in the interval $[t_i, t_{i+1})$ the unit is 1 (on), then in the interval $[t_{i+1}, t_{i+2})$ it becomes 0 (off) and vice versa.

$$u_{i+1,j} = |u_{i,j} - 1|, \quad j \in \mathcal{J}, i \in I.$$

The ending time instant of a time interval, which is the beginning of the next one, is obtained by adding up the starting time instant with the length of the interval.

$$t_{i+1,j} = t_{i,j} + \alpha_{i,j}[T^{on}_{min,j}u_{i,j} + T^{off}_{min,j}(1-u_{i,j})], \quad j \in \mathcal{J}, i \in I.$$

In addition, the power production and, for convenience, the unit status must also be defined for each time instant.

$$u_j(t) = u_{i,j}, \quad j \in \mathcal{J}, i \in I, t \in [t_i, t_{i+1}),$$

$$y_j(t) = \begin{cases} 0 & \text{if } u_j(t) = 0, \\ y_j(t_i) + \int_{t_i}^t \delta_j(s)ds, \text{with } i = \max\{i : t_i \le t\}, & \text{if } u_j(t) = 1, \end{cases} \quad t \in \mathcal{T}, j \in \mathcal{J}.$$

Control Constraints:

Due to the mechanical characteristics and thermal stress limitations, the instantaneous output variation level of each online unit is restricted by ramp rate constraints, both up and down.

$$\delta_j(t) \in [-\Delta_j^{dn}, \Delta_j^{up}], \quad j \in \mathcal{J}, t \in \mathcal{T}.$$

The magnitude of the stretch is limited both from below and from above, since one must assure that the $T_{min,j}^{on/off}$ are satisfied and that the scheduling does not go beyond the scheduling horizon.

$$\alpha_{i,j} \in [1, A_{i,j}], \text{ for } i \in I, j \in \mathcal{J},$$
$$\text{with } A_{i,j} = \begin{cases} \frac{24-t_i}{T_{min,j}^{on} u_{i,j} + T_{min,j}^{off}(1-u_{i,j})} & \text{if } t_i \leq 24 - (T_{min,j}^{on} u_{i,j} + T_{min,j}^{off}(1-u_{i,j})), \\ 1 & \text{otherwise.} \end{cases}$$

Pathwise State Constraints:

Each unit has maximum and minimum output capacity limits.

$$y_j(t) \in [Ymin_j.u_j(t), Ymax_j.u_j(t)] \quad j \in \mathcal{J}, \ t \in \mathcal{T}.$$

The power generated at each time instant must meet the respective load demand.

$$\sum_{j \in \mathcal{J}} y_j(t) \geq D(t) \quad t \in \mathcal{T}.$$

where $D(t)$ is the load demand at time instant t, in $[MW]$.

The spinning reserve is the amount of real power available from online units net of their current production level and it must satisfy a pre-specified value, at each time instant.

$$\sum_{j \in \mathcal{J}} Ymax_j.u_j(t) \geq R(t) + D(t) \quad t \in \mathcal{T}.$$

where $R(t)$ is the pre-specified value of spinning reserve at time instant t, in $[MW]$.

Initial State Constraints:

The initial status of each unit is given.

$$u_{0,j} = \begin{cases} 1, & \texttt{if } T^{on}_{0,j} > 0, \\ 0, & \texttt{if } T^{on}_{0,j} = 0. \end{cases}$$

Also

$u_j(0) = u_{0,j}, \quad j \in \mathcal{J}.$

The first switching interval starts at the beginning of the scheduling horizon and thus

$t_{0,j} = 0, \quad j \in \mathcal{J}.$

Finally, the power production of each online unit has to be within its capacity limits.

$y_j(0) \in [Ymin_j.u_{0,j}, Ymax_j.u_{0,j}], \quad j \in \mathcal{J}.$

The numerical solution of continuous-time optimal control problems has been a well-studied subject for many decades [12] and also has been having recent developments and available solvers such as ICLOCS [31], BOCOP [11], and ACADO [52]. The use of one of these solvers involves always to discretize the problem, transcribe it into a nonlinear programming problem, and use an NLP solver.

The use of a continuous-time formulation for the UC problem has some advantages: (i) the possibility of accommodating any changes in the data or parameters that occur not on an hourly basis, but at any time in between; (ii) in particular, the formulation proposed can deal with continuous-time varying demand (which is more realistic), resulting in an output strategy that responds with continuous-time variations; (iii) however, in the case that the demand and all remaining data vary only on an hourly basis, the resulting output strategy will follow very closely to the one obtained with a discrete-time model; (iv) the complexity of the optimization problem obtained is not increased, possibly being easier to find an optimal solution, since the decision variables involved are all real-valued. It is well known that real-valued nonlinear programming problems are, in general, less difficult to solve than mixed-integer nonlinear programming problems.

5.5 Conclusions

We have addressed the UC problem, a well-researched problem in the literature, which is usually formulated using a mixed-integer nonlinear programming model. Here, we have explored the formulation of this problem using optimal control models. Previous works on an optimal control approach to the UC problem, as far as we are aware of, are limited to the works in [38] and [34] that use a discrete-time optimal control model.

We have proposed here a formulation of the UC problem using a continuous-time optimal control model. An interesting feature of the continuous-time formulation is the fact that, contrary to the usual mixed-integer programming models in the literature, all decision variables are real-valued, which enables the use of more efficient optimization methods for its solution.

Additional advantages of the continuous-time optimal control formulation are the possibility of dealing more accurately with data that is provided with irregular or fast-sampled time intervals, or even continuous-time varying. In particular, this formulation can deal appropriately with continuous-time varying demand data.

In this chapter we discuss how the UC problem can be formulated with an optimal control model, describe previous discrete-time optimal control models, and propose a continuous-time optimal control model.

Chapter 6

Conclusions and future work

6.1 Summary

This thesis addresses the traditional and the environmental/economic unit commitment problems. These problems play a key role in planning and operating of modern electric power systems. In the liberalized markets, an efficient operation, focused on operating cost reduction, is essential. Also with the increased environmental awareness, utilities are forced to change their operational strategies to reduce air pollution and atmospheric emissions.

For these UC problems we proposed two Biased Random Key GAs approaches: the hybrid BRKGA and BRKGA adapted to multi-objective UC problem. Biased Random Key GAs have been developed for and applied to several combinatorial optimization problems with interesting results. The results proved the effectiveness of the BRKGA algorithms for traditional and environmental/economic UC problems under reasonable execution times.

6.2 Single objective UC problem

At the first stage of this work, we address the Unit Commitment problem with a single objective function, namely, the minimization of total operating costs. We propose a Hybrid Random Key Genetic Algorithm with Local Search to address the unit commitment problem. A specific decoding that includes repair procedures within the BRKGA general framework was developed [95]. This way, schedules are constructed using a decode procedure that guarantees the UC solutions feasibility. The algorithm has been tested on a set of UC benchmark problems commonly used and other UC problems found in the literature. For all problem instances, simulation results reveal a satisfactory performance of the HBRKGA, regarding both the quality of solutions and the computational requirements, which are typically smaller or of the same magnitude of alternative methods. Furthermore, the results show a further very important feature, a lower variability. This is very important since the methods to be used in industrial applications are required to be robust, otherwise they may lead to poor solutions being used.

6.3 Multi-objective UC problem

Another important focus of this thesis was the application of the BRKGA to find Pareto sets for the multiobjective environmental/economic unit commitment problem.

This problem involves more information and conflicting objective functions. The simultaneous minimization of operating fuel costs and CO_2, SO_x and NO_x emissions are the objectives. The proposed algorithm combines the biased selection and biased crossover of the BRKGA approach with nondominated sorting procedure and crowded comparison operator used in the NSGA II technique. The algorithm maintains a finite-sized archive of nondominated solutions which gets iteratively updated in the presence of new solutions based on the concept of Pareto dominance.

The proposed approach has been assessed through a comparative study, for two case study problems, with the other state of the art multiobjective optimization techniques. The best results are obtained for BRKGA and SPEA2 approaches regarding most of multiobjective performance metrics. Comparatively to the SPEA2, the BRKGA algorithm has best coverage performance but worst diversity performance in the first case study, while it has worst performance coverage but best diversity performance in the second case study. The results show that BRKGA can be an effective method for producing tradeoff curves. Tradeoff curves such as those presented here may give decision makers the capability of making better decisions. Moreover, the best diversity performance of the BRKGA in second case study allows the decision maker to have more choices in the selection of a solution. Given that the approaches have similar decode procedures, the improvement in performance is most likely due to elitism. Elitism also guarantees that no good solutions are lost. Therefore, the proposed technique can help to reduce the fuel costs and the pollutant emissions simultaneously on daily operation in electric power systems. This issue can lead to less dependence of the fossil fuels and, in consequence, the pollutant emissions reduction, which have a positive impact in environmental issues and the global warming effect.

6.4 Unit Commitment as an optimal control problem

As already referred, the UC problem is a dynamical decision problem. Therefore, it can be formulated as an OCP (with some discrete decision variables). Since the mixed-integer optimization problem thus obtained is very hard to solve, we propose a variable time transformation method that converts the Mixed Integer-OCP into a real-valued OCP (with significant lower dimension than using a ”general” transformation method). The obtained real-valued Non Linear Problem is a reformulation of the original MI-NLP, so it is still a hard problem. It is nonconvex and standard NLP solvers will just find a local, not necessarily global, optimum. However, the OCP approach can be useful as a local search optimizer to be applied after a global heuristic method. In addition, the proposed continuous-time optimal control formulation has the advantage of involving only real-valued decision variables (controls) and enables extra degrees of freedom as well as more accuracy, since it allows to consider sets of demand data that are not sampled hourly.

6.5 Main Contributions

The main contributions of this thesis are the following:

- The decoding procedure of the HBRKGA approach adapted to address the single and multi-objective UC problem.

- The specific repairing mechanisms to find only feasible UC solutions.

- The HBRKGA approach that combines the concept of repairing procedure and elitist strategy has been applied to the UC problem and obtained good results comparatively to other meta-heuristics.

- The BRKGA method adapted to the multi-objective UC problem. This method approach is combined with non-dominated sorted procedure including a ranking selection method and a crowded comparison procedure. The BRKGA itself would be very useful for power planning and/or operating to treat jointly the cost and environmental objective of power system.

- The simultaneous address of the UC and Economic Emission Dispatch (EED) problems. In the past and in recent papers, the economic emission dispatch problem has been addressed. However, the EED does not include the start-up and shut down costs, and it is assumed that all generators are on-line, which is a much simpler problem.

- The effectiveness of the proposed approach is shown through a comparative study, for two test systems with 10-100 generating units, with the other multi-objective optimization techniques, where the performances as well as its variability are evaluated.

- The proposed BRKGA provides many Pareto-optimal solutions in a single run, which gives to decision makers multiples choices.

- The formulation of the UC problem as an optimal control problem.

- The conversion of the single objective UC problem into an Optimal Control approach with only real valued controls.

- The proposed Optimal Control approach allows decisions to be taken at any time moment, and not only at specific points in time (usually, hourly) and it may provide better solutions. In addition, it no longer forces utilities to treat demand variations as instantaneous, i.e., time steps.

6.6 Future work

This thesis has discussed contributions to the UC problem at several levels. Of course, several new research directions are raised. We restrict ourselves to present some pointers:

- In terms of practical implementations, the UC problem including the valve point effect could not be easily addressed from any optimization algorithms. The HBRKGA approach, unlike some other algorithms used in standard UC problem solution, can be applied to the case where the cost function is non-convex, such as it happens when UC problem takes the valve point effect into consideration.

- The development of other local search heuristics to improve the performance in regions near local optima would be an important open task.

- The UC problem is nonconvex and standard NLP solvers will just find a local, not necessarily global, optimum. Thus, the Optimal Control approach could be useful as a local search optimizer to be applied after a global heuristic method (e.g. HBRKGA). However, the combined implementation of the hybridized approach is still unfinished and is expected to bring better results in some cases and, of course, would be an important advantage.

- Another research direction of some importance would be to extend the daily UC problem to more complex planning and operating tasks, such as weekly UC problem and including more input dimensions like, for instance, the number of generation units.

Appendix A

Data for the case studies: single objective optimization

A.1 Data for case study 1

Table A.1: Problem data for the 10-unit base UC problem.

	Unit1	Unit2	Unit3	Unit4	Unit5	Unit6	Unit7	Unit8	Unit9	Unit10
Ymax (MW)	455	455	130	130	162	80	85	55	55	55
Ymin (MW)	150	150	20	20	25	20	25	10	10	10
c ($/h)	1000	970	700	680	450	370	480	660	665	670
b ($/MWh)	16.19	17.26	16.60	16.50	19.70	22.26	27.74	25.92	27.27	27.79
a($\$/MW^2-h$)	0.00048	0.00031	0.002	0.00211	0.00398	0.00712	0.00079	0.00413	0.00222	0.00173
$T^{on}_{min,j}(h)$	8	8	5	5	6	3	3	1	1	1
$T^{off}_{min,j}(h)$	8	8	5	5	6	3	3	1	1	1
hot start cost ($)	4500	5000	550	560	900	170	260	30	30	30
cold start cost ($)	9000	10000	1100	1120	1800	340	520	60	60	60
cold start hrs(h)	5	5	4	4	4	2	2	0	0	0
initial status (h)	8	8	-5	-5	-6	-3	-3	-1	-1	-1

Table A.2: Load Demand

hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MW)	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500
hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MW)	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

A.2 Data for case study 2

Details can be found in [53].

Table A.3: Load Demand

hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MW)	5700	5400	5150	4850	4950	4800	4850	5400	6700	7850	8000	8100
hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MW)	6900	8150	8250	8000	7800	7100	6800	7300	7100	6800	6550	6450

Table A.4: 38-generator system data I.

Unit	Ymax	Ymin	RU	RD	T_i^{on}	T_i^{off}
1	220	550	92	138	18	8
2	220	550	92	138	18	8
3	200	500	84	120	18	8
4	200	500	84	120	18	8
5	200	500	84	120	18	8
6	200	500	84	120	18	8
7	200	500	84	120	18	8
8	200	500	84	120	18	8
9	200	500	84	120	7	7
10	114	500	128	256	7	7
11	114	500	128	256	7	7
12	114	500	128	256	7	7
13	110	500	110	170	9	8
14	90	365	92	125	12	8
15	82	365	92	125	12	8

Table A.5: 38-generator system data I.

Unit	Ymax	Ymin	RU	RD	T_i^{on}	T_i^{off}
16	120	325	82	125	10	8
17	65	315	320	70	1	1
18	65	315	320	70	1	1
19	65	315	320	70	1	1
20	120	272	55	91	9	8
21	120	272	55	91	9	8
22	110	260	53	132	11	8
23	80	190	48	98	14	7
24	10	150	460	20	1	1
25	60	125	42	60	8	8
26	55	110	28	56	14	7
27	35	75	20	38	14	14
28	20	70	70	30	1	1
29	20	70	70	30	1	1
30	20	70	70	30	1	1
31	20	70	75	30	1	1
32	20	60	70	30	1	1
33	25	60	70	30	1	1
34	18	60	70	20	1	1
35	8	60	70	20	1	1
36	25	60	75	30	1	1
37	20	38	10	20	11	8
38	20	38	10	20	11	8

Table A.6: 38-generator system data I.

Unit	A_i	B_i	C_i	HSU_i
1	64782	796.9	0.3133	805000
2	64782	796.9	0.3133	805000
3	64670	795.5	0.3127	805000
4	64670	795.5	0.3127	805000
5	64670	795.5	0.3127	805000
6	64670	795.5	0.3127	805000
7	64670	795.5	0.3127	805000
8	64670	795.5	0.3127	805000
9	172832	915.7	0.7075	402500
10	172832	915.7	0.7075	402500
11	176003	884.2	0.7515	402500
12	173028	884.2	0.7083	402500
13	91340	1250.1	0.4211	575000
14	63440	1298.6	0.5145	575000
15	65486	1298.6	0.5691	575000
16	72282	1290.8	0.5691	575000
17	190928	238.1	2.5881	23000
18	285372	1149.5	3.8734	23000
19	271376	1269.1	3.6842	23000
20	39197	696.1	0.4921	575000
21	45576	690.2	0.5728	575000
22	28770	803.2	0.3572	460000
23	36902	818.2	0.9415	92000
24	105510	33.5	52.123	23000
25	22233	805.4	1.1421	115000
26	30953	707.1	2.0275	287500
27	17044	833.6	3.0744	253000
28	81079	2188.7	16.765	5750
29	124767	1024.4	26.355	5750

Table A.7: 38-generator system data I.

Unit	A_i	B_i	C_i	HSU_i
30	121915	837.1	30.575	5750
31	120780	1305.2	25.098	5750
32	104441	716.6	33.722	7670
33	83224	1633.9	23.915	7670
34	111281	969.5	32.562	7670
35	64142	2625.8	18.362	7670
36	103519	1633.9	23.915	7670
37	13547	694.7	8.482	69000
38	13518	655.9	9.693	69000

A.3 Data for case study 3

26 generating units from [119] with the cost coefficients of generators given in [122].

Table A.8: Load Demand

hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MW)	1700	1730	1690	1700	1750	1850	2000	2430	2540	2600	2670	2590
Spn.res. (MW)	400	400	400	400	400	400	400	400	400	400	400	400
hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MW)	2590	2550	2620	2650	2550	2530	2500	2550	2600	2480	2200	1840
Spn.res. (MW)	400	400	400	400	400	400	400	400	400	400	400	400

Table A.9: 26-generator system (generator data).

Unit	Initial.con.	Start-up cost ($)	Min.up time(h)	Min.down time (h)	Ramp rate ($MW/10-min$)
1-5	-1	0.01	1	0	900
6-9	-1	20	2	0	900
10-13	3	50	3	2	900
14-16	-3	70	4	2	900
17-20	5	150	5	3	900
21-23	-4	200	5	4	900
24	10	300	8	5	900
25	10	500	8	5	900
26	10	500	8	5	900

Table A.10: 26-generator system (generator data).

Unit.	Ymin(MW)	Ymax(MW)	A,$	B,$/$MW$	C,$/$MW^2$
1	2.4	12.0	0.02533	25.5472	24.3891
2	2.4	12.0	0.02649	25.6753	24.4110
3	2.4	12.0	0.02801	25.8027	24.6382
4	2.4	12.0	0.02842	25.9318	24.7605
5	2.4	12.0	0.02855	26.0611	24.8882
6	4.0	20.0	0.01199	37.5510	117.7551
7	4.0	20.0	0.01261	37.6637	118.1083
8	4.0	20.0	0.01359	37.7770	118.4576
9	4.0	20.0	0.01433	37.8896	118.8206
10	15.2	76.0	0.00876	13.3272	81.1364
11	15.2	76.0	0.00895	13.3538	81.2980
12	15.2	76.0	0.00910	13.3805	81.4641
13	15.2	76.0	0.00932	13.4073	81.6259
14	25.0	100.0	0.00623	18.0000	217.8952
15	25.0	100.0	0.00612	18.1000	218.3350

Table A.11: 26-generator system (generator data).

Unit.	Ymin(MW)	Ymax(MW)	A,$	B,$/MW	C,$/MW^2
16	25.0	100.0	0.00598	18.2000	218.7752
17	54.25	155.0	0.00463	10.6940	142.7348
18	54.25	155.0	0.00473	10.7154	143.0288
19	54.25	155.0	0.00481	10.7367	143.3179
20	54.25	155.0	0.00487	10.7583	143.5972
21	68.95	197.0	0.00259	23.0000	259.1310
22	68.95	197.0	0.00260	23.1000	259.6490
23	68.95	197.0	0.00263	23.2000	260.1760
24	140.0	350.0	0.00153	10.8616	177.0575
25	100.0	400.0	0.00194	7.4921	310.0021
26	100.0	400.0	0.00195	7.5031	311.9102

A.4 Data for case study 4

Details can be found in [5].

Table A.12: Load Demand

hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MW)	14000	13500	13400	13500	13700	14000	14400	14500	14700	15000	14700	14500
hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MW)	14300	14000	13800	13700	13700	14000	14300	14300	14700	15000	14500	14000

Table A.13: 45-generator system data I.

Unit	Ymin	Ymax	RU	RD	SU	Sd	UT	DT	In.Status	$Y(0)$
1	888.7	887.2	120	120	888.7	888.7	45	2	1	888.7
2	888.7	887.2	120	120	888.7	888.7	45	2	1	888.7
3	895.0	895.0	120	120	895.0	895.0	45	2	1	895.0
4	895.0	895.0	120	120	895.0	895.0	45	2	1	895.0
5	958.0	958.0	120	120	958.0	958.0	45	2	1	958.0
6	152.0	152.0	40	40	152.0	152.0	45	2	1	152.0
7	989.0	989.0	120	120	989.0	989.0	45	2	1	989.0
8	933.0	933.0	120	120	933.0	933.0	45	2	1	933.0
9	330.0	160.0	165	165	330.0	330.0	45	2	1	300.0
10	138.0	70.0	74	74	138.0	138.0	45	2	1	138.0
11	496.6	76.4	235	235	496.6	496.6	45	2	1	300.0
12	496.6	76.4	300	300	496.6	496.6	45	2	1	300.0
13	139.0	88.0	74	74	139.0	139.0	45	2	1	139.0
14	326.0	163.0	191	191	326.0	326.0	45	2	1	326.0
15	140.0	70.0	60	60	140.0	140.0	45	2	1	100.0
16	331.0	170.0	161	161	331.0	331.0	45	2	1	331.0
17	506.0	230.0	276	276	506.0	506.0	45	2	1	276.0
18	141.0	61.0	64	64	141.0	141.0	45	2	1	100.0
19	326.0	172.0	121	121	326.0	326.0	45	2	1	205.0
20	325.0	214.0	111	111	325.0	325.0	45	2	1	325.0
21	325.0	214.0	111	111	325.0	325.0	45	2	1	325.0
22	325.0	214.0	111	111	325.0	325.0	45	2	1	325.0
23	325.0	214.0	111	111	325.0	325.0	45	2	1	325.0

Table A.14: 45-generator system data I.

Unit	Ymin	Ymax	RU	RD	SU	Sd	UT	DT	In.Status	$Y(0)$
24	205.0	74.0	131	131	205.0	205.0	45	2	1	205.0
25	291.0	158.0	150	150	291.0	291.0	45	2	1	291.0
26	252.0	112.0	60	60	252.0	252.0	45	2	1	192.0
27	326.0	172.0	121	121	326.0	326.0	45	2	1	250.0
28	358.4	133.0	222	222	358.4	358.4	45	2	1	200.0
29	517.2	133.7	300	300	517.3	517.3	45	2	1	250.0
30	147.0	72.0	75	75	147.0	147.0	45	2	1	147.0
31	219.0	150.0	69	69	219.0	219.0	45	2	1	219.0
32	330.0	165.0	165	165	330.0	330.0	45	2	1	330.0
33	330.0	170.0	132	132	330.0	330.0	45	2	1	200.0
34	330.0	170.0	132	132	330.0	330.0	45	2	1	200.0
35	330.0	170.0	132	132	330.0	330.0	45	2	1	200.0
36	525.0	169.0	356	356	525.0	525.0	3	2	3	525.0
37	133.0	37.0	96	96	133.0	133.0	2	2	-9	0.0
38	272.0	94.0	178	178	272.0	272.0	2	2	-2	0.0
39	534.0	175.0	276	276	534.0	534.0	4	2	1	450.0
40	57.0	33.0	24	24	57.0	57.0	4	2	1	57.0
41	206.0	101.0	105	105	206.0	206.0	4	2	1	206.0
42	332.5	87.4	235	235	332.5	332.5	2	2	-4	0.0
43	60.0	45.0	15	15	60.0	60.0	4	2	1	60.0
44	329.0	94.0	235	235	329.0	329.0	2	2	-4	0.0
45	329.0	94.0	80	80	329.0	329.0	2	2	-4	0.0

Table A.15: 45-generator system data II.

Unit	A1	A2	A3	CF	CC	α	C	H
1	0.00	1000	0.000	0	0	2	0	1.00
2	0.00	1000	0.000	0	0	2	0	1.00
3	0.00	1000	0.000	0	0	2	0	1.00
4	0.00	1000	0.000	0	0	2	0	1.00
5	0.00	1000	0.000	0	0	2	0	1.00
6	0.00	1000	0.000	0	0	2	0	1.00
7	0.00	1000	0.000	0	0	2	0	1.00
8	0.00	1000	0.000	0	0	2	0	1.00
9	1.69	1332	164.434	1151000	0	2	20	2.16
10	2.65	1942	68.064	476445	0	2	20	2.18
11	0.51	1872	103.819	726700	0	2	20	5.34
12	0.45	2006	93.919	657400	0	2	20	4.30
13	0.69	2520	22.714	158995	0	2	20	2.12
14	0.96	1858	88.924	622450	0	2	20	2.12
15	8.15	528	135.203	9496400	0	2	20	2.05
16	1.65	1476	179.304	1255100	0	2	20	2.05
17	1.30	1709	301.548	2110800	0	2	20	1.95
18	13.50	-663	178.234	1247600	0	2	20	2.13
19	1.18	1710	115.704	809900	0	2	20	2.13
20	3.09	1405	213.279	1492950	0	2	20	1.38
21	3.09	1405	213.279	1492950	0	2	20	1.38
22	3.09	1405	213.279	1492950	0	2	20	1.38
23	3.09	1405	213.279	1492950	0	2	20	1.38

Table A.16: 45-generator system data II.

Unit	A1	A2	A3	CF	CC	α	C	H
24	2.37	1511	121.064	847400	0	2	20	2.37
25	0.40	2224	47.797	334575	0	2	20	2.27
26	1.89	1536	101.589	711100	0	2	20	2.17
27	1.84	1403	166.427	1164950	0	2	20	2.03
28	0.24	1989	109.896	769250	0	2	20	4.89
29	0.18	2124	99.450	696150	0	2	20	4.89
30	5.08	1118	96.248	673700	0	2	20	2.37
31	0.00	2425	14.807	45969	0	2	20	2.26
32	1.40	1409	136.476	955300	0	2	20	2.26
33	1.98	1553	135.658	949600	0	2	20	1.98
34	1.98	1553	135.658	949600	0	2	20	1.98
35	1.98	1553	135.658	949600	0	2	20	1.98
36	0.10	2029	115.837	81085	0	2	20	3.10
37	0.00	2380	30.000	479960	0	2	20	5.26
38	0.64	2019	69.474	479960	0	2	20	5.32
39	0.48	1864	100.532	703700	0	2	20	3.52
40	-9.58	2401	27.629	57330	0	2	20	2.13
41	1.99	1875	56.816	411710	0	2	20	3.37
42	1.47	1618	120.095	840650	0	2	20	5.36
43	16.33	1374	47.399	331790	0	2	20	2.26
44	0.55	2094	73.451	514150	0	2	20	5.34
45	0.50	2204	61.565	430955	0	2	20	4.30

A.5 Data for case study 5

A.5.1 The First problem data set

Details can be found in [115].

Table A.17: The first problem data of case study 5.

	Unit1	Unit2	Unit3	Unit4	Unit5	Unit6	Unit7	Unit8	Unit9	Unit10
Ymax (MW)	520	320	280	200	150	150	120	100	80	60
Ymin (MW)	250	120	75	75	50	50	25	30	20	15
c ($/h)	105	49	72	82	100	29	32	40	25	15
b ($/MWh)	1.3954	1.2643	1.35	1.2136	1.3285	1.54	1.4	1.35	1.5	1.4
a($\$/MW^2-h$)	0.00127	0.00289	0.00261	0.00148	0.00135	0.00212	0.00382	0.00393	0.00396	0.0051
$T^{on}_{min,j}(h)$	5	5	5	5	5	5	5	5	5	5
$T^{off}_{min,j}(h)$	2	2	2	2	2	2	2	2	2	2
b_0	267	187	176	227	282	113	94	114	101	85
b_1	0.749	0.617	0.568	0.641	0.749	0.639	0.65	0.57	0.594	0.588
b_2	0.09	0.130	0.15	0.11	0.09	0.18	0.18	0.2	0.2	0.2
initial status (h)	6	6	6	6	6	6	6	6	6	6

Table A.18: Load Demand and spinning reserve for first problem of case study 5

hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MWH)	1459	1372	1299	1285	1271	1314	1372	1314	1271	1242	1197	1182
Reserve (MWH)	146	137	130	129	127	131	137	131	127	124	120	118
hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MWH)	1154	1138	1124	1095	1066	1037	993	978	963	1022	1081	1459
Reserve (MWH)	115	114	112	110	107	104	99	98	96	102	108	146

A.5.2 The second problem data set

Details can be found in [8].

Table A.19: The second problem data of case study 5.

	Unit1	Unit2	Unit3	Unit4	Unit5	Unit6	Unit7	Unit8	Unit9	Unit10
Ymax (MW)	1000	850	750	700	600	420	400	375	250	200
Ymin (MW)	300	275	250	225	165	130	130	110	75	50
c ($/h)	820	725	600	540	600	420	400	400	200	175
b ($/MWh)	9.023	8.162	9.121	9.223	8.752	8.431	7.654	7.762	8.149	7.054
a($\$/MW^2-h$)	0.00113	0.00128	0.00131	0.00234	0.00147	0.00150	0.00160	0.00171	0.00452	0.00515
$T^{on}_{min,j}(h)$	5	4	3	4	2	1	3	1	2	2
$T^{off}_{min,j}(h)$	4	3	4	5	4	3	2	3	1	2
b_0	2050	2200	2300	2100	2100	1480	1460	1370	1180	1360
b_1	825	950	950	900	950	650	650	550	625	750
b_2	4	4	4	3	4	4	3	3	2	2
initial status (h)	-4	2	6	-8	1	-2	5	-1	-7	-1

Table A.20: Load Demand and spinning reserve for second problem of case study 5

hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand (MWH)	1025	1000	900	850	1025	1400	1970	2400	2850	3150	3300	3400
Reserve (MWH)	85	85	65	55	85	110	165	190	210	230	250	275
hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand (MWH)	3275	2950	2700	2550	2725	3200	3300	2900	2125	1650	1300	1150
Reserve (MWH)	240	210	200	195	200	220	250	210	170	130	100	90

Appendix B

Data for the case studies: multiobjective optimization

B.1 Data for case study 1

For more details see [6, 129].

Table B.1: Load demand (MW) in case study 1.

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Load demand (MW)	700	750	850	950	1000	1100	1372	1314	1271	1400	1450	1500
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Load demand (MW)	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

Table B.2: Generation constraints in case study 1.

Unit	$Ymax_j(MW)$	$Ymin_j(MW)$	$\mathbf{T^{on}_{min,j}}(h)$	$\mathbf{T^{off}_{min,j}}(h)$	Ramp rate (MW/h)
1	455	150	8	8	250
2	455	150	8	8	250
3	130	20	5	5	80
4	130	20	5	5	80
5	162	25	6	6	100
6	80	20	3	3	80
7	85	25	3	3	85
8	55	10	1	1	55
9	55	10	1	1	55
10	55	10	1	1	55

Table B.3: Data fuel costs evaluation in case study 1.

Unit	a_j $(t-CO_2/MW^2h)$	b_j $(t-CO_2//MWh)$	c_j $(t-CO_2/h)$	startupCO_2 $(t-CO_2)$
1	2.240E-05	0.7557	46.677	210.0
2	1.446E-05	0.8056	45.276	233.3
3	9.335E-05	0.7748	32.674	25.67
4	9.848E-05	0.7701	31.740	26.13
5	3.197E-05	0.1582	3.6157	7.231
6	5.720E-05	0.1788	2.9729	1.365
7	7.282E-05	0.2557	4.4248	2.396
8	3.807E-05	0.2389	6.0841	0.2765
9	2.046E-05	0.2513	6.1302	0.2765
10	1.594E-05	0.2561	6.1763	0.2765

Table B.4: Data fuel costs evaluation in case study 1.

Unit	$A_j(\$/MW^2h)$	$B_j(\$/MWh)$	$C_j(\$/h)$	startup cost ($)
1	0.000528	17.809	1100	4950
2	0.000341	18.986	1067	5500
3	0.0022	18.26	770	605
4	0.002321	18.15	748	616
5	0.004378	21.67	495	990
6	0.007832	24.486	407	187
7	0.000869	30.514	528	286
8	0.004543	28.512	726	33
9	0.002442	29.997	731.5	33
10	0.001903	30.569	737	33

B.2 Data for case study 2

For more details see [126, 136].

Table B.5: Data fuel costs evaluation in case study 2.

Unit	$Ymax_j$ (MW)	$\mathbf{T^{on}_{min,j}}$ (h)	$\mathbf{T^{off}_{min,j}}$ (h)	I_s (h)	A_j $(m.u./MW^2)$	B_j $(m.u./MW)$	C_j $(m.u.)$
1	520	8	4	-5	0.0085	19.566	4437.2
2	320	5	2	-6	0.0050	20.927	1044.20
3	280	5	2	3	0.0253	18.995	1236.9
4	200	5	2	-3	0.0091	23.107	416.58
5	150	5	3	-7	0.0106	20.765	485.69
6	150	4	2	3	0.0116	22.251	300.86
7	120	4	2	5	0.0212	15.031	315.44
8	100	4	2	1	0.0254	15.031	262.87
9	80	3	1	-1	0.0356	10.375	222.16
10	60	3	1	-1	0.0454	9.9214	159.33

Table B.6: Start-up costs, shut down costs and NO_x emissions coefficients in case study 2.

Unit	a_j $(m.u.)$	b_j $(m.u.)$	c_j $(m.u.)$	$\mathbf{Sd_j}$ $(m.u.)$	D_j	E_j	F_j
1	267	34.75	0.09	75	-0.245	154.16	-1154.6
2	187	38.62	0.13	70	-0.002	16.414	-691.1
3	176	27.57	0.15	42	-0.069	36.931	-1626
4	227	26.64	0.11	62	0.1313	-20.77	1885.6
5	113	18.64	0.18	29	-0.005	16.287	-321.4
6	282	45.48	0.09	49	0.1686	-20.0	1361.8
7	94	10.65	0.18	32	0.016	1.7774	276.59
8	114	22.57	0.20	40	0.0193	1.7774	230.49
9	101	20.59	0.20	25	-1.793	246.71	-2636
10	85	20.59	0.20	15	-2.286	235.92	-1890

Table B.7: Load demand (MW) in case study 2.

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Load demand (MW)	1459	1372	1299	1280	1271	1314	1372	1314	1271	1242	1197	1182
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Load demand (MW)	1154	1138	1124	1095	1066	1037	993	978	963	1022	1081	1459

Bibliography

[1] M.A. Abido. Multiobjective Evolutionary Algorithms for Electric Power Dispatch Problem. *IEEE Transactions on Evolutionary Computation*, ‘10:315–329, 2006.

[2] K. Abookazemi, M.W. Mustafa, and H. Ahmad. Structured Genetic Algorithm Technique for Unit Commitment Problem. *International Journal of Recent Trends in Engineering*, 1(3):135–139, 2009.

[3] R. Akbari and K. Ziarati. A multilevel evolutionary algorithm for optimizing numerical functions. *International Journal of Industrial Engineering Computations*, 2(1):419–430, 2011.

[4] G. Anders. Commitment techniques for combined-cycle generating units. *Report no.:T053700-3103, Kinectrics Inc.*, 2005.

[5] J.M. Arroyo and A.J. Conejo. A parallel repair genetic algorithm to solve the unit commitment problem. *IEEE Transactions on Power Systems*, 17:1216–1224, 2002.

[6] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa. A Hybrid LR-EP for Solving New Profit-Based UC Problem Under Competitive Environment. *IEEE Transactions on Power Systems*, 18:229–237, 2003.

[7] A.G. Bakirtzis and C.E. Zoumas. Lambda of Lagrangian Relaxation Solution to Unit Commitment Problem. *Proc. Inst. Elect. Eng., Gener. Trans. Distr.*, 147(2):131–136, 2000.

[8] J.F. Bard. Short-term scheduling of thermal electric generators using Lagragian Relaxation. *Operation Research*, 36(5):756–766, 1988.

[9] J.C. Bean. Genetic Algorithms and Random Keys for Sequencing and Optimization. *Operations Research Society of America Journal on Computing*, 6(2):154–160, 1994.

[10] J. Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2nd edition, 2010.

[11] J.F. Bonnans, V. Grelard, and P. Martinon. BOCOP, the optimal control solver: Open source toolbox for optimal control problems. http://bocop.org, 2011.

[12] A.E. Bryson and Y. C. Ho. *Applied Optimal Control*. John Wiley and Sons, New York, 1975.

[13] M. Carrion and J.M. Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 21(3):1371–1378, 2006.

[14] J.P.S. Catalão, J.P.S. Mariano, V.M.F. Mendes, and L.A.F.M. Ferreira. Short-term scheduling of thermal units: emission constraints and trade-off curves. *European Transactions on Electrical Power*, 18:1–14, 2008.

[15] J.P.S. Catalão, J.P.S. Mariano, V.M.F. Mendes, and L.A.F.M. Ferreira. A practical approach for profit-based unit commitment with emission limitations. *Electrical Power and Energy Systems*, 32:218–224, 2010.

[16] João Catalão, Sílvio Mariano, Victor Mendes, and Luís Ferreira. Unit commitment with environmental considerations: a practical approach. In *Proceedings of the 15th Power Systems Computation Conference, Liege, Belgium*, 2005.

[17] J.P.S. Catalao, S.J.P.S. Mariano, V.M.F. Mendes, and L.A.F.M. Ferreira. Unit commitment in a competitive and emission constrained environment. *IEEE Latin America Transactions*, 7(5):560–568, 2009.

[18] K. Chandrasekaran and S.P. Simon. Multi-objective unit commitment problem with reliability function using fuzzified binary real coded artificial bee colony algorithm. *IET generation, transmission & distribution*, 6(10):1060–1073, 2012.

[19] Y.M. Chen and W.S. Wang. Fast solution technique for unit commitment by particle swarm optimisation and genetic algorithm. *International Journal of Energy Technology and Policy*, 5(4):440–456, 2007.

[20] Y.M. Chen and W.S. Wang. Environmentally constrained economic dispatch using Pareto archive particle swarm optimisation. *International Journal of Systems Science*, 41:593–605, 2010.

[21] A. I. Cohen and M. Yoshimura. A Branch-and-Bound Algorithm for Unit Commitment. *IEEE Transactions on Power Apparatus and Systems*, 102(2):444–451, 1983.

[22] C. Dang and M. Li. A floating-point genetic algorithm for solving the unit commitment problem. *European Journal Operational Research*, 181(4):1370–1395, 2007.

[23] K. Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering (in Press)*, 2001.

[24] K. Deb. *Optimization using evolutionary algorithms*. 2nd ed. New York: Wiley, 2001.

[25] K. Deb. *Multi-Objective Optimization using evolutionary algorithms*. 1st ed. New York: Wiley, 2009.

[26] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115–148, 1995.

[27] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[28] S. Dhanalakshmi, S. Kannan, K. Mahadevan, and S. Baskar. Application of modified NSGA-II algorithm to Combined Economic and Emission Dispatch problem. *Electrical Power and Energy Systems*, 33:992–1002, 2011.

[29] G. Dudek. Unit commitment by genetic algorithm with specialized search operators. *Electric Power Systems Research*, 72(3):299–308, 2004.

[30] M. Ericsson, M.G.C. Resende, and P.M. Pardalos. A Genetic Algorithm for the Weight Setting Problem in OSPF Routing. *Journal of Combinatorial Optimization*, 6(3):299–333, 2002.

[31] Paola Falugi, Eric Kerrigan, and Eugene van Wyk. Imperial College London Optimal Control Software User Guide (ICLOCS). . Department of Electrical Engineering, Imperial College London, London, UK, 2010.

[32] W. Fan, Y. Liao, J.B. Lee, and Y.K. Kim. Evaluation of Two Lagrangian Dual Optimization Algorithms for Large-Scale Unit Commitment Problems. *Journal of Electrical Engineering & Technology*, 7(1):17–22, 2012.

[33] P. Ferreira. *Electricity power planning in portugal: The role of wind energy*. PhD thesis, University of Minho, 2008.

[34] Dalila B. M. M Fontes, Fernando A. C. C. Fontes, and Luís A.C. Roque. Optimal Control Formulations for the Unit Commitment Problem. *Dynamics of Information Systems: Computational and Mathematical Challenges. C. Vogiatzis, J. L. Walteros , P. M. Pardalos (eds). Springer*, 2014.

[35] Dalila B.M.M. Fontes and António Gaspar-Cunha. On multi-objective evolutionary algorithms. In *Handbook of Multicriteria Analysis*, pages 287–310. Springer, 2010.

[36] Dalila B.M.M. Fontes and José Fernando Gonçalves. Heuristic solutions for general concave minimum cost network flow problems. *Networks*, 50(1):67–76, 2007.

[37] Dalila B.M.M. Fontes and José Fernando Gonçalves. A multi-population hybrid biased random key genetic algorithm for hop-constrained trees in nonlinear cost flow networks. *Optimization Letters*, pages 1–22, 2012.

[38] Fernando A.C.C. Fontes, Dalila B.M.M. Fontes, and Luís A.C. Roque. An optimal control approach to the unit commitment problem. In *CDC - Conference on Decision and Control*, pages 7069–7074, Maui, USA, 2012.

[39] A. Frangioni and C. Gentile. Solving nonlinear single-unit commitment problems with ramping constraints. *Operations Research*, 54(4):767–775, 2006.

[40] A. Frangioni, C. Gentile, and F. Lacalandra. Solving unit commitment problems with general ramp constraints. *Electrical Power and Energy Systems*, 30(5):316–326, 2008.

[41] A. Frangioni, C. Gentile, and F. Lacalandra. Tighter Approximated MILP Formulations for Unit Commitment Problems. *IEEE Transactions on Power Systems*, 24(1):105–113, 2009.

[42] M.R. Gent and J.W. Lamont. Minimum-emission Dispatch . *IEEE Transactions on Power Apparatus and Systems*, 78:2650–2660, 1971.

[43] M. Gerdts. Solving mixed-integer optimal control problems by branch and bound: a case study from automobile test driving with gear shift. *Optim. Contr. Appl. Methods*, 26:1–18, 2005.

[44] M. Gerdts. A variable time transformation method for mixed-integer optimal control problems. *Optim. Contr. Appl. Methods*, 27:169–182, 2006.

[45] J.F. Gonçalves, J.J. M. Mendes, and M.G.C. Resende. A genetic algorithm for the resource constrained multi-project scheduling problem. *European Journal Operational Research*, 189(3):1171–1190, 2008.

[46] J.F. Gonçalves and M.G.C. Resende. A parallel multi-population genetic algorithm for a constrained two-dimensional orthogonal packing problem. *Journal of Combinatorial Optimization*, 22(2):180–201, 2011.

[47] J.F. Gonçalves and M.G.C. Resende. Biased random-key genetic algorithms for combinatorial optimization. *Journal of Heuristics*, 17(5):487–525, 2011.

[48] M.M. Hadji and B. Vahidi. A Solution to the Unit Commitment Problem Using Imperialistic Competition Algorithm . *IEEE Transactions on Power Systems*, 27(1):117–124, 2012.

[49] H. Hamedi. Solving the combined economic load and emission dispatch problems using new heuristic algorithm. *Electrical Power and Energy Systems*, 46:10–16, 2013.

[50] J. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, 1975.

[51] J. Horn, N. Nafpliotis, and D.E. Goldberg. A Niched Pareto genetic algorithm for multiobjective optimization. In *1st IEEE Conf. Evol. Comput., IEEE World Congr. Comput. Intell.*, volume 1, pages 67–72, 1994.

[52] B. Houska, H. J. Ferreau, and M. Diehl. Acado toolkit: An open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298–312, 2011.

[53] K. Huang, H. Yang, and C. Yang. A new thermal unit commitment approach using constraint logic programming. *IEEE Transactions Power Systems*, 13(3), 1998.

[54] A.D. Ioffe and V.M. Tihomirov. *Theory of Extremal Problems*, volume 6 of *Studies in Mathematics and its Applications*. North-Holland Publishing Company, Amsterdam, New York, Oxford, 1979.

[55] B. Jeddi and V. Vahidinasab. A modified harmony search method for environmental/economic load dispatch of real-world power systems . *Energy Conversion and Management*, 78:661–675, 2014.

[56] Y.W. Jeong, J.B. Park, J.R. Shin, and K.Y. Lee. A thermal unit commitment approach using an improved quantum evolutionary algorithm. *Electric Power Components and Systems*, 37(7):770–786, 2009.

[57] Ruiwei Jiang, Jianhui Wang, and Yongpei Guan. Robust unit commitment with wind power and pumped storage hydro. *IEEE Transactions on Power Systems*, 27(2):800–810, 2012.

[58] K.A. Juste, H. Kita, E. Tanaka, and J. Hasegawa. An evolutionary programming solution to the unit commitment problem. *IEEE Transactions on Power Systems*, 14(4):1452–1459, 1999.

[59] J. Kallrath, P.M. Pardalos, S. Rebennack, and M. Scheidt. *Optimization in the Energy Industry* . Energy Systems, Springer, 2009.

[60] S.A. Kazarlis, A.G. Bakirtzis, and V. Petridis. A Genetic Algorithm Solution to the Unit Commitment Problem. *IEEE Transactions on Power Systems*, 11:83–92, 1996.

[61] E. Khmelnitsky. A combinatorial, graph-based solution method for a class of continuous-time optimal control problems. *Mathematics of Operations Research*, 27:312–325, 2002.

[62] I.S. Kotsireas, C. Koukouvinos, P.M. Pardalos, and D.E. Simos. Competent genetic algorithms for weighing matrices. *Journal of Combinatorial Optimization*, 24(4):508–525, 2012.

[63] S. Kuloor, G.S. Hope, and O.P. Malik. Environmentally Constrained Unit commitment . In *IEE Proc. Gener. Trans. Distr., 139, 122-128*, 1992.

[64] T.W. Lau, C.Y. Chung, K.P. Wong, T.S. Chung, and S.L. Ho. Quantum-Inspired Evolutionary Algorithm Approach for Unit Commitment. *IEEE Transactions on Power Systems*, 24(3):1503–1512, 2009.

[65] G.S. Lauer, N.R. Sandell, D.P. Bertsekas, and T.A Posbergh. Solution of large scale optimal unit commitment problems. *IEEE Transactions on Power Apparatus and Systems*, PAS-101(1):79–96, 1982.

[66] F.N. Lee. The coordination of multiple constrained fuels. *IEEE Transactions on Power Systems*, 6:699–707, 1991.

[67] F.N. Lee. Short-term unit commitment-a new method. *IEEE Transactions on Power Systems*, 3(2):421–428, 1998.

[68] H. W. J. Lee, K. L. Teo, and X. Q. Cai. An optimal control approach to nonlinear mixed integer programming problems. *Computers and Mathematics with Applications*, 36:87–105, 1998.

[69] H. W. J. Lee, K. L. Teo, V. Rehbock, and L.S. Jennings. Control parameterization enhancing technique for time optimal control problems. *Dynamic Systems and Applications*, 6:243–262, 1997.

[70] H. W. J. Lee, K. L. Teo, V. Rehbock, and L.S. Jennings. Control parametrization enhancing technique for optimal discrete valued control problems. *Automatica*, 35:1401–1407, 1999.

[71] Yan-Fu Li, Nicola Pedroni, and Enrico Zio. A memetic evolutionary multi-objective optimization method for environmental power unit commitment. *Power Systems, IEEE Transactions on*, 28(3):2660–2669, 2013.

[72] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang. Environmental/economic dispatch problem of power system by using an enhanced multi-objective differential evolution algorithm. *Energy Conversion and Management*, 52:1175–1183, 2011.

[73] A.H. Mantawy, Y.L. Abdel-Magid, and S.Z. Selim. Unit commitment by tabu search. *IEE Proc. Gener. Trans. Distr.*, 145(1):56–64, 1998.

[74] A.H. Mantawy, L. Abdel-Magid Youssef, and Z. Selim Shokri. A simulated annealing algorithm for unit commitment. *IEEE Transactions on Power Systems*, 13(1):197–204, 1998.

[75] H. Meunier, E.G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. In *Proceedings of the 2000 Congress on Evolutionary Computation(CEC'00), IEEE Press*, pages 317–324, 2000.

[76] Z. Michalewicz. Genetic algorithms, numerical optimization, and constraints. In *in: L. Eshelman (Ed.), Proceedings of the Sixth International Conference on Genetic Algorithms, San Mateo,*, pages 151–158. Morgan Kaufmann Publishers, 1995.

[77] Z. Michalewicz and C.Z. Janikow. Handling Constraints in Genetic Algorithms. In *In Belew, R.K., Booker, L.B., eds.: Proceedings of the Fourth International Conference on Genetic Algorithms (ICGA-91), San Mateo, California, University of California, San Diego*, pages 151–157. Morgan Kaufmann Publishers, 1991.

[78] M. Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press., 1996.

[79] J.A. Muckstadt and S.A. Koenig. An application of Lagrangian relaxation to scheduling in power-generation systems . *Operations Research*, 25(3):387–403, 1977.

[80] T. Niknam, H.D. Mojarrad, and B.B. Firouzi. A new optimization algorithm for multi-objective Economic/Emission Dispatch. *Electrical Power and Energy Systems*, 46:283–293, 2013.

[81] N.P. Padhy. Unit commitment using hybrid models: a comparative study for dynamic programming, expert system, fuzzy system and genetic algorithms. *International Journal of Electrical Power & Energy Systems*, 23(8):827–836, 2000.

[82] N.P. Padhy. Unit Commitment-A Bibliographical Survey. *IEEE Transactions on Power Systems*, 19(2):1196–1205, 2004.

[83] V.R. Pandi, B.K. Panigrahi, W.C. Hong, and R. Sharma. A Multiobjective Bacterial Foraging Algorithm to Solve the Environmental Economic Dispatch Problem. *Energy Sources, Part B: Economics, Planning, and Policy*, 9:236–247, 2014.

[84] N. Pandit, A. Tripathi, S. Tapaswi, and M. Pandit. An improved bacterial foraging algorithm for combined static/dynamic environmental economic dispatch. *Applied Soft Computing*, 12:3500–3513, 2012.

[85] S. Patra, S.K. Goswami, and B. Goswami. Fuzzy and simulated annealing based dynamic programming for the unit commitment problem. *Expert Systems with Applications*, 36(3):5081–5086, 2009.

[86] D. Powell and M.M. Skolnick. Using genetic algorithms in engineering design optimization with nonlinear constraints. In *Proceedings of the Fifth International Conference on Genetic Algorithms, San Mateo, California, July 1993, University of Illinois at Urbana Champaign*, pages 424–431. Morgan Kaufmann Publishers, 1993.

[87] I. J. Raglend and N. P. Padhy. Comparison of Practical Unit Commitment Solutions. *Electric Power Components and Systems*, 36(8):844–863, 2008.

[88] I. Jacob Raglend, Sowjanya Veeravalli, Kasanur Sailaja, B. Sudheera, and D.P. Kothari. Comparison of ai techniques to solve combined economic emission dispatch problem with line flow constraints. *International Journal of Electrical Power & Energy Systems*, 32(6):592–598, 2010.

[89] C.C.A. Rajan and M.R. Mohan. An evolutionary programming-based tabu search method for solving the unit commitment problem. *IEEE Transactions on Power Systems*, 19(1):577–585, 2004.

[90] S. Rebennack, B. Flach, M.V.P. Pereira, and P.M. Pardalos. Stochastic Hydro-Thermal Scheduling under CO2 Emission Constraints. *IEEE Transactions on Power Systems*, 27(1):58–68, 2012.

[91] S. Rebennack, P.M. Pardalos, M. V.F. Pereira, and N.A. Iliadis. *Handbook of Power Systems I*. Energy Systems, Springer, 2010.

[92] S. Rebennack, P.M. Pardalos, M. V.F. Pereira, and N.A. Iliadis. *Handbook of Power Systems II*. Energy Systems, Springer, 2010.

[93] C. R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems*. chapter Genetic Algorithms, Blackwell Scientific Publications, 1993.

[94] A. Rong, H. Hakonen, and R. Lahdelma. A variant of the dynamic programming algorithm for unit commitment of combined heat and power systems. *European Journal Operational Research*, 190:741–755, 2008.

[95] L. Roque, D. B. M. M Fontes, and F. A. C. C. Fontes. A Biased Random Key Genetic Algorithm Approach for Unit Commitment Problem. *Lecture Notes in Computer Science*, 6630(1):327–339, 2011.

[96] L. A. C. Roque, D. B. M. M. Fontes, and F. A. C. C. Fontes. A Hybrid Biased Random Key Genetic Algorithm Approach for the Unit Commitment Problem. *Journal of Combinatorial Optimization*, 28:140–166, 2014.

[97] Luís A. C. Roque, Dalila B. M. M. Fontes, and Fernando A. C. C. Fontes. BRKGA adapted to multi-objective unit commitment problem Solving Pareto frontier for UC multi-objective Problem using BRKGA, SPEA2 , NPGA and NSGA II techniques. In *ICORES - International Conference on Operations Research and Enterprise Systems*, Vilamoura, Portugal, 2012.

[98] S. Salam. Unit commitment solution methods. In *Proceedings of World Academy of Science, Engineering and Technology*, volume 26, pages 600–605, 2007.

[99] B. Saravanan, D.A.S. Siddharth, S. Sikri, and D.P. Kothari. A solution to the unit commitment problem - a review. *Frontiers in Energy*, 7(2):223 – 236, 2013.

[100] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of multiobjective optimization*. Orlando: Academic Press, 1985.

[101] Frank Schneider, Diego Klabjan, and Ulrich Thonemann. Incorporating demand response with load shifting into stochastic unit commitment. *Available at SSRN 2245548*, 2013.

[102] S. Sen and D.P. Kothari. Optimal thermal generating unit commitment: a review. *Electrical Power and Energy Systems*, 20:443–451, 1998.

[103] B. Shaw, V. Mukherjee, and S.P. Ghoshal. A novel opposition-based gravitational search algorithm for combined economic and emission dispatch problems of power systems. *Electrical Power and Energy Systems*, 35:21–33, 2012.

[104] A. Siburian. *Numerical methods for robust, singular and discrete valued optimal control problems*. Ph.d. thesis, Curtin University of Technology, Perth, Australia, 2004.

[105] D.N. Simopoulos, S.D. Kavatza, and C.D. Vournas. Unit commitment by an enhanced simulated annealing algorithm. *IEEE Transactions on Power Systems*, 21(1):68–76, 2006.

[106] K. Sourirajan, L. Ozsen, and R. Uzsoy. A genetic algorithm for a single product network design model with lead time and safety stock considerations. *European Journal Operational Research*, 197(2):38–53, 2009.

[107] L. Sun, Y. Zhang, and C. Jiang. A matrix real-coded genetic algorithm to the unit commitment problem. *Electric Power Systems Research*, 76:716–728, 2006.

[108] K.S. Swarup and S. Yamashiro. Unit Commitment Solution Methodology Using Genetic Algorithm. *IEEE Transactions on Power Systems*, 17:87–91, 2002.

[109] S. Takriti and J.R. Birge. Using Integer Programming To Refine Lagrangian-Based Unit Commitment Solutions. *IEEE Transaction on Power Systems*, 15(1):151–156, 2000.

[110] K.L. Teo, L.S. Jennings, and V. Rehbock. The control parameterization enhancing transform for constrained optimal control problems. *Journal of the Australian Mathematics Society*, 40:314–335, 1999.

[111] T. Ting, M.V.C. Rao, C.K. Loo, and S.S. Ngu. Solving Unit Commitment Problem Using Hybrid Particle Swarm Optimization. *Journal of Heuristics*, 9:507–520, 2003.

[112] Anupam Trivedi, Naran M. Pindoriya, Dipti Srinivasan, and Deepak Sharma. Improved multi-objective evolutionary algorithm for day-ahead thermal generation scheduling. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 2170–2177. IEEE, 2011.

[113] Anupam Trivedi, N.M. Pindoriya, and Dipti Srinivasan. Modified NSGA-II for day-ahead multi-objective thermal generation scheduling. In *IPEC, 2010 Conference Proceedings*, pages 752–757. IEEE, 2010.

[114] C. Tseng. *On power system generation unit commitment problems*. Ph.d. thesis, Department of Industry Engineering and Operations Research, University of California at Berkeley, California, USA, 1996.

[115] A. Turgeon. Optimal Scheduling of Thermal Generating Units. *IEEE Trans. on Automatic Control*, 23(6):100–105, 1978.

[116] B.C. Ummels. *Power System Operation with Large-Scale Wind Power in Liberalised Environments*. PhD thesis, Delft University of Technology, 2009.

[117] J. Valenzuela and A.E. Smith. A seeded memetic algorithm for large unit commitment problems. *Journal of Heuristics*, 8(2):173–195, 2002.

[118] T. Venkatesan and M.Y. Sanavullah. SFLA approach to solve PBUC problem with emission limitation. *Electrical Power and Energy Systems*, 46:1–9, 2013.

[119] B. Venkatesh, T. Jamtsho, and H. Gooi. Unit commitment - a fuzzy mixed integer linear programming solution. *IET Generation, Transmission and Distribution*, 1(5):836–846, 2007.

[120] A. Viana and J.P. Pedroso. A new MILP-based approach for unit commitment in power production planning. *Electrical Power and Energy Systems*, 44(1):997–1005, 2013.

[121] T.A.A. Victoire and A.E. Jeyakumar. A tabu search based hybrid optimization approach for a fuzzy modelled unit commitment problem. *Electric Power Systems Research*, 76:413–425, 2006.

[122] C. Wang and S.M. Shahidehpour. Effects of ramp-rate limits on unit commitment and economic dispatch. *IEEE Transactions on Power Systems*, 8(3):1341–1350, 1993.

[123] Qianfan Wang, Yongpei Guan, and Jianhui Wang. A chance-constrained two-stage stochastic program for unit commitment with uncertain wind power output. *IEEE Transactions on Power Systems*, 27(1):206–215, 2012.

[124] S. Wang, M. Shahidehpour, D.S. Kirschen, S. Mokhtari, and G. Irissari. Short-term generation scheduling with transmission and environmental constraints us-

ing an augmented lagrangian relaxation. *IEEE Trans Power Systems*, ‘10:1294–300, 1995.

[125] D. Whitley. A genetic algorithm tutorial. *Statistics and Computing*, 4(2):65–85, 1994.

[126] G. Winter, D. Greiner, B. Gonzalez, and B. Galvan. Economical and environmental electric power dispatch optimisation. 2003.

[127] Bruce Wollenberg and Allen Wood. Power generation, operation and control. *editorial John Wiley & Sons, USA*, 1996.

[128] J. Xie, J. Zhong, Z. Li, and D. Gan. Environmental-economic unit commitment using mixed-integer linear programming. *European transactions on electrical power*, 21(1):772–786, 2011.

[129] D. Yamashita, T. Niimura, R. Yokoyama, and M. Marmiroli. Trade-off analysis of CO2 versus cost by multi-objective unit commitment. In *Power and Energy Society General Meeting, 2010 IEEE*, pages 1–6. IEEE, 2010.

[130] H.Y. Yamin, Q. El-Dwairi, and S.M. Shaihidehpour. A new approach for GenCo profit based unit commitment in day-ahead competitive electricity markets considering reserve uncertainty. *Int J Elec Power Energy Systems*, 29:609–16, 2007.

[131] C. Yasar and S. Ozyon. Solution to scalarized environmental economic power dispatch problem by using genetic algorithm. *Electrical Power and Energy Systems*, 38:54–62, 2012.

[132] Y. Zhang, D.G. Gung, and Z. Ding. A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch. *Information Sciences*, 192:213–227, 2012.

[133] B. Zhao, C.X. Guo, B.R. Bai, and Y.J. Cao. An improved particle swarm optimization algorithm for unit commitment. *International Journal of Electrical Power & Energy Systems*, 28(7):482–490, 2006.

[134] Q.P. Zheng, S. Rebennack, P.M. Pardalos, and M.V.F. Pereira. *Handbook of CO2 in Power Systems (Energy Systems)*. Springer, 2012.

[135] Q.P. Zheng, J. Wang, P.M. Pardalos, and Y. Guan. A decomposition approach to the two-stage stochastic unit commitment problem. *Annals of Operations Research*, 210(1):387–410, 2013.

[136] E. Zio, P. Baraldia, and N. Pedroni. Optimal power system generation scheduling by multi-objective genetic algorithms with preferences. *Reliability Engineering and System Safety*, 94:432–444, 2009.

[137] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8:173–195, 1999.

[138] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-Rep. 103*, 2001.

[139] E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. *TIK-Rep.*, 43, 1998.

[140] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. Evol. Comput.*, 3:257–271, 1999.