



UNIVERSIDAD DE EXTREMADURA
Escuela Politécnica
Departamento de Tecnología de los Computadores y
de las Comunicaciones

Doctor of Philosophy Dissertation

APLICANDO METAHEURÍSTICAS
MULTIOBJETIVO AL PROBLEMA DE
ASIGNACIÓN DE FRECUENCIAS EN
REDES GSM

*Applying Multiobjective Metaheuristics to the Frequency
Assignment Problem in GSM Networks*

Autor
Marisa da Silva Maximiano

Director
Dr. Miguel Ángel Vega Rodríguez

Cáceres, 2011

***Edita: Universidad de Extremadura
Servicio de Publicaciones***

Caldereros 2. Planta 3^a
Cáceres 10071
Correo e.: publicac@unex.es
<http://www.unex.es/publicaciones>

Acknowledgments

There are an enormous number of people who I would like to express my gratitude to. Firstly, to my thesis director, Dr. Miguel Ángel Vega Rodríguez, for leading me on the thesis objectives. His ability to give me numerous valuable suggestions, exchange of knowledge, excellent ideas throughout all the important steps of the research project was fundamental. I also recognize his amazing capacity of encouragement along this hard journey.

I would also like to thank my family for the support they have provided me through my entire life and in particular over these last years of my research, specially because I have been less present in their lives.

A very special thanks goes out to all my closest friends without whose motivation, understanding and encouragement I would not have accomplished this hard working task. Their support, suggestions and encouragement throughout these last years have given me courage enough for keeping on working when I needed it the most. Fortunately, some of them are indeed my coworkers with whom I have had many fruitful discussions and suggestions, which were fundamental to finish this thesis.

I would also like to thank numerous entities that in some way provided partial financial or logistic support for all related activities:

- The Polytechnic Institute of Leiria, through their Innovation, Development and Science Association - AIDC (IPL).
- The Department of Computer Engineering of the School of Management and Technology of Leiria - DEI (<http://www.dei.estg.ipleiria.pt/>) ESTG.
- The Portuguese Science and Technology Foundation (FCT).
- The Spanish Ministry of Science and Innovation and FEDER under the contract TIN2008-06491-C04-04 (the M* Project).

Abstract

The Frequency Assignment (FAP) is a problem that has risen in the telecommunication area in the last decades, due to the enormously growth of the mobile users. That is the reason why it has been so important, especially in the GSM (Global System for Mobile Communications) networks since it represents one of the most important and critical tasks for current (and future) mobile communication operators.

The main problem in FAP is that in GSM networks a fixed frequency spectrum is available, but it is also very scarce. Therefore frequencies need to be reused throughout the network, and consequently, interferences may occur and some separation constraints may be violated. Therefore, the FAP aims to minimize these unwanted occurrences, in order to provide the network users with a good quality of service. Due to the usable range of frequency spectrum being very limited, the FAP becomes very important to guaranty the traffic performance on mobile communications. As a result, in the GSM technology FAP is under the most relevant and significant problems.

In the above context, the need to develop new strategies in order to address this problem becomes extremely important. Generally speaking, the key ambition is to study and develop different approaches based on new and also hybrid metaheuristics. In fact, to deal with the realistic problem we work with a complex mathematical formulation, which takes in consideration the requirements of real-world GSM networks. The main contribution of this work is a study developed to solve FAP using distinct Evolutionary Algorithms (EA). Specifically, it considers the use of population-based metaheuristics and trajectory-based metaheuristics. EAs are a class of stochastic search and optimization methods. Their algorithms are based on the principles of natural biological evolution, and have received an increasing interest in the last decades. EAs encompass a range of different methods inspired in the natural evolution, including Genetic Algorithms, Genetic Programming, and also other related techniques, e.g., Differential Evolution (DE), etc.

First of all, several already known metaheuristics have been designed and adapted to be analyzed with a single objective formulation of FAP. Initially,

our goal was the application of several metaheuristics considering only the level of interferences occurring in the network. Following this, and considering that solving real-life engineering problems it is not an easy task because normally they represent a multiobjective optimization scenario, a more complex formulation is being investigated. Indeed, with FAP, two different and conflicting objectives that characterize a multiobjective optimization (MO) problem can be considered, in which an assignment of frequencies is sought to a high number of transmitters in a way as efficient as possible. Therefore, we have designed a more complex formulation of FAP considering it as a multiobjective optimization problem. This new multiobjective formulation of FAP considers as objectives to improvement of the interference costs and the separation costs.

Although we have started with a single-objective formulation, our main contribution in addressing FAP was focused on comparing different multiobjective metaheuristics, specially comparing population-based metaheuristics against the ones based on trajectory. Evaluating the performance of multiobjective algorithms is far from being a trivial task. Therefore, we have incorporated several complementary measurements to assess the quality of the accomplished solutions. First of all, the validation of results uses the common statistical comparisons and after that we used the Hypervolume indicator that defines the volume of the objective space dominated by the Pareto Front. Besides the Hypervolume, the Coverage Relation was also used as additional metrics in order to assess which the best Pareto Front is.

Two real-world instances of FAP, currently being operated, will be used in the experiments scenario. Comparisons with other well-known algorithms are also performed in order to assess the efficiency of our approaches.

Resumen

El problema de la asignación de frecuencias (en inglés, Frequency Assignment Problem, FAP) es un problema que ha ganado importancia en el área de las telecomunicaciones durante las últimas décadas, debido al enorme crecimiento de los usuarios de comunicaciones móviles. Esta es la razón por la que es tan importante, especialmente en las redes GSM (en inglés, Global System for Mobile communications), representando una de las tareas más críticas e importantes para los actuales (y futuros) operadores de comunicaciones móviles.

La principal dificultad en el FAP es que en las redes GSM existe disponible un espectro fijo de frecuencias, que es además muy escaso. Por consiguiente, las frecuencias tienen que ser reutilizadas a lo largo de la red, y por tanto, pueden producirse interferencias y se pueden violar ciertas restricciones de separación. En conclusión, al resolver el FAP se busca minimizar estos dos efectos no deseados, proporcionando a los usuarios de la red una buena calidad de servicio. Debido a que el rango utilizable del espectro de frecuencias es muy limitado, el FAP se ha convertido en un aspecto muy importante para garantizar el rendimiento del tráfico en las comunicaciones móviles. Como resultado, en la tecnología GSM, el FAP está entre los problemas más relevantes y significativos.

Dado el contexto anterior, la necesidad de desarrollar nuevas estrategias para solventar este problema llega a ser extremadamente importante. Hablando en general, la ambición clave es estudiar y desarrollar distintas aproximaciones basadas en nuevas y también híbridas metaheurísticas. De hecho, para tratar de forma realista este problema, hemos trabajado con una formulación matemática compleja, que considera los requisitos de las redes GSM del mundo real. La principal contribución de este trabajo es el estudio desarrollado para resolver el FAP usando distintos algoritmos evolutivos (en inglés, Evolutionary Algorithms, EAs). Más concretamente, se han considerado metaheurísticas basadas en población y metaheurísticas basadas en trayectoria. Los EAs son una clase de métodos de optimización y búsqueda estocástica. Sus algoritmos están basados en los principios de la evolución biológica natural, y han recibido un creciente interés en las últimas décadas. Los EAs abarcan un amplio rango de distintos métodos inspirados en la evolución natural, que incluyen algoritmos genéticos, programación genética, y

también otras técnicas relacionadas, como por ejemplo, evolución diferencial (en inglés, Differential Evolution, DE), etc.

En este trabajo, en primer lugar, varias metaheurísticas ya conocidas han sido diseñadas y adaptadas para ser analizadas con una formulación mono-objetivo del FAP. Inicialmente, nuestro objetivo fue la aplicación de diversas metaheurísticas considerando únicamente el nivel de interferencias que ocurren en la red. Tras esto, hemos tenido en cuenta que resolver problemas de ingeniería de la vida real no es una tarea fácil, puesto que normalmente representan escenarios de optimización multiobjetivo, por lo que hemos investigado una formulación más compleja del FAP. De hecho, en el FAP, se pueden considerar dos objetivos distintos y conflictivos entre sí, caracterizando un problema de optimización multiobjetivo (en inglés, Multiobjective Optimization, MO), en el que se busca una asignación de frecuencias a un gran número de transmisores-receptores de la forma más eficiente posible. En conclusión, hemos diseñado una formulación más compleja del FAP que lo considera como un problema MO. Esta nueva formulación multiobjetivo del FAP considera como objetivos la mejora en los costes de interferencias y los costes de separación.

Aunque hemos iniciado nuestra investigación con una formulación mono-objetivo, nuestra principal contribución a la hora de resolver el FAP se centra en comparar distintas metaheurísticas multiobjetivo, especialmente comparando metaheurísticas basadas en población y metaheurísticas basadas en trayectoria. Evaluar el rendimiento de algoritmos multiobjetivo no es una tarea trivial. Por tanto, hemos incorporado varias métricas complementarias para evaluar la calidad de las soluciones conseguidas. En primer lugar, la validación de los resultados utiliza las comparaciones estadísticas habituales, y después usamos el indicador del hipervolumen que define el volumen del espacio objetivo dominado por el frente de Pareto. Además del hipervolumen, también hemos utilizado la relación de cobertura como métrica adicional para evaluar cuál es el mejor frente de Pareto.

Dos instancias FAP del mundo real, actualmente operativas, han sido utilizadas en los escenarios de los experimentos. Además, comparaciones con otros algoritmos bien conocidos y con otros autores también han sido realizadas para evaluar la eficiencia de nuestras propuestas.

Contents

Acknowledgments	i
Abstract	iii
Resumen	v
Contents	vii
List of Symbols and Abbreviations	xi
List of Figures	xiii
List of Tables	xv
List of Algorithms	xvii
1 Introduction	1
1.1 Introduction and Motivation	1
1.1.1 Working Plan and Main Milestones	3
1.2 Organization of this Dissertation	5
2 The Frequency Assignment Problem	7
2.1 General Overview	8
2.1.1 Frequency Planning: Global Concepts	8
2.1.2 Classification of the Frequency Assignment Problem	10
2.1.3 Standard FAP Representation	11
2.2 FAP in GSM Networks	14
2.2.1 Structure of a GSM network	14
2.2.2 Frequency planning in GSM	17
2.3 Multiobjective Formulation of FAP	19
2.3.1 The Interference Cost	20
2.3.2 The Separation Cost	21
2.4 Summary	22

3 FAP Literature Review	23
3.1 Frequency Planning: Research Approaches	23
3.1.1 Test Problems Over FAP	24
3.2 Metaheuristics for FAP: A Summary	26
3.2.1 Related Work in GSM Networks	28
3.3 Multiobjective Optimization: Applied Research	29
3.4 Summary	30
4 Single-objective Metaheuristics on FAP	33
4.1 Metaheuristics Base Approach	34
4.2 Problem Domain Information Incorporated in the Algorithms	36
4.2.1 Problem Domain Knowledge	36
4.2.2 FAP Local Search Heuristic	37
4.3 Differential Evolution Algorithm	38
4.3.1 Intrinsic Control Parameters	39
4.3.2 Differential Evolution Strategies	41
4.3.3 The Classic Algorithm Structure	41
4.3.4 Hybrid Differential Evolution Approach	43
4.4 Variable Neighborhood Search Algorithm	46
4.4.1 Algorithm Structure	47
4.5 Skewed Variable Neighborhood Search Algorithm	48
4.5.1 Algorithm Structure	48
4.6 Summary	49
5 Multiobjective Metaheuristics: New Models	51
5.1 Introduction	52
5.2 Multiobjective Optimization: Basic Concepts	52
5.2.1 Multiobjective Formulation	53
5.2.2 Dominance Concept	53
5.3 Evaluating the Quality of Solutions in Multiobjective FAP	56
5.4 Differential Evolution with Pareto Tournaments Algorithm	57
5.4.1 Algorithm Structure	57
5.4.2 Pareto Tournaments	58
5.5 Multiobjective Variable Neighborhood Search Algorithm	60
5.5.1 Algorithm Structure	60
5.5.2 Mutation Operator	62
5.6 Multiobjective Skewed Variable Neighborhood Search Algorithm	62
5.6.1 Algorithm Structure	63
5.6.2 Distance Function	63
5.7 Greedy Versions of MO-VNS and MO-SVNS Algorithms	64
5.7.1 Greedy Mutation Operator	65
5.8 Multiobjective Artificial Bee Colony	66

5.8.1	Standard ABC Algorithm	66
5.8.2	Pseudocode of the MO-ABC Algorithm	67
5.8.3	Pareto Tournament	69
5.9	Other Multiobjective Metaheuristics: Comparative Overview	71
5.9.1	Nondominated Sorting Genetic Algorithm - II	71
5.9.2	Strength Pareto Evolutionary Algorithm 2	73
5.10	Multiobjective Metaheuristics Classification	74
5.11	Summary	75
6	Experimental Results	77
6.1	Real-world Problem Instances	78
6.1.1	Characteristics of the GSM Instances	78
6.1.2	Instances Configuration Files	80
6.2	Codification of the Problem	84
6.2.1	Individual Representation	84
6.2.2	Creating the Initial Solution	86
6.2.3	Considerations Regarding the Software that was Developed	87
6.3	Methodology and Metrics	89
6.3.1	Experimental Conditions	89
6.3.2	Experiments on Tune Metaheuristics Parameters	90
6.3.3	Multiobjective Performance Indicators	91
6.4	Single-objective Metaheuristics Empirical Results	94
6.4.1	Configuration Parameters: Experimental Study	94
6.4.2	Parameterization	104
6.4.3	Empirical Results: Comparisons Among Algorithms	106
6.5	Multiobjective Metaheuristics Empirical Results	111
6.5.1	Configuration Parameters: Experimental Study	111
6.5.2	Parameterization	144
6.5.3	Results Comparison Analysis	144
6.5.4	Multiobjective Conclusions	151
6.6	Summary	153
7	Conclusions and Future Work	155
7.1	Main Achievements in the Investigation	155
7.2	Future Work	158
7.3	Publication Record and Scientific Merits	159
7.4	Scientific Events	161
Bibliography		163
A Software Development		181
A.1	Metaheuristics Applications	181

A.2 Graphic Interface to Multiobjective Optimization Indicators	184
Index	191

List of Symbols and Abbreviations

Abbreviation	Description	Definition
ABC	Artificial Bee Colony	page 66
ACO	Ant Colony Optimization	page 34
AFP	Automatic Frequency Planning	page 7
CALMA project	Combinatorial ALgorithms for Military Applications project	page 24
CAP	Channel Assignment Problem	page 7
CDMA	Code Division Multiple Access	page 14
CELAR	Centre d'Electronique de l'ARMement France	page 24
DCA	Dynamic Channel Assignment	page 10
DE	Differential Evolution	page 38
DEPT	Differential Evolution with Pareto Tournaments algorithm	page 57
EA	Evolutionary Algorithms	page 29
EC	Evolutionary Computing	page 27
EMC	ElectroMagnetic Compatibility	page 9
FAP	Frequency Assignment Problem	page 7
FCA	Fixed Channel Assignment	page 10
GRAPH	Generating Radio Link Frequency Assignment Problems Heuristically	page 24
GMO-VNS	Greedy Multiobjective Variable Neighborhood Search	page 64
GMO-SVNS	Greedy Multiobjective Skewed Variable Neighborhood Search	page 64
GSM	Global System for Mobile Communications	page 14
HV	Hypervolume indicator	page 91
ILS	Iterated Local Search	page 34
IM	Interference Matrix	page 11
LS	Local Search	page 37
LSHR	Local Search with Heuristic Restart	page 105
MDI	Multiple Document Interface	page 185
MO	Multiobjective Optimization	page 52
MO-ABC	Multiobjective Artificial Bee Colony	page 66

Abbreviation	Description	Definition
MO-SVNS	Multiobjective Skewed Variable Neighborhood Search algorithm	page 62
MO-VNS	Multiobjective Variable Neighborhood Search algorithm	page 60
NP-hard	Non-Deterministic Polynomial-Time Hard	page 9
NSGA-II	Nondominated Sorting Genetic Algorithm - II	page 71
QoS	Quality of Service	page 8
RND	Radio Network Design	page 28
SA	Simulated Annealing	page 34
SPEA2	Strength Pareto Evolutionary Algorithm 2	page 73
SS	Scatter Search	page 105
SVNS	Skewed Variable Neighborhood Search	page 48
TDMA	Time Division Multiple Access	page 14
TS	Tabu Search	page 34
VNS	Variable Neighborhood Search	page 46
XML	eXtensible Markup Language	page 87

List of Figures

2.1	Channel assignment with 4 cells and 11 channels.	12
2.2	FAP representation of the C matrix and demand vector D.	13
2.3	GSM architecture.	15
2.4	Schema of the components existing in a sector.	16
2.5	GSM network example.	17
4.1	Metaheuristics classification.	35
4.2	Classic DE algorithm.	43
5.1	The dominance concept in the two-objective case.	54
5.2	Classification of our multiobjective metaheuristics approaches.	75
6.1	Topology of the GSM Denver instance.	79
6.2	Topology of the GSM Seattle instance.	80
6.3	Configuration file containing the data about the TRX.	81
6.4	Configuration file with the sectors data.	81
6.5	Configuration file with the data about the sites.	82
6.6	Configuration file with the IM data.	82
6.7	Configuration file containing data related to the network options.	82
6.8	Configuration file about the 1-order neighbors in the network instance.	83
6.9	Configuration file about the 2-order neighbors in the network instance.	83
6.10	Solution encoding.	85
6.11	Base classes definition to address FAP.	86
6.12	XML configuration files example.	88
6.13	The hypervolume representation considering two-objective functions.	93
6.14	DE results for the Local Search method tuning.	96
6.15	Results for the NP values obtained with Denver instance.	96
6.16	Evolution results for the CR values obtained with Denver instance.	97
6.17	Evolution results for the CR values obtained with Denver instance.	99
6.18	DE results using the best-so-far selection strategies.	100
6.19	DE results using the random selection strategies.	100
6.20	Evolution results between the most efficient strategies.	103
6.21	α experiment results from the SVNS algorithm (Denver instance).	103

6.22 Comparison results between the three metaheuristics used.	107
6.23 Pareto Fronts for NP values of DEPT using Denver instance.	112
6.24 Pareto Fronts for NP values of DEPT using Seattle instance.	112
6.25 Pareto Fronts using DEPT random-base-schemes/*/bin in Denver. . .	116
6.26 Pareto Fronts using DEPT best-base-schemes/*/bin in Denver. . . .	116
6.27 Pareto Fronts using DEPT random-base-schemes/*/exp in Denver. .	118
6.28 Pareto Fronts using DEPT best-base-schemes/*/exp in Denver. . . .	118
6.29 Pareto Fronts using DEPT random-base-schemes/*/bin in Seattle in- stance.	119
6.30 Pareto Fronts using DEPT best-base-schemes/*/bin in Seattle instance.	120
6.31 Pareto Fronts using DEPT random-base-schemes/*/exp in Seattle in- stance.	120
6.32 Pareto Fronts using DEPT best-base-schemes/*/exp in Seattle instance.	121
6.33 α parameter results of MO-SVNS algorithm with Denver.	121
6.34 α parameter results of MO-SVNS algorithm with Seattle instance. .	122
6.35 Pareto Fronts for mutation probability of MO-ABC over Denver in- stance.	124
6.36 Pareto Fronts for mutation probability of MO-ABC over Seattle in- stance.	126
6.37 Pareto Fronts for PM parameter of NSGA-II over Denver instance. .	131
6.38 Pareto Fronts for PM parameter of NSGA-II over Seattle instance. .	132
6.39 Pareto Fronts for CR operator of NSGA-II over Denver.	134
6.40 Pareto Fronts for CR operator of NSGA-II over Seattle.	134
6.41 Pareto Fronts for PC parameter of SPEA2 over Denver instance. .	137
6.42 Pareto Fronts for PC parameter of SPEA2 over Seattle instance. .	139
6.43 Pareto Fronts for PM parameter of SPEA2 over Denver instance. .	141
6.44 Pareto Fronts for PM parameter of SPEA2 over Seattle instance. .	142
6.45 Pareto Fronts for $CxOp$ parameter of SPEA2 over Denver instance. .	143
6.46 Pareto Fronts for $CxOp$ parameter of SPEA2 over Seattle instance. .	143
6.47 Pareto Fronts comparing various metaheuristics over Denver. . . .	146
6.48 Pareto Fronts comparing various metaheuristics over Seattle. . . .	147
A.1 Solution	182
A.2 Command line interface on Windows to launch DEPT algorithm. . . .	183
A.3 Configuration interface for the Hypervolume reference points. . . .	186
A.4 Graphic interface showing the Pareto Front solutions and the related data.	187
A.5 Graphic interface showing two separated Pareto Front solutions and and the coverage relation information.	188
A.6 Graphic interface showing the coverage relation information.	189
A.7 Graphic interface showing the coverage relation information and their algorithms Pareto Fronts.	190

List of Tables

2.1	Constraints summary.	13
4.1	Notation for the Differential Evolution schemes.	42
6.1	DE results for the Local Search methods.	95
6.2	Population size results for DE algorithm.	97
6.3	Crossover probability results for DE algorithm.	98
6.4	Mutation operator results for DE algorithm.	98
6.5	Results with the Binomial crossover strategies in DE algorithm.	101
6.6	Results with the exponential crossover strategies in DE algorithm.	102
6.7	α results for SVNS algorithm.	104
6.8	DE, VNS and SVNS results with Denver instance.	106
6.9	Comparison results with the single-objective metaheuristics.	110
6.10	Results for the DEPT crossover parameter.	114
6.11	Results for the DEPT mutation intensity (F) parameter.	115
6.12	Results for the DEPT strategies with both instances.	117
6.13	α parameter results in <i>greedy</i> MO-SVNS.	123
6.14	Results for <i>colony size</i> parameter of MO-ABC.	125
6.15	Results for <i>limit</i> parameter of MO-ABC.	126
6.16	Results for the mutation probability of MO-ABC.	127
6.17	Results for the population size parameter tuning in NSGA-II.	128
6.18	Results for the crossover parameter tuning in NSGA-II.	130
6.19	Results for the population size (N) parameter in SPEA2.	136
6.20	Results for the crossover probability (PC) parameter in SPEA2.	138
6.21	Comparison between multiobjective algorithms with Denver instance.	147
6.22	Comparison between multiobjective algorithms with Seattle.	147
6.23	Hypervolume values obtained by the metaheuristics for both instances.	149
6.24	Coverage relation results among multiobjective metaheuristics.	150

List of Algorithms

1	Pseudocode for the Differential Evolution algorithm.	44
2	Pseudocode for Hybrid Differential Evolution algorithm.	45
3	Pseudocode for Variable Neighbourhood Search algorithm.	47
4	Pseudocode for Skewed Variable Neighbourhood Search algorithm.	49
5	Pseudocode for DEPT algorithm.	58
6	Pseudocode for the MO-VNS algorithm.	61
7	Pseudocode for the MO-SVNS algorithm.	64
8	Pseudocode for <i>greedy mutation in environment</i> function.	65
9	Detailed pseudocode for Multiobjective Artificial Bee Colony (MO-ABC) algorithm.	67
10	Pseudocode for the mutation operator of MO-ABC.	68
11	Pseudocode for the main function of NSGA-II algorithm.	73
12	Pseudocode for SPEA2 algorithm	74

Introduction

1.1 Introduction and Motivation

During the last decade the tremendous success of mobile phone systems has caused some significant technological advances as well as the investigation of mathematical models and optimization algorithms to support the planning of mobile networks. Frequency planning is one of the phases in the deployment of these types of systems.

Global System for Mobile (GSM) is one of the most popular systems for mobile communications. Indeed, by mid 2010 GSM services were used by more than 3 billion subscribers across 220 countries, representing approximately 85% of the world's cellular market [56]. For a mobile operator the frequency spectrum is one of the scarcest resources, leading to the necessity of assuring that their reuse leads to an increased capacity of the network without compromising the quality of service (QoS) for the end users.

In GSM the planning problem can be subdivided into two distinct sub-problems: coverage planning, in which the antennas are located so as to maximize service coverage, and capacity planning, in which frequencies are assigned to the antennas so as to maximize a measure of the overall quality of the received signals [6]. Our work is focused on the second scenario and on concepts and models which are relevant on the current frequency planning on GSM networks. FAP has no universal mathematical model and therefore it can assume several formulations, depending on the spectrum size, technology specification, etc. We focus on the GSM standard and a realistic formulation of FAP was adopted, which was for-

mulated by [92]. This formulation takes full advantage of realistic and accurate interference information from real GSM networks. Due to the usable range of frequency spectrum being very limited, the FAP becomes very important to ensure the traffic performance on mobile communications. FAP intends to obtain an assignment of channels to base stations, in a way that the quality of service is guaranteed. The FAP is in charge of the optimum use of the frequency spectrum allocated to the cellular mobile systems. Hence, the service area of the system is divided into a large number of cells with customers simultaneously requesting channels for their communication. Because of this, the system must accommodate each and every customer request (or call), by assigning a channel (spectrum frequency), while satisfying the constraints imposed to avoid the radio interference among channels assigned to the same cell or in relatively near adjacent cells, which are known as ElectroMagnetic Compatibility constraints (EMC). Under this limitative context of restrictions, mobile operators still need to guarantee an admissible communication infrastructure to the end users.

Initially, an overview of some of the most significant optimization metaheuristics that have risen when planning this generation cellular networks is presented and they clearly demonstrate the importance of this research subject, for which we describe the main corresponding mathematical models, and we briefly mention some of the computational approaches that have been devised to tackle them.

The main goals of this thesis project consists in the development of new solutions to deal with a real-world Frequency Assignment problem in the context of GSM networks. Therefore, the core of the work consists in designing and applying different types of metaheuristics to this telecommunication problem. Two distinct approaches of FAP formulations were investigated: (1) a single-objective FAP formulation and then (2) a multiobjective formulation was proposed, which was then analyzed. For both formulations, several metaheuristics were analyzed and comparisons were made. In the first scenario, to improve the efficiency of the several enhanced techniques, some hybrid algorithms were also proposed and used. In the second scenario we have designed new metaheuristics to address the multiobjective FAP formulation. These new proposed metaheuristics were compared against several well-known multiobjective metaheuristics. The relevance of our contribution considers the importance of assessing the efficiency of the several metaheuristics to solve FAP in the two perspectives that we have adopted for this research project.

In the specific case of GSM networks, FAP had already started being investigated. Therefore, some metaheuristics have already been applied to a single-objective FAP formulation. Our contribution in this researching area is possible because no work about FAP, using the metaheuristics proposed in here, has been found so far in the literature. Indeed, to the best of our knowledge the meta-

heuristics that we are using in both FAP formulations represent the first attempt to test the effectiveness of these metaheuristics, when addressing our real-world instances of FAP.

Our work is focused on the design, implementation and test of the metaheuristics performance when addressing this specific FAP scenario. It is also relevant to perform comparisons against several population-based and trajectory-based metaheuristics. Although we have used some already known metaheuristics, we also propose a multiobjective formulation of FAP, based on the mathematical model that we have adopted. The work developed under this approach represents a novelty in this applied research.

The importance and the benefits of this investigation project can be viewed according to several perspectives:

- At first, and regarding the area in which the problem rises, with this work we can find a new contribution for the telecommunication research area and also for people working in multiobjective optimization. Even if multiobjective algorithms have been used in many real-world problems (a well-studied field), the results that will be obtained represent an improvement in the frequency assignment problem.
- Scientifically, it implies the development of new metaheuristics that can be applied in the future by other authors working in multiobjective algorithms. Indeed, other techniques were investigated and the PhD candidate has developed new strategies to address a multiobjective approach of FAP.
- From an industrial point of view, this research project involves the study of a real problem applied to telecommunications and mobile networks. These networks have an increasing importance.
- Considering the country point of view, this problem is part of a national research project with the objective of being applied into the Spanish telecommunication and mobile networks. If we enhanced the current solutions, this could be a major benefit to the telecommunication systems.

1.1.1 Working Plan and Main Milestones

As already mentioned, the main goals of our research are the development and performance analysis of metaheuristics in the optimization of frequency planning in GSM networks, which is a hard problem in the design and planning of a mobile network. Our major accomplishment consists in developing and assessing the ability of distinct single-objective and multiobjective metaheuristics to solve FAP in the more efficient way possible. For that reason a single-objective study of

FAP, as well as multiobjective formulation were performed. Therefore, for these two types of FAP formulations several metaheuristics were implemented and their parameters were tuned to the best configuration possible. As a consequence, from the wide range of metaheuristics approached in our research subject, some of the multiobjective metaheuristics are new and were developed by us to address the multiobjective formulation of FAP that was followed in this work.

To achieve the above objectives, the following tasks were performed:

- In the first phase the most relevant papers were read to acknowledge the Frequency Assignment problem and the characteristics of the several metaheuristics that have been applied over the years to address the several formulations that have emerged under this research subject.
- Implementation of different mechanisms and techniques using distinct evolutionary algorithms to solve the Frequency Assignment problem in GSM Networks. These outcomes have permitted to achieve the first partial results. This task requires collecting and analyzing the data to be used to solve the addressed problem (FAP) in a real-world instance of a GSM Network. It also includes the implementation of hybrid metaheuristics to optimize the solutions accomplished to the problem. In this phase it is considered the identification and the tuning of the parameters that are required by the analyzed metaheuristics and their impact on performance.
- Optimizing the implemented algorithms exploring optimization methods such as Variable Neighborhood Search algorithm, Skewed Variable Neighborhood, etc. Analysis and tests, using realistic-sized real-world instances of FAP, in order to validate the performance of the applied metaheuristics. Indeed, first evolutionary algorithms are used, and with their outcomes, the next step is to test alternative approaches and compare them with other previous investigations (and with other authors).
- Formulating FAP as a multiobjective optimization problem and then designing and implementing new metaheuristics capable of solving the multiobjective FAP over GSM networks. Evaluating the developed algorithms with a possible extent of the mathematical model to deal with more advanced issues in GSM frequency plans. The new metaheuristics incorporate concepts associated with multiobjective optimization; therefore new indicators for evaluating the results need to be implemented.
- Validation of the achieved results using well-known indicators in order to give results with a high confidence level. The analysis of these latest results is also made using real-world scenarios. Throughout the analysis of the multiobjective outcomes accomplished for the several implemented approaches

adopted to solve the multiobjective FAP it should be possible to develop and propose new enhanced features to optimize the followed approaches. Furthermore, multiobjective implementations for FAP using popular multi-objective metaheuristics will be investigated, e.g the NSGA-II and SPEA2, some of the most well-known multiobjective metaheuristics. Finally, the outcomes of the several investigated algorithms are used to compare the existent metaheuristics with the new designed metaheuristics. The final report aims to highlight the most relevant results and appraise the efficiency on addressing the Frequency Assignment problem.

1.2 Organization of this Dissertation

The rest of this dissertation is organized as follows: In Chapter 2 we cover the Frequency Assignment problem area and give a theoretical perspective of the problem. We start by presenting a generalist perspective of FAP and then we focus on the specific scenario of FAP in GSM networks. A mathematical formulation of FAP considering it as a multiobjective optimization problem is also given.

In Chapter 3 we present some background information about the research line followed in the previous decades using FAP classic formulation. Basically, it approaches the techniques investigated over the recent years and the developments performed in the field. Despite the review made for FAP in which we present a wide range of metaheuristics that were applied to the problem, we also present the special scenario of FAP in GSM networks and give a perspective of how it has been applied under that type of mobile networks.

Chapter 4 presents some modeling techniques to address FAP formulation in GSM networks. This Chapter is entirely dedicated to describing the features and structure of the metaheuristics that were implemented to solve a single-objective formulation of FAP under GSM networks. We also describe some of the hybridizations made to the adopted algorithm. The metaheuristics were selected in order to cover the two classes of metaheuristics: the trajectory-based metaheuristics and the population-based metaheuristics.

Chapter 5 models some new multiobjective metaheuristics designed to be applied in addressing FAP. Furthermore, some basic concepts related to multiobjective optimization problems are also presented and we specify how they were incorporated on the metaheuristics that we have developed. Nevertheless, as for the first initial FAP formulation, also in the multiobjective FAP research we specify trajectory-based metaheuristics and population-based metaheuristics.

Chapter 6 is concerned with experiments performed with the proposed multiobjective metaheuristics developed by us, as well as other well-known opti-

mization multiobjective metaheuristics. Additionally, we also present the results obtained with our first approach using a single-objective metaheuristics that was implemented to address FAP. In the experiments scenario, two real-world GSM networks were considered. Therefore, they represent two large-scale instances of GSM networks and demonstrate the relevance of the problem as a real engineering optimization problem. This chapter ends with a comparison of the performances of all the metaheuristics.

Finally, Chapter 7 presents the main conclusions and future work outlines.

The Frequency Assignment Problem

In this chapter, we address the main concepts related to the Frequency Assignment Problem (FAP) [57, 111], which is one of the key issues in the design of cellular networks. The FAP is also known by other denominations as Automatic Frequency Planning (AFP) or Channel Assignment Problem (CAP) [39]. In our work, we focus on solving FAP in the Global System for Mobile Communication (GSM) networks context. As a result a FAP formulation representing this kind of mobile networks is our key issue, although we have included some of the basic concepts used by benchmark problems. In Frequency Planning, the available frequency spectrum is assigned to the physical equipment that grants users the access to the network services.

First, we outline the more common concepts related to frequency planning and some of the constraints that need to be handled when tackling this optimization problem. FAP is explained considering the two commonly used scenarios. The first one is more related to benchmark problems, which was the starting scenario when addressing FAP in cellular radio networks. The second one uses real-world GSM networks. Thus, we also provide a brief description of the GSM system components as well as the definition of the frequency assignment problem in the GSM context.

We finalize the chapter by describing the mathematical model used to address the real-world instances of FAP that we have adopted in our study. Our formulation of FAP outlines the problem as a multiobjective optimization problem.

2.1 General Overview

The Frequency Assignment Problem is a challenging task that rises while developing a digital mobile telecommunication network. The main goal is to serve the maximum number of network users, but considering having only available a limited number of transmission resources. FAP is a well-known NP-hard combinatorial problem [57], which was initially formalized as an instance of the Graph Coloring problem [47]. Since it represents a real-world engineering problems it has been extensively analyzed [3, 40, 86]. Depending on the particular network, the understanding of frequency assignment varies. Therefore, FAP has no universal mathematical model and therefore it can assume several formulations, depending on the spectrum size, technology specifications, etc. Our work has its main focus on the GSM standard.

Several different problem types are included under the global FAP denomination, and therefore several mathematical models have been proposed [3] in the last years. Our work is based on concepts and models which are relevant for the current real-world GSM frequency planning [38], explained in Section 2.2.

There are two major tasks when developing a mobile network: the Site Position Problem (SPP) and the Frequency Assignment Problem (FAP). The first task consists in placing the antennas in the area where the mobile telecommunication network has to be developed. There are many different factors that influence the positioning of the antennas. Information such as the number of possible mobile users and their distributions is used to position the antennas. The terrain morphology is another aspect to be considered.

In the design of mobile networks, antennas should be placed and distributed in such a way as to yield an appropriate quality of service (QoS) to users. In order to provide a mobile service it is also required to predict the number of frequencies that are needed in every antenna to maximize the number of users that the network can hold. After having handled the distribution of antennas, the next step is to assign frequencies to every antenna. That operation is known as Frequency Planning.

2.1.1 Frequency Planning: Global Concepts

In the Frequency Planning design phase of mobile networks the number of available frequencies is very limited and considerably smaller than the total number of required frequencies by all the antennas in the network. Because frequencies are very scarce, they need to be reused. This can disturb the service quality of the network, namely because this can provoke undesired interferences if the electromagnetic constraints (explained below) are not met.

Thus, when assigning frequencies to the antennas, several type of constraints need to be satisfied. Generally, the hard constraints to be satisfied are the interference constraints, known as the ElectroMagnetic Compatibility (EMC) constraints. The *interference constraints* are classified into three distinct types that need to be satisfied to avoid the radio interferences [129], namely as:

co-site constraint: defines that frequencies assigned in the same antenna must have a minimal distance in the frequency spectrum between each other.

co-channel constraint: limits that equal frequencies may be assigned only to antennas which are at a minimum distance from each other. Otherwise the frequencies would interfere with each other.

adjacent-channel constraint: it is similar to the previous one, but considering adjacent frequencies regarding the frequency spectrum.

The EMC constraints are normally determined by the characteristics of the radio frequency propagation and the spatial density of the expected traffic requirements.

Besides the interferences constraints explained above, other two types of constraints are classified under FAP category. They are identified as: the *frequency constraints* that specify the range or number of available frequencies in the radio spectrum, and they are imposed by an international regulation; and the *traffic constraints* that indicate the minimum number of frequencies required by each station [33]. The traffic constraints data is provided by telecommunications operator and they represent the more limitative constraints that needed to be fulfilled when addressing FAP. *

Different optimization versions of FAP are available, representing different classifications under FAP denomination. Thus, several approaches have been developed such as: maximizing the total traffic, minimizing the number of frequencies used and minimizing the interferences over the network, is all possible. The common characteristic is the fact that all of them represent a problem that can be formulated as a NP-hard problem [47, 57]. It has been shown that this problem is equivalent to a generalized graph-coloring problem, which is NP-hard.

A problem is NP-hard if an algorithm for solving it can be translated into one for solving any NP-problem (nondeterministic polynomial time) problem. NP-hard therefore means "at least as hard as any NP-problem," although it might, in fact, be harder. †

*In our case, we are addressing FAP using network information that is available and it is defined in several configuration files. This information is used by the metaheuristics developed to conduct the empirical study.

†<http://mathworld.wolfram.com/NP-HardProblem.html>

Summarizing, FAP definition sustains that the general objective is to assign a number of available frequencies to a set of cells (antennas), in which the assignment process is subject to a set of constraints in such a way that the required bandwidth (number of used frequencies) is minimized [128]. As a result, a NP-hard characteristic is implicit in solving the problem. For that reason, in the last decades the propose of several formulations and also a high level of metaheuristics [59] have shown to provide FAP solutions with accurate results [6]. Next, different optimization variants of FAP are outlined.

2.1.2 Classification of the Frequency Assignment Problem

The Frequency Assignment Problem is commonly classified as fixed or as dynamic. In the Fixed Channel Assignment (FCA), the channels are normally assigned to cells in advance. This assignment is made according to the pre-determined estimated traffic intensity. In the Dynamic Channel Assignment (DCA), the channels are assigned dynamically as requests arrive [39].

The dynamic method can make the cellular systems more efficient, especially if the traffic distribution is unknown, or if it has many changes along the time. Regardless of these advantages it has the disadvantages of requiring a more complex control and being more time consuming [66, 128].

Despite of dynamic assignment present better performances, it is not always the best option because the fixed channel assignment has a better performance when used in scenarios with a heavy traffic load conditions [120].

In this context and due to the fact that in GSM network a significant increase of users, provoking an increase of traffic demands has happened [56], it implies that fixed channel assignment still is a very current problem, with a significant practical application. Hence, with the increase of user's demands it is extremely important to provide a high efficient usage of the available spectrum. For that reason, solving FAP under the fixed channel approach is the main focus of our work. This decision is also supported by the fact that DCA is not supported in GSM, so we only consider FCA.

Four different classification variants of FAP are available when addressing the problem as a Fixed Channel Assignment problem [39].

- Minimum Order Frequency Assignment (MO-FAP).
- Minimum Span Frequency Assignment (MS-FAP).
- Minimum Blocking Frequency Assignment (MB-FAP).
- Minimum Interference Frequency Assignment (MI-FAP).

In a simpler way, the MO-FAP intends to minimize the number of used frequencies when assigning the frequencies, regardful of the aim in minimizing the unacceptable interferences.

The MS-FAP also intends to minimize the occurrence of unacceptable interferences, but considering the need to minimize the difference between the maximum and the minimum used frequencies. Therefore, the span is minimized.

In the MI-FAP it is intended to minimize the total sum of interference levels when assigning the frequencies from a limited number of available frequencies. In plenty situations the MI-FAP is applied as a sub procedure to find the minimum span of FAP.

Finally, the MB-FAP is considered in situations where the assignment contains unacceptable interferences and therefore it is tried to minimize the overall blocking probability of the network when it is tried to assign frequencies.

2.1.3 Standard FAP Representation

In the past years several formulations have been proposed when tackled FAP in a cellular radio network [2, 34, 128]. A few ways to specify the constraint of FAP are possible, but in 1982, Gamst and Rave [47] defined a well common form to the FAP problem in an arbitrary inhomogeneous cellular radio network. In FAP it is necessary to fulfill the frequency constraints, the traffic constraints and the interference constraints represented by the EMC. The most severe are indeed the ones related to the limited number of available frequencies. These are the limitations that impose the need to reuse the frequencies by the several base stations, consequently increasing the necessity and hardness of satisfying the interference constraints.

The interference constraints are the central focus of FAP. The prior formulations of FAP [16, 37, 39] consider that such constraints can be summarized by a symmetric interference matrix, called ElectroMagnetic Compatibility matrix (C) in a n -cell network where it is described by a $n \times n$ symmetric matrix. An element (c_{ij}) specifies the required minimal distance between frequencies (frequency spectrum) assigned to an arbitrary pair of cells when one of the pair is demanded in the i -th and the other in the j -th cell. Therefore, each non-diagonal element c_{ij} in C represents the minimum separation distance in the frequency domain between a frequency assigned to cell $\#i$ and a frequency assigned to cell $\#j$. The co-channel constraint is represented by $c_{ij} = 1$, and the adjacent channel constraint is represented by $c_{ij} = 2$. The value $c_{ij} = 0$ indicates that cells $\#i$ and $\#j$ are allowed to use the same frequency value. Each diagonal element c_{ii} in C represents the minimum separation distance between any two frequencies assigned to cell $\#i$, which represents the co-site constraint, where $c_{ii} > 1$ is always respected.

$$C = \begin{bmatrix} 5 & 4 & 0 & 0 \\ 4 & 5 & 0 & 1 \\ 0 & 0 & 5 & 2 \\ 0 & 1 & 2 & 5 \end{bmatrix} \quad D = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 3 \end{bmatrix}$$

Figure 2.1: Channel assignment with 4 cells and 11 channels.

The channel requirements for each cell in the n -cell network are described by an n -element vector, which is called demand vector D . Each element in d_i in D represents the number of frequencies to be assigned to the cell # i . Where f_{ik} indicates the k -th frequency assigned to cell # i , the electromagnetic compatibility constraints are represented by:

$$|f_{ik} - f_{jl}|, \text{ for } i, j = 1, \dots, n; k = 1, \dots, d_i; l = 1, \dots, d_j; \text{ for all } i \neq j, k \neq l \quad (2.1)$$

The problem consists in finding a conflict-free channel assignment, i.e., the f_{ik} such that the bandwidth required by the system, that is the $\max_{i,k}\{f_{ik}\}$, is minimized.

As an example, Figure 2.1 shows the FAP problem scenario, adopted from [28] for 4 cells and 11 channels, where the C and D value represent the compatibility matrix and the demand vector, respectively. Table 2.1 summarizes the interpretation of these requirements.

The difference between each cell is their traffic demand value. The cells #1, #2 and #3, denoted as $d_1=1$, $d_2=1$, $d_3=1$ respectively only require one radio frequency. But, the cell #4, denoted as $d_4=3$ requires three frequencies. These traffic demands can be represented as $\sum_{m=1}^M f_{mi} = d_i$, where the value M represents the maximum number of frequency spectrum.

The number of required frequencies can change from problem to problem. For the above example, it can be calculated as $c_{ii} * (d_i - 1) + 1$, where c_{ii} is the maximum value of the c_{ii} in C and the d_i is the maximum value of d_i in D .

From the example, a frequency within distance 4 from the frequency assigned to cell # i cannot be assigned to cell #2, because of $c_{12} = c_{21} = 4$. In the same way, also any two frequencies assigned to cell #4 must have at least distance 5 because of $c_{44} = 5$. The diagonal element $c_{ii} = 5$ means that in order to satisfy the co-site constraints, all the channels assigned at i must have a separation of at least five frequencies distance. In a similar way, the non-diagonal element c_{ij} (e.g. $c_{21} = 4$) states that channel assignment to cell #2 and cell #1 must have a separation of at least four frequencies, i.e. the channel (frequency) 2 will be

	Cell #1	Cell #2	Cell #3	Cell #4
Traffic demand	1 channel	1 channel	1 channel	3 channels
Minimum frequency separation distance	co-site = 5 to cell 2 = 4	co-site = 5 to cell 1 = 4	co-site = 5 to cell 4 = 2	co-site = 5 to cell 3 = 2 to cell 2 = 1

Table 2.1: Constraints summary.

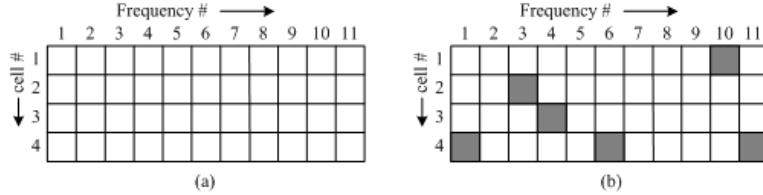


Figure 2.2: Representation for FAP of the C matrix and demand vector D. (a) 4x11 representation. (b) corresponds to the optimum solution to the four-cell channel assignment.

assigned to cell #1, hence it will be necessary to assign at least the channel 6 to cell #2.

In this example, the minimum number of the total frequencies is 11 because cell #4 requires at least 11 ($=1+5 \times 2$) frequencies. In the Figure 2.2, the optimal solution to this problem is shown, where the frequency 10 is assigned to cell #1, frequency 3 to cell #2, frequency 4 to cell #3, and frequency 1, 6 and 11 to cell #4.

In the most common approach of frequency assignment problem it is only considered the co-channel constraint. Nevertheless, the optimal frequency assignment scheme involves the determination of the minimum required channel span and the way to distribute the frequencies among the cells satisfying the interference and the required traffic constraints.

Besides the electromagnetic compatibility constraints it is also defined a set of antennas (base stations), a set of available frequencies and the requirements. The requirement vector defines how many frequencies need to be assigned to every base station.

This basic formulation is the foundation for the remaining evolutions that have been emerging when addressing FAP in other types of telecommunication networks. Indeed, it represents the formulation that is followed by the more common benchmark problems available for FAP.

In the last years, the basic FAP formulation has been extended in order to tackle real-world mobile networks. In real-world models, all types of possible

sources of interferences need to be considered, as well as regulatory concerns, and technology limitations [38].

Currently, because of the large traffic demand and the reduced frequency spectrum, it is impossible to find an interference free frequency assignment. Therefore, the work is currently done trying to formulate FAP in such a way that it can make possible to obtain frequency plans that minimize the overall level of interferences in the network, maximizing simultaneously the quality of service. Although many of the contributions are being done using benchmark-like problems, our work is focused on real-world networks and therefore it uses more realistic and accurate interference information from GSM networks [92].

2.2 FAP in GSM Networks

The Global System for Mobile Communication (GSM) is one of the most popular systems for mobile communications [110]. GSM defines a standard from the telecommunication area and it is used in mobile communications. Indeed, by mid 2009 GSM services were used by more than 3 billion subscribers across 220 countries, representing approximately 85% of the world's cellular market [56]. In the last decade, the number of mobile users has grown enormously. For a mobile operator the frequency spectrum is one of the scarcest resources, leading to the necessity of assuring that their reuse leads to an increased capacity of the network without compromising the QoS for the end users. In this context, the formulation of FAP in a real-world scenario is an important contribution.

Several different problem types make use of the general terms used by FAP. Indeed, some mathematical models have been proposed in the last decades to address it [2]. In addition to the FAP terminology commonly used by benchmark-like problems, in GSM networks other concepts are more frequently used to describe their architecture and to formulate the problem constraints.

2.2.1 Structure of a GSM network

Figure 2.3 outlines the structure of a GSM network. The GSM is a standard for mobile communications. Specifically, it is a digital mobile telephone system that uses a variation of time division multiple access. It is the most widely used of the three digital wireless telephone technologies (TDMA[‡], GSM, and CDMA[§]).

[‡]Time Division Multiple Access (TDMA) is a technology used in digital cellular telephone communication that divides each cellular channel into three time slots in order to increase the amount of data that can be carried [2].

[§]Code Division Multiple Access (CDMA) refers to any of several protocols used in so-called second-generation (2G) and third-generation (3G) wireless communications. As the term implies, CDMA is a form of multiplexing, which allows numerous signals to occupy a single transmission channel, optimizing the use of available bandwidth. (see <http://www.infosecwriters.com/>)

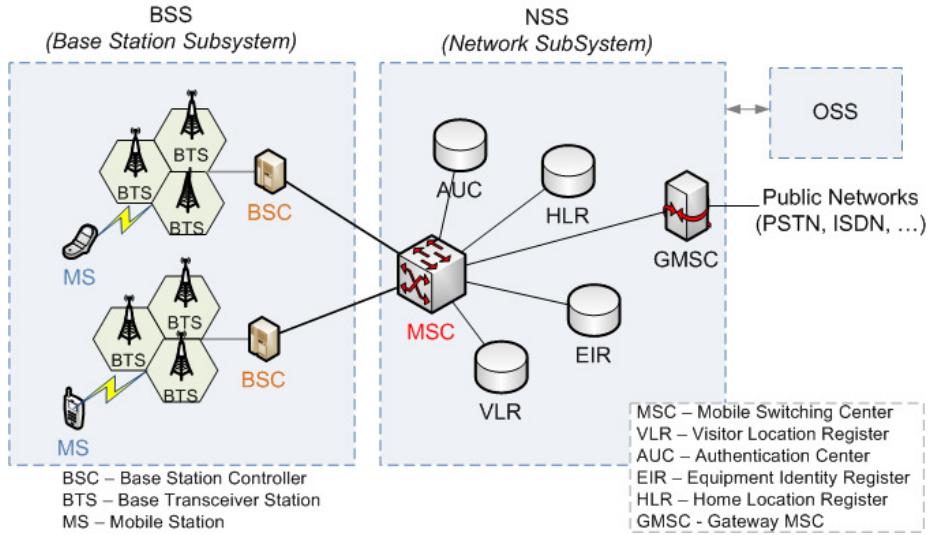


Figure 2.3: GSM architecture.

A GSM system contains many different components. In fact, as shown in Figure 2.3, a GSM system can be divided into three major systems:

- Base Station Subsystem (BSS).
- Network and Switching Subsystem (NSS).
- Operation and Maintenance Subsystem (OSS).

NSS is responsible for performing call processing and subscriber-related functions. It includes several functional units: HLR, MSC, VLR, AUC, EIR. All radio-related functions are performed in the BSS subsystem. NSS is composed by Base Station Controllers (BSCs) and Base Transceiver Stations (BTSs). The main function of BTS is to handle the radio interface to the mobile station (MS). The BTS is the radio transceiver needed in the network to serve each cell in the network. As we can see in Figure 2.3, the architecture of GSM networks has many different components, but the most relevant in FAP formulation are the Base Transceiver Station (BTS) and the Transceivers (TRX).

Mobile communication is dependent on a radio link between a user's mobile terminal and the base stations (BTS) on the network. After the placement of the base stations, the next fundamental phase is the selection and configuration of their antennas. The configuration is performed in order to provide desired network coverage.

When addressing FAP, the most relevant components in a GSM infrastructure are essentially BTSs, inside which a set of transceivers (TRXs) have been

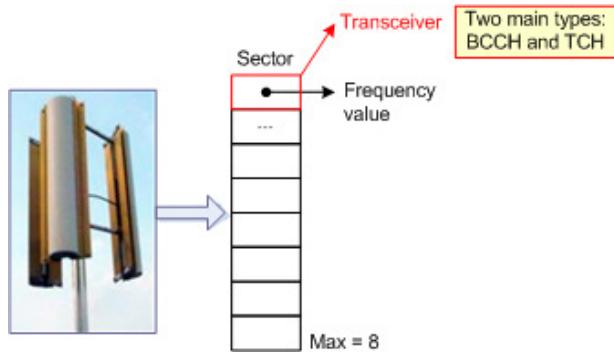


Figure 2.4: Schema of the components existing in a sector.

installed, either broadcast control channel (BCCH) or traffic channel (TCH). A base station typically serves three different sectors, or cells. The number of TRXs to be installed inside each sector is fixed and it depends on the traffic demand the sector has to support.

In each sector several transceivers are installed (visible in Figure 2.4). In the instances of the problem that were used, each sector only has eight transceivers: one transceiver of type BCCH and seven of type TCH. The number of available TCH transceivers within a sector is variable, but in our scenario we have used instances that only have seven. The number of TRXs installed in each sector depends on the traffic demand of the network (traffic that the sector needs to support). The sector is similar to the concept of cell, which was addressed in the previous section.

In Figure 2.5 it is given a network example that is formed by several *sites* and each one has three *sectors* installed (e.g. sector A operates the A1, A2 and A3). In this topology, each sector has a list of neighbors associated that can potentially generate interferences between each others. The neighbor sectors can be categorized as first order neighbors or second order neighbors [91].

Finalizing, in the formulation of a GSM frequency planning, the network architecture is composed mainly by a BTS that has a set of TRXs. The transceivers task is to make possible the conversion between the digital traffic data on the network side and radio communication between the mobile terminal and the GSM network [92]. The *site* in which the BTS is positioned is formed by *sectors*: the most typical is the existence of one to three sectors (see Figure 2.5). Here, a cell corresponds to the area in which each sector operates.

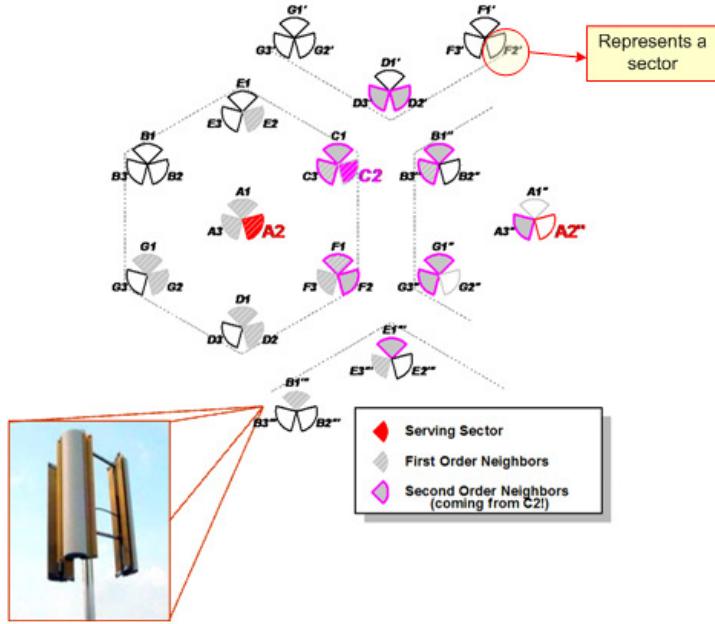


Figure 2.5: GSM network example.

2.2.2 Frequency planning in GSM

Frequency Planning is the last phase in the layout of a GSM network. The GSM network is built with different sectors that operate in a specific service area. The number of necessary TRXs to be installed per sector has to be fixed, and it depends on the traffic demand that the sectors need to sustain.

The quality of the service (QoS) offered to the users of that serving area is guaranteed by the frequencies assigned to the several TRXs installed in the sector of the serving area. The quality of the service relies on the number of interferences prevailing in the network.

Technically, the available frequency band is put into channels (or frequencies) which have to be allocated to the elementary transceivers (TRXs) installed in the base stations (BTS) of the network. Each transceiver uses a frequency slot, also called channel. Nearby transceivers have to use different channels. An operator has to reuse its channels multiple times to operate enormous amounts of transceivers in the network due to limited radio spectrum. Unfortunately, significant interferences between transmitters using the same channel (co-channel) or an adjacent channel may occur. Consequently, service quality of the GSM network degrades itself. FAP tries to properly allocate a frequency to each and every TRX of the network to minimize such undesired interferences.

Thus, a valid channel (frequency) from the available spectrum needs to be assigned to each TRX that composes the network. In our real-world GSM frequency planning formulation, the problem consists in the assignment of a channel (a frequency) to every TRX, considering however that frequencies should be reused by several TRX of the network. However, two distinct goals corresponding to the interferences cost and the separation cost need to be considered simultaneously.

Caused by the assignment of frequencies to the TRXs, several interferences may occur if the same frequency or adjacent frequencies are used in neighboring cells. They are referred to as co-channel and adjacent-channel interferences, which were already explained. To avoid the nasty effect of the interferences in the GSM network, constraints are defined and they indicate the degree of closeness that the frequencies in the same TRX may have. They were formally represented in the section formulating FAP as separation constraints.

Summarizing, in the frequency planning phase of a GSM network, the possible constraint separation types available are: the co-site separation constraint, when the transceivers are installed in the same site, or the co-cell separation, when the transceivers serve the same cell (they are installed in the same sector). In the mathematical formulation of FAP it is fundamental to consider the effects of existing interferences. Also important are the constraints associated to the assignment of the frequencies inside each sector serving the several areas of the network.

Mathematical Formulation

To clearly understand the proposed formulation, it is fundamental to know all the terminology used in a GSM network topology (previously described in the above sections). Next, it is presented the formulation used to represent the real-world instance of the GSM networks.

To simplify the understanding of the applied formulation, we use the following notation, which is also adopted by other researchers [92, 93]:

- $T = \{t_1, t_2, \dots, t_n\}$ is the set of n transceivers.
- $F = \{f_{i1}, f_{i2}, \dots, f_{ik}\} \subset N$ is the set of *valid frequencies* of transceiver $t_i \in T, i = 1, \dots, n$. K is the $k - th$ of F_i . It is not necessarily the same for all the transceivers.
- $S = \{s_1, s_2, \dots, s_m\}$ is the set of given sectors of cardinality m .
- Each transceiver $t_i \in T$ is installed at exactly one of the m sectors.
- $s(t_i) \in S$, denotes the sector in which the transceiver t_i is installed.

- $M = \{(\mu_{ij}, \sigma_{ij})\}_{m \times m}$ is the interference matrix.

Then, the T is a set of transceivers and F represents the set of available frequencies that can be assigned to each transceiver. These values change depending on what transceiver it is associated with, therefore the number of the available spectrum of frequencies (F_i) in each transceiver it is not always the same. Additionally, a set of sector S (or cells) of cardinality m is also available. Each transceiver is installed in one of the available sectors denoted by m .

The two values, from the interference matrix μ_{ij} and σ_{ij} , represent an entry in $M(i, j) = (\mu_{ij}, \sigma_{ij})$ and are the mean and the standard derivation of a Gaussian probability distribution, respectively. The Gaussian distribution describes the carrier-to-interference ratio (C/I) when the sector i and j operate on a same frequency, whenever both sectors have assigned the same frequency value. Less interference is represented by a higher value for the mean (μ), implying a better quality in the communication.

The interference matrix is the key point of the fitness function and is defined at a sector level. Therefore, all the transceivers installed in the same sectors serve the same area. The fitness function to this real-world instance evaluates the effect (interference) of the frequencies assigned to each transceiver installed in each sector. It penalizes in a more severe manner the provoked interferences caused by frequencies installed in the same sector, because they have a great influence in the quality of service of that serving area.

A solution p to the problem consists in the assignment of a frequency from F_i to each transceiver $t_i \in T$. Therefore a frequency plan (solution) is represented by $p \in F_1 \times F_2 \times \dots \times F_n$, where $p(t_i) \in F_i$ is the frequency assigned to transceiver t_i .

2.3 Multiobjective Formulation of FAP

The Frequency Planning in GSM comprehends the optimization of several objectives. Initially, in our work we have approached FAP as a single-objective optimization problem. Then, an enhancement of the initial approach was performed. In this new approach we formulate FAP to address two distinct objectives and it tries to minimize both values when assigning the frequencies to the TRXs.

Detailing the adopted approaches, in the single-objective optimization version of FAP it is attempted to minimize the level of interferences propagated to the network. This formulation only uses the interference cost value. Then, the second approach, which is the most complex, considers the following distinct goals: the interference cost and the separation cost value.

Next, the two cost values, identified above for our formulation of FAP, are described in more detail.

2.3.1 The Interference Cost

The interferences occur when frequencies are reused by several TRXs. When it happens, they provoke some degree of degradation in the service quality of the GSM network.

Considering that $T = \{t_1, t_2, \dots, t_n\}$ is a set of n TRXs, and let $F_i = \{f_{i1}, \dots, f_{ik}\} \in N$ be the set of valid frequencies that can be assigned to a transceiver $t_i \in T, i = 1, \dots, n$ (the cardinality of F_i could be different for each TRX). Furthermore, let $S = \{S_1, S_2, \dots, S_m\}$ be a set of given sectors (or cells) of cardinality m . Each transceiver $t_i \in T$ is installed in exactly one of the m sectors and is denoted as $s(t_i) \in S$.

To quantify the interference cost an *interference matrix* denoted by M is used [84], where each element $M(i, j)$ represents the degradation of the network quality due to co-channel interferences, or adjacent-channel interferences sector [93]. Co-channel interference occurs when sectors i and j operate with the same frequency, while adjacent-channel interferences happen when two TRXs operate on adjacent channels (i.e., when one TRX operates on frequency f and the other on frequency $f + 1$ or $f - 1$).

Two values μ_{ij} and σ_{ij} of a matrix entry $M(i, j) = (\mu_{ij}, \sigma_{ij})$ representing numeric values of the mean and standard deviation respectively, of a Gaussian probability distribution, are used to quantify the interferences on the GSM network.

A better communication quality is accomplished when having a lower interference level, only possible when the mean value of $M(i, j)$, that is μ_{ij} is an higher value.

A solution to the problem p lies in assigning to all the TRXs of a network a valid frequency (t_i) from its domain (F_i). Thus, interference cost is accordingly defined as [92]:

$$C_I(p) = \sum_{t \in T} \sum_{u \in T, u \neq t} C_{sig}(p, t, u) \quad (2.2)$$

where, given a set T of TRXs, C_{sig} computes the *co-channel* interferences (C_{co}) and the *adjacent-channel* interferences (C_{adj}) for all sectors s_t and s_u , in which the transceivers t and u are installed. Our real-world approach of FAP considers that each TRX has a set of valid frequencies that may be different depending on the TRX.

The computation of the $C_{sig}(p, t, u)$ is performed considering the following:

$$\begin{cases} K & \text{if } s_t = s_u, |p(t) - p(u)| < 2 \\ C_{co}(\mu_{sts_u}, \sigma_{sts_u}) & \text{if } s_t \neq s_u, \mu_{sts_u} > 0, |p(t) - p(u)| = 0 \\ C_{adj}(\mu_{sts_u}, \sigma_{sts_u}) & \text{if } s_t \neq s_u, \mu_{sts_u} > 0, |p(t) - p(u)| = 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

where K is a very large value, defined in the configuration files of the network. The K value makes it undesirable to allocate the same or adjacent frequencies to TRXs that are installed in the same sector.

In [92] a more detailed explanation about the mathematical formulation used to compute these costs is available.

2.3.2 The Separation Cost

Technical limitations in the construction of sectors and BTSs mean that certain combinations of TRX channel are not permitted. Consequently the constraints that rise include the following:

Site Channel Separation: Any pair of frequencies at a site (BTS) must be separated by a certain fixed amount, typically 2 channels for a large problem. If a BTS uses high power TRXs then its channel separation should be larger. The violation of this separation involves a cost (C_{site}). In our case, in order to calculate this cost we count the number of site channel separations that are violated by a proposed solution to the FAP.

Sector Channel Separation: This is similar to the previous one, but at sector level. In conclusion, any pair of frequencies at a sector must be separated by a certain fixed amount, typically 3 channels for a large problem. It is important to observe that the sector channel separation generally is larger than the site channel separation, due to the greater nearness of the involved TRXs. As in the previous case, the violation of the sector channel separation involves a cost (C_{sector}). In our case, in order to compute this cost we count the number of sector channel separations that are violated by a proposed solution.

As a result of all the previously presented constraints, a solution p lies in assigning to all the TRXs of a network a valid frequency in order to minimize the following separation cost function:

$$C_S(p) = C_{site}(p) + C_{sector}(p) \quad (2.4)$$

2.4 Summary

In this chapter, we have introduced FAP and its main concepts. We have first described the problem in a theoretical perspective, presenting its mathematical model that is commonly used by benchmark problems.

We have then moved to a more realistic and accurate representation of the problem, using the context of real GSM networks. Indeed, this represents the scenario where our study was conducted. The addressed GSM frequency planning includes the main concepts related to the GSM network structure and their implications in the FAP formulation.

The chapter finalizes with the presentation of an alternative formulation that we propose and that can be applied in the optimization of distinct objectives in FAP. It consists in a new version for a multiobjective optimization formulation of FAP in GSM networks.

FAP Literature Review

In this chapter, we outline the current state-of-the-art related to the main research subject of our work. Initially, we present the more relevant accomplishments done with benchmark-like problems in FAP. In this research line, several metaheuristics are highlighted to solve traditional FAP formulation.

Following, after the test scenarios adopted by initial FAP approaches, we then present a more realistic approach of FAP, which is extended to the GSM network context. Further details about the enhancements made recently with a more realistic formulation of FAP are also presented.

Finally, we address the recent developments in FAP but according to a new perspective, which is a multiobjective optimization problem.

3.1 Frequency Planning: Research Approaches

In recent years, due to the enormous growth of mobile users and since the communication infrastructure continues to have a limitative bandwidth for providing a good service to them, the reuse of the available radio spectrum still is a critical research subject. For that reason, several attempts to solve the several versions of FAP have been investigated over the last decades.

Although FAP is formulated under various classifications, the main aim is that frequencies can be assigned to multiple base stations if they respect the reuse distance. Satisfying the electromagnetic compatibility constraints is one of the common considerations of the various approaches that have emerged over the years. The types of interferences, normally taken by FAP formulation, are the

co-channel interferences, co-site interferences and the adjacent-channel interferences*.

Tracing the path followed by FAP research line over the years, the earlier works were developed using a standard classifications of FAP, as the one described in Section 2.1.2. Indeed, an extensive summary of the work done before 1980 can be found in [57].

The starting point of FAP research has defined its classic formulation in a cellular networks context. The key characteristics of this formulation are the limitations imposed by the need of reuse the limitative radio spectrum in the cells in order to avoid the propagation of radio interference in the network. Those interferences are the ones that compromise mostly the quality of the service provided for the mobile users. Different types of telecommunication networks have emerged along the last decades, but the same restrictions and considerations still remain necessary.

Under the available classifications of FAP, the earlier researches intended to solve FAP as a minimum span problem (MS-FAP), for which various heuristics have been proposed for its resolution [80, 129]. Further details about other approaches are discussed in the next subsection.

3.1.1 Test Problems Over FAP

Frequently, when addressing FAP, a common approach that considers the network restrictions is followed. Those restrictions are associated with the several formulations proposed to the known classifications of the problem. In order to compare the performance of the analyzed algorithms which were investigated over the years to address frequency planning, a common approach was the use of the same benchmark instances by most of the researchers. The choice was focused on adopting well known-benchmark instances of the problem for the experiments scenarios. Initially the Philadelphia benchmarks [39] were the most widely used in the literature. The Philadelphia instances use the formulation presented in Section 2.1.3, representing a 21-node cellular network that represents eight different benchmark instances. In fact, a broad range of investigators have used Philadelphia test problems to conduct their research investigation [3, 11, 12, 22, 28, 44, 49, 50, 77, 80, 126–129, 134, 139, 146].

Further, among the available benchmark instances and besides the Philadelphia test problems, more realistic versions were available. In the benchmark context, instances as the COST 259, CALMA project instances (namely the CELAR and GRAPH instances) represent important reference points in this research area.

*These three types of interferences considered by FAP are presented in more detail in Section 2.3 where we propose and describe a multiobjective formulation for a real world FAP formulation on GSM networks.

In [39] relevant information describing those benchmark test problems as well as other types of instances that represent realistic networks are given in more detail.

In brief, the COST 259 benchmark is one of instances that tries to represent realistic concepts, but still is far from incorporating the same real concepts that we are using in our FAP formulation and test scenarios (see Section 2.2 and Section 6.1). Besides the COST 259 benchmark used by [13, 16, 31, 63, 96], indeed the first well-known benchmarks were the Philadelphia, CELAR and GRAPH instances [2, 19, 81, 143], but they represent even a more minimalist approach. In fact, these instances were randomly generated, although they have tried to incorporate realistic information.

Besides the addressed benchmark problems, in the performance evaluation of the investigated algorithms proposed over the last decades, other types of instances were also used, some of them tried to simulate a realistic scenario of FAP, portraying real networks. For example, problem instances as the one representing the Helsinki, Finland data that was taken from [83]. It represents a realistic demand distribution on a 9000x9000 m^2 area in the city of Zürich.

In the course of the following working scenarios it began to be more usual the use of real network data representing real cities over the world. That was the case scenario of our investigation in which we have used network information from the Denver and Seattle instance that represent real networks currently operating. The information defining these two instances was made available by one of the group members [91] of the OPLINK research project.[†]. The mathematical model followed by the Denver and the Seattle instances deal with the issues that are relevant in a GSM frequency planning scenario.

Despite the traditional benchmark instances, some researchers started to adopt real scenarios to test the performance of proposed algorithms to solve FAP [83, 85, 91, 93]. Indeed, the performance of distinct types of metaheuristics on the frequency assignment problem is the topic of several ongoing researches. Specifying, in our case the main work is based on GSM networks. Therefore, a more recent formulation of FAP using GSM networks has been recently proposed [91–93].

Throughout last decade and because of the tremendous success of mobile phone systems, FAP has been investigated under specific types of mobile networks as the Wireless networks, GSM networks, etc. Recent contributions were given by [6, 14, 26, 28, 46, 80, 91–93, 95, 126] as well as our own outcomes using the GSM networks [27, 103–107], etc. Nowadays, FAP still is a major research issue and therefore one of the goals is based on tackling FAP in GSM networks. For this

[†]The OPLINK project is composed by four Spanish research groups. Each group is responsible of developing different optimization methods. Information about the research group can be found in <http://oplink.lcc.uma.es>.

type of mobile networks the current available problem instances use large-scale real-world information. They are used to assess the performance of the developed algorithms.

3.2 Metaheuristics for FAP: A Summary

The development of the digital cellular phone standard GSM in the late 1980 and 1990 originated an increasing interest for the frequency assignment problem [39, 128]. Indeed, in the last decades, the number of mobile users has continued to increase but the communication bandwidth still remains scarce. For that reason, the problem of efficiently use the radio spectrum for cellular mobile communication still remains currently to be a critical research issue [95].

In short, in the last years many researchers have applied different techniques to solve FAP. One of the first approaches resorts to the use of graph methods, heuristics as well as other types of optimization procedures that are used combining several different techniques. The use of a wide variety of methods is due to the fact that an algorithm able to solve this problem in a polynomial time does not exist and therefore, it belongs to the NP complete class problems.

Initially, Hale [57] presented a wide group of different versions of FAP formulations. In his work, he has shown that the channel assignment problem is equivalent to the graph-coloring problem, when only considering the co-channel constraints. Then, a more complex approach was presented by Gamst and Rave that have derived FAP using a graph theoretic view point [47].

The more relevant accomplishments derived with FAP investigated techniques accomplished by embracing the graph theory approach and their extensions were conducted by [5, 45, 65, 130, 131, 139]. A common path of all of them was the use of the Philadelphia instances that represent the most well-known used instances in FAP literature. Since FAP can also be reduced to a special case of the vertex coloring problem, several approaches have been investigated under that perspective. A weighted coloring version of the FAP using a specific optimization algorithm [149] was also proposed.

According to the above presented perspective of FAP, earlier researchers attempted to solve frequency planning from a graph theoretic point of view, for which they started to propose distinct heuristics [80, 129]. A common indicator of all those studies is the use of a graph to model the cellular network that ignores the geometry of the network. However, the cellular network has a very regular structure. Therefore, other algorithms developed based on graph concepts under hexagonal cellular networks continued to be studied [124, 125]. The following researches have then considered the regular structure of the network and then a further graph modeling of a channel assignment problem was conducted by

[76, 87, 112, 113, 123]. The use of Graph Theoretical concepts in the cellular networks was also followed by [125] and then in GSM networks by [46].

As has been said, commonly FAP started to be formulated as a generalization of the graph coloring problem for which several metaheuristics were proposed, including classic methods and heuristic methods.

Undeniably, a wide range of approaches have emerged over the years to solve the already existent FAP formulations. The metaheuristics developed under the classic methods were the Graph Coloring algorithms [45, 47] and Integer Programming [1, 7, 82, 99]. Later improved approximate algorithms, using Heuristic methods [17, 45, 47, 57, 129, 141] were also proposed to address FAP.

Under that class of algorithms the following techniques were investigated.

- Neural Networks [25, 43, 44, 62, 83, 132, 134].
- Genetic Algorithms (GA) or Evolutionary Computation based approaches [12, 13, 21–24, 28–30, 34, 70, 77, 85, 121, 133, 144].
- Local Search approaches such as Simulated Annealing [35, 36, 65, 98], Tabu Search [61, 65] and Constraint Programming [122].

Approaches using GA have been highly considered. Further, the use of a Local Search algorithm [146] was also analyzed for FAP. Then, some hybrid approaches combining several techniques were also investigated, such as the use of Simulated Annealing or Neural Networks [12]; a genetic algorithm coupled with fuzzy logic [68]; a Pattern approach developed to be incorporated in a two-phase heuristic algorithm [78], etc.

Heuristic algorithms [22, 142] designed to address FAP formulation, as well as other techniques, e.g. the use of saturation degree heuristic [11], which was first proposed as a technique for solving the graph coloring problem, were also under the investigated algorithms. Alternative approaches using FAP generic model were also used by [49] to propose an algorithm that attempts to provide an optimal assignment for particular benchmark problems.

In short, the key factor when reusing the radio spectrum by the network cells is the channel interferences. Neglecting other technical influencing factors, the channel interference resides mainly in a frequency value and in a distance value. A channel can simultaneously be used by multiple base stations, if their mutual separation is more than the reuse distance, i.e., the minimum distance at which two signals of the same frequency do not interfere. In this approach, representing a single-objective optimization problem (the same that we have initially followed in our line of work) the initial works were developed using Evolutionary Computing (EC) [10] techniques. Its purpose addresses a parallel genetic algorithm

developed by Cromptson et al. [29]. Then, under the CALMA project[‡] some other techniques using GA algorithms were also investigated [69].

Meanwhile, the integration of FAP under the omnibus area of NP-hard telecommunication problems has started to be investigated [4]. Soon, solutions combining FAP with other telecommunication problems have issued. For example the inclusion of FAP with the Radio Network Design (RND) problem, defining a new optimization front that tries to simultaneously optimize both problems aggregating them in a whole optimization goal. Again, some new approaches combining the Base station positioning problem and FAP started to be investigated by [67, 147] but under other perspectives and different instances.

Considering a retrospective covering the line of work followed over the last decades, clearly the most widely used FAP formulation was proposed by Gamst and Rave [47], which was then thoroughly studied in many distinct works, like for example the ones developed by [17, 36, 44, 45, 78, 83, 85, 98, 128, 129, 146, 148].

In our initial contribution, we have also started to address FAP under its classic formulation, for which we have applied Differential Evolution algorithm [135, 136] and a hybrid version that was tested over the already presented benchmark Philadelphia instances [100–102].

3.2.1 Related Work in GSM Networks

As it has been said, until the early 1980s most contributions used heuristics based on the Graph Coloring problem. This is the most simplistic approach to the problem. Afterward, more resolutions appeared and some of them showing interesting applications and results. One of the emerging scenarios is the formulation of the problem for GSM networks.

Hence, in the last years, several researchers have contributed with different overviews for the frequency assignment problem [82], but not focusing specifically the GSM system. Propelled by the increased development and acceptance of the digital cellular phone standard GSM, which has originated an increasing interest for the frequency assignment problem, it has allowed a contribution of works such as the ones developed by [39, 128]. And so, more diverse approaches for this type of networks have been published in literature.

The NP-hard nature of FAP boosted the research attempts to develop more efficient heuristics and algorithms. However, the achievement of an optimal solution cannot be guaranteed. Indeed, over the recent years, a much more realistic formulation under GSM networks has emerged. Therefore, from the many methods proposed to solve the FAP under GSM networks, we highlight the work

[‡]The CALMA project represents the Combinatorial ALgorithms for Military Applications.

developed by Alba and Chicano with [4], where they have reinforced the importance of the Telecommunication area as an important field of research, addressing many related research lines. The main focus of that work was indeed the use of Evolutionary Algorithms (EA)[§] approaches covering the wide telecommunication field.

Among the later accomplishments, most contributions using GSM networks were based on the work developed by the OPLINK project [4, 26, 27, 91–94]. The frequency planning base formulation developed by Luna et. al [92] for GSM networks tries to mathematically define an accurate interference information coming from real GSM networks. Indeed, it was adopted by us for the experiments scenario, defining the test problems that were used to evaluate our proposed optimization strategies.

Concluding, as we can see by the many research approaches that have been applied in real telecommunication networks, the success of the GSM networks has led to the adoption of different models and solutions. Those approaches have influenced the implementation of hybrid versions using some of the already mentioned techniques, as for example the use of several Evolutionary Algorithms (EA) techniques with modified version of some already known algorithms. Combining problem knowledge information within different strategies for the recent evaluated algorithms has begun to represent some of the recent contributions made to the progress of this domain scope.

3.3 Multiobjective Optimization: Applied Research

In the last decade, the design of metaheuristics for multiobjective optimization problems is being addressed and applied in many different research areas [140]. It has led to an increasing interest in applying evolutionary algorithms to multi-objective optimization problems. Solving real-life engineering problems requires often a multiobjective approach. Thus, multiobjective metaheuristics have become increasingly valuable, since real world optimization problems often involve several conflicting objectives for which a tradeoff must be found. Fortunately, the formulation of real-world problems in a multiobjective perspective has received a growing interest. Therefore, detailed studies have been reported in the literature for multiobjective approaches [8, 9, 48, 88, 114, 118, 150].

To find a set of representative Pareto optimal solutions in a single run, Multiobjective Evolutionary Algorithms (MOEAs), a class of stochastic optimization techniques that simulate biological evolution to solve multiobjective problems, have been highly proposed.

[§]Four main types of EAs were mainly considered: genetic algorithms, evolutionary strategy, evolutionary programming and genetic algorithms. The Genetic Algorithms are the most well-known EA.

Although some work related to FAP applied research has been found in classic cellular networks and in wireless networks, for the most part it has been studied extensively for the past three decades using mainly benchmark-like instances. Consequently, more recently, instances representing real cities and currently operating have started to be considered in more recent studies. As shown in the previous section, most of the existent studies have applied distinct classes of algorithms to address FAP as a single-objective optimization problem. Normally they have disregarded the possibility of formulating it such as a multiobjective optimization problem.

In frequency planning, recent works developed on GSM networks have applied distinct optimization strategies but using a FAP formulation that considers a unique optimization goal. The common approach intents to accomplish a frequency assignment containing the smallest level of interferences possible. Indeed, a more realistic and at the same time harsh scenario can be formulated using accurate information coming from real GSM networks context. Therefore, currently as well as in the future, the formulation of FAP under a multiobjective optimization problem in the real context of large-scale GSM networks still is a very important research subject in the network planning.

In FAP formulation for a GSM system, the algorithms proposed so far have only started to be analyzed over the recent years. Indeed, the multiobjective formulation proposed by [92] for GSM networks has not been yet applied on multiobjective algorithms. Hence, it still remains an open research area that it is starting to be investigated.

3.4 Summary

In this chapter we have presented some of the basic concepts and algorithms that represent distinct attempts to address FAP. First, based on its classic formulation and then in the specific scenario of GSM system. The relevance of the first scenario is due to the fact that their results still remain to be considered over the last years in the attempts that have then come up with the emergence of new technologies in the telecommunication area.

As a conclusion, current literature review shows that most of the research done in the last decades was focused on cellular networks. Most attempts made to solve FAP with the investigated algorithms have centered the results on that network technology. Indeed, a common setback is that every work analyzed normally tackles a benchmark-like FAP formulation, which is then adapted to other network technology. Therefore, in our specific line of work using GSM networks, it still exists a large spectrum of research available to develop new optimization approaches on optimizing a given frequency planning.

In the end, when bringing to focus a possible multiobjective formulation for FAP in GSM networks, we show some relevant arguments that satisfy an open area that permits the analysis of FAP at a real GSM optimization level.

In the following chapter we discuss in more detail some challenges that we have addressed when developing the algorithmic adaptation proposed to track FAP. Since it is a real-life engineering problem, solving it is far from being a simple task. Therefore we have investigated several types of metaheuristics and some comparisons were presented.

Single-objective Metaheuristics on FAP

In this chapter, we present several single-objective metaheuristics that were chosen by us to deal with the FAP formulation presented in Chapter 2.

The selected algorithms are a population-based metaheuristics named Differential Evolution (DE), and two trajectory-based metaheuristics, Variable Neighborhood Search (VNS) and its variant Skewed Variable Neighborhood Search (SVNS). All these algorithms have used a Local Search (LS) method that was specially designed to this problem. Both population-based and trajectory-based metaheuristics are discussed here together with hybridizations that we have designed to be incorporated on them.

We start by presenting some concepts that were taken from the FAP formulation in the context of a GSM network. These concepts were included in the metaheuristics that we present here. The information about the problem domain was incorporated in the metaheuristics to improve their performance on FAP. Having in mind that our aim is the optimization of FAP, we also present a Local Search method that was incorporated in the algorithms to improve their performance. Although the local search method was not developed by us it was considered in our research because it was used by other researchers using the same instances that us.

4.1 Metaheuristics Base Approach

The use of metaheuristics on combinatorial optimization problems is a rapidly growing field of research. Indeed, the importance increase of combinatorial optimization problems for the scientific as well as the industrial world has emerged. Likewise, that background scenario also happens in our research subject, which is FAP in GSM networks.

Many optimization problems of practical as well as theoretical importance consist in the search for a best configuration of a set of variables to achieve some goals. In our case, the best configuration is not known; therefore our goal is to look for a solution that tries to satisfy as many possible domain restrictions. Metaheuristics are an evident choice to address such a scenario.

A *metaheuristic* was formally defined by Osman and Laporte [115] as "an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search space, learning strategies are used to structure information in order to find efficiently near-optimal solutions". Therefore, metaheuristics are typically high-level strategies which guide an underlying, more problem specific heuristic, to increase their performance.

There are different ways to classify and describe metaheuristics. Among the possible classifications, the most popular distinguish between the trajectory-based heuristics and the population-based heuristics [15]. The main difference resides in the number of solutions used at the same time. In the first one, the algorithm works on a single solution at any time, and in the second one it works on a population of solutions. The trajectory methods include local search-based metaheuristics, such as the Tabu Search (TS) [51, 52], Variable Neighborhood Search (VNS) [58, 109], Simulated Annealing (SA) [20, 79], Iterated Local Search (ILS) [89, 90, 97, 137, 138], etc. They share a common feature, which consists in following a trajectory during search process. On the other hand, in the population-based metaheuristics the search process describes the evolution of a set (population) of solutions in the search space, rather than a single solution like in the trajectory-based metaheuristics. Included in this class of algorithms are all the Evolutionary Computing (EC) based metaheuristics, and the Ant Colony Optimization (ACO). Indeed, the Genetic Algorithm (GA) [64] is one of the most well known evolution algorithms, but both Differential Evolution (DE) and Scatter Search (SS) [53, 54] are widely known evolutionary based methods*. Their

*Differential Evolution algorithm corresponds to one of the metaheuristics that we have investigated to address FAP. The Scatter Search [26, 27] algorithm corresponds to other of the strategies that were investigated for the same formulation of FAP that we are following in our research, but by other research group member. Likewise, also the ACO and GA have already been investigated [92].

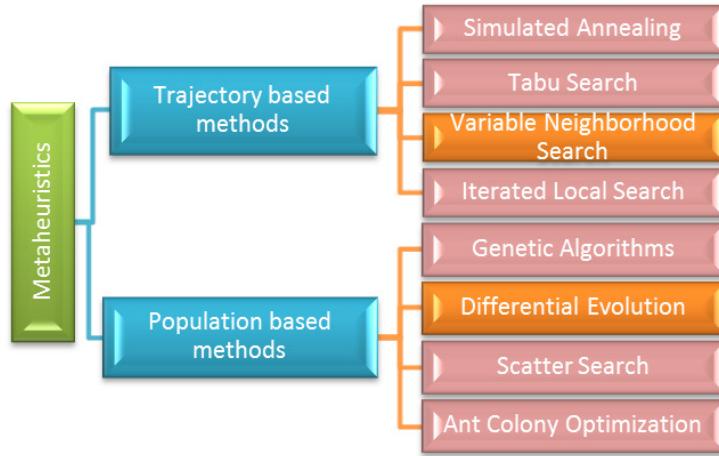


Figure 4.1: Metaheuristics classification, showing some algorithms examples for every classification type.

similarity resides in the fact that at each iteration a number of operators are applied to the individuals of the current population to generate the individuals of the population of the next generation (iteration).

Figure 4.1 shows some of the algorithms included in the two classes of metaheuristics, which were both addressed in our work. In our particular context, we have used the two main classes of metaheuristics that are described above, which correspond to the trajectory-based and the population-based metaheuristics.

Additionally, in both metaheuristics types that we have developed and implemented to track FAP they make use of domain-specific knowledge to guide the search and to efficiently explore the search space in the finding of the best solutions. Therefore, our proposed metaheuristics correspond to hybrid metaheuristics that were developed over the classic characteristics of the algorithms for which we have then applied some hybridizing methods, such as a local search improvement. In fact, with this method, metaheuristics have performed considerably better than with their classic versions.

A more detailed overview that outlines the basic metaheuristics that do not make part of this research aim is available in [15] and it proposes a conceptual comparison of different metaheuristics.

Even though metaheuristics are different in the sense that some of them are population-based like DE, and others are trajectory methods like the VNS and SVNS, our goal is to identify which are the metaheuristics that perform better when addressing the real-world instances of FAP formulation that are under our research area.

4.2 Problem Domain Information Incorporated in the Algorithms

In a technical level, the optimization of FAP aims to obtain a frequency plan with a minimum conflict free assignment of interferences regarding the frequencies assigned to the TRXs in the mobile network. In our initial approach of FAP, using its single-objective formulation, all the investigated metaheuristics use the interference cost value as evaluation criterion to assess which are the frequency plans with the smallest interference cost, representing therefore the best solution.

Since our research is based on a real-world scenario, we have used large-scale real-world instances of GSM networks. For us to be able to address FAP and optimize the performance of the metaheuristics being compared, we have incorporated some characteristics of the mobile networks. It consists in the inclusion of some level of knowledge about the problem domain in some of the operations that were adapted by us, originating the design of new hybrid metaheuristics.

Next we make a more detailed description regarding the features that we have incorporated in the metaheuristics.

4.2.1 Problem Domain Knowledge

To generate more efficient solutions, normally the incorporation of some knowledge about the problem domain is required. Using additional background information we aim to obtain more profitable solutions in a smaller number of iterations. In our scenario it also permits us to guarantee that all the generated solutions are valid solutions, and therefore they represent real frequency plans that can be used in the mobile network structure.

In FAP, when creating a frequency plan, the main task consists in assigning a frequency value (from the available frequency spectrum) to every transceiver in the network. The simplest way to perform this operation is by assigning a random frequency value to every TRX, from its range of valid frequencies. The information regarding the valid frequencies of a TRX is obtained from a network configuration file. Another important fact that needs to be taken into account to be fulfilled during the assignment task is that TRXs have different ranges of valid frequencies. Therefore, the frequencies that can be assigned to a TRX can be different to the ones to be assigned to other TRX. As a result, in our work, every time a new frequency plan is being generated, the algorithm performs an additional operation that assures that a frequency is assigned to a TRX installed in a sector only if this frequency does not produce adjacent-channel interferences or co-channel interferences within this same sector. This restriction tries to minimize the highest-cost kind of interferences (representing the penalty cost K) that can occur in the frequency planning (see Chapter 2.3.1). It is indeed

a very frequent operation that is performed several times in all the iterations of an algorithm. It basically consists in a validation that is performed every time a new frequency value is assigned to a TRX. Consequently, it represents one of the more common operations performed by all the metaheuristics addressed within the scope of our research. [†]

Considering all the restrictions imposed by FAP, we realize that it will be useful to keep, during algorithm execution, a set containing all the available frequencies for every sector of the mobile network. Thus, this set is continuously being updated.

In the beginning, since every sector has several TRX, the same set of available frequencies is shared by all the TRX inside that sector. First, when starting, the range of valid frequencies is completely available to all the sectors. Then, every time a new frequency f is assigned to a TRX, that new value f as well as its respective adjacent frequencies $f + 1$ and $f - 1$, are removed from the set of available frequencies to the sector in which the TRX is installed. Therefore, they become unavailable. Meanwhile, the older frequency value reappears in the set, becoming from now on available.

The outline is that incorporating in the metaheuristics the strategy presented here, it contributes to the improvement of solutions because it permits a significant decreasing in the final cost of the frequency plan. This happens because when we incorporate information about the problem domain, we are suppressing one of the most costly penalties that cause some of the most severe interferences that may severely reduce the quality of service of the mobile network, down to unsatisfactory levels.

4.2.2 FAP Local Search Heuristic

The use of a FAP formulation that uses real-world interference data, which is available in two large-scale instances that are being used by us and also other researches in this field of expertise, has lead us to share some optimization strategies. The results accomplished by the work developed by the other authors are considered by us in order to compare the performance between our metaheuristics and the ones they are investigating.

In this context, a common Local Search (LS) heuristic [93] was used to enable the improvement of the accomplished FAP solutions. Indeed, using a Local Search heuristic normally permits the accomplishment of more profitable solutions. For that reason, we have incorporated the LS in the metaheuristics that we are analyzing.

[†]In the multiobjective metaheuristics proposed by us to address FAP (see Chapter 5), we also have incorporated these same concepts.

The LS heuristic developed to undertake the real-world version of FAP that we address uses information about the domain of the problem. Specifically, considering a candidate solution (p), it tries to optimize the assignment of frequencies to the TRXs in a given sector by analyzing all the neighborhood solutions of those solutions. The assignment of frequencies is performed without changing the remaining network assignments. The LS method is applied to the candidate solution until none of the neighbors can improve it.

Considering that p is coded as a vector of integers, in which every index $p(t)$ represents a TRX t and the value assigned to that position represents a frequency value. The LS starts by identifying the neighborhood solutions. The neighborhood solutions of p are established by replacing the frequencies in the TRXs of each sector. The reassigning of frequencies for all the TRXs inside a sector is performed considering the cost of the assignment. The neighborhood solution that will be selected will be the one with the smallest cost.

Indeed, when the same range of valid frequencies is available to all TRXs in a given sector, the neighborhood solutions that are generated by the LS method permits to obtain the optimal frequency planning inside that sector. The LS method also considers the restriction presented in the previous subsection, that is, it will never assign the frequency or adjacent frequencies to TRXs that are inside the same sector.

The neighborhood solutions are analyzed at random. An auxiliary structure is used to identify which sectors might improve the current solution. A more detailed explanation about the local search algorithm developed for FAP is available in [93].

In conclusion, the evidence that integrating a Local Search method [34, 41, 75, 145, 146] in the design of the metaheuristics can indeed make some improvements in the accomplished solution, lead us to propose the hybridization of the metaheuristics to make them more efficient to FAP. Take for example the case of the Differential Evolution algorithm [135, 136], in which we have made some enhancements to its classic version proposing a new hybrid version of DE that incorporates a Local Search method in the reproduction phase.

4.3 Differential Evolution Algorithm

Differential Evolution (DE) is an evolutionary algorithm (EA) proposed by Ken Price and Rainer Storn [116, 117, 135, 136]. It is a population-based and stochastic search algorithm of evolutionary computation that combines simple arithmetic operators with the classical operators of crossover, mutation and selection to evolve from a randomly generated starting population to a better population. These three steps are performed until a stop criterion is not reached.

The main difference between DE and other EAs is the implementation of the mutation operator. DE starts with an initial population to which a scheme (that is the main feature behind DE) is applied that permits to generate a new individual by computing new parameter vectors by adding the weighted difference vector between two individuals in the population to a third individual. This candidate solution is a better individual if it has a lower fitness value (interference cost). In that case, it will prevail to the next generation.

4.3.1 Intrinsic Control Parameters

The performance of DE is intrinsically related to the parameter settings required for its control parameters, which are:

- Population size (NP).
- Crossover probability (CR).
- Mutation intensity (F).
- Differential Evolution strategy.

The population size is set at the beginning of the algorithm and it is formed by NP individuals, each one with n components. In our case, with FAP the n value represents the number of TRXs in the mobile network instance. In DE the population size does not change during algorithm execution. The individuals generated to form the initial population (parent population) are created randomly. This process is not entirely random, because it incorporates some knowledge of the problem domain to grant valid individuals.

Afterwards, two main tasks are performed: the generation phase and the selection mechanism. In every iteration of DE, both operations will generate a child population, representing the new solutions to the problem. In the generation phase the mutation and the crossover parameters (F and CR parameters, respectively) are used. In DE the mutation is applied once at a time to every member of the population, forming a child population that will also contain NP trial vectors. To the mutated vectors is then applied a crossover operator. More specifically, the DE basic strategy can be described as follows:

Mutation

The mutation process combines in a new candidate solution, defined by a vector named x_{trial} , a scaled difference vector to a third vector. The F parameter is therefore used as a scale to the mutation step length. Differential mutation requires, at least, the selection of three random distinct vectors, $x_{r0}, x_{r1}, x_{r2} \in$

$1, 2, \dots, NP$ that are selected from the parent population. The randomly chosen integers $r1, r2$ and $r3$ are also chosen to be different from the running index i , so that NP must be greater or equal to four to allow this condition to happen.

The first selected vector is used as based value (x_{r0}) for the mutated vector, to which is added the scale factor F that is multiplied by a difference vector obtained by the other two random selected vectors from the parent population x_{r1}, x_{r2} .

Equation 4.1 defines this process that represents the $DE/Rand/1/\gamma$ schema.

A new mutation vector is created for every member in the parent population. Therefore, one at a time every vector of the population (x_{target}) is used to generate the x_{trial} vector. Thus, the x_{target} vector corresponds to the solution in the index i of the population. DE uses a uniform crossover to complement the differential mutation strategy.

$$x_{trial}^t = x_{r0}^t + F(x_{r1}^t - x_{r2}^t) \quad (4.1)$$

Crossover

In the crossover operation, the mutated vector is recombined with another predetermined vector from the parent population (x_{target}), to produce the trial vector (x_{trial}). Indeed, the crossover builds the trial vector out of the parameter values that are copied from two different vectors.

$$x_{trial,j}^{t+1} = \begin{cases} x_{trial,j}^t & \text{if } rand_j(0, 1) \leq CR \\ x_{target,j}^t & \text{otherwise} \end{cases} \quad (4.2)$$

where, $i = 1, \dots, NP$, and $j = 1, \dots, n$.

This operation is controlled by the crossover probability, defined by the parameter $CR \in [0, 1]$ that controls the amount of index values that are copied for the trial vector from the mutated vector or from to the target vector. Equation 4.2 formulates the recombination process. If $rand_j(0, 1) \leq CR$, the index to the trial vector is taken from the mutated vector, otherwise it is inherited from the target vector (that was copied from the current population member - $x_{target,j}$). If $CR = 1$, the trial vector is exactly the same as the mutated vector.

Selection

After the creation of the candidate solution, named x_{trial} that uses both F and CR parameter, DE starts the selection phase.

Selection aims to decide whether or not the x_{trial} should become a member of the next generation ($t+1$). Each vector in the child population will compete against the vectors in the parent population using the objective function value. Therefore, vectors are compared using the greedy criterion, meaning that the vector (individual) with the best fitness value will prevail to the next generation. Thus, the x_{trial} vector will replace the x_{target} vector if it yields a better objective function value than the x_{target} vector. Equation 4.3 formulates the selection phase mathematically.

Selection is the final operation carried out to generate the vectors in the child population that will become the parent population for the next generation.

$$x_i^{t+1} = \begin{cases} x_{trial}^{t+1} & \text{if } f(x_{trial}^{t+1}) \leq f(x_i^t) \\ x_i^t & \text{otherwise} \end{cases} \quad (4.3)$$

where,

$$x_i^t = x_{target}^t$$

4.3.2 Differential Evolution Strategies

In the creation of the trial vectors, DE uses the mutation intensity factor within a scheme. Price and Storn [116, 135] have suggested several different schemes which can be classified using the notation $DE/\alpha/\beta/\gamma$. The α specifies the vector to be mutated and therefore it will be added to the weighted difference vector. The α value can assume the value *rand* or the value *best*. The first one indicates that vectors will be selected randomly from the population and with *best* it will be used the best vector from the current population. The β value represents the number of difference vector pairs used, that normally are 1 or 2. Finally, the γ denotes the crossover scheme type that can be binomial (*bin*) or exponential (*exp*).

Equation 4.1 shows the use of the above notation the $DE/Rand/1/\gamma$ and Table 4.1 summarizes some of the most popular schemes available in DE.

Algorithm 1 presents the structure followed by DE, in which it is possible to identify where the operations described above are performed in every iteration of the algorithm.

4.3.3 The Classic Algorithm Structure

The classic DE algorithm structure starts with the setting of the intrinsic parameters values that influence the three main steps which are executed in each generation of the algorithm. The generation, mutation and selection phase are performed until a stop criterion is not reached. The stop criterion can be e.g.

Table 4.1: Formulation of the different strategies used by DE algorithm in the creation of the mutation vector.

DE strategy	DE mutation definition
Rand/1/ γ	$x_{trial}^t = x_{r0}^t + F(x_{r1}^t - x_{r2}^t)$
Best/1/ γ	$x_{trial}^t = x_{best}^t + F(x_{r0}^t - x_{r1}^t)$
Rand/2/ γ	$x_{trial}^t = x_{r0}^t + F(x_{r1}^t + x_{r2}^t - x_{r3}^t - x_{r4}^t)$
Best/2/ γ	$x_{trial}^t = x_{best}^t + F(x_{r0}^t + x_{r1}^t - x_{r2}^t - x_{r3}^t)$
Rand-to-Best/1/ γ	$x_{trial}^t = x_{r0}^t + F(x_{best}^t - x_{r3}^t) + F(x_{r1}^t - x_{r2}^t)$

the maximum number of generations, a not reached pre-determined value, the execution time, etc.

The iterative process of DE is initialized with the creation of the initial population P that is formed by NP vectors. Every vector is an individual of the population.

In DE, at every generation (t), and for every population member (i) the following operations are performed (see Figure 4.2):

1. Choosing the target individual (it is the current population member);
2. Randomly choosing three population members r_0, r_1, r_2 , all different, that will participate in the creation of the new individual;
3. Creating the trial individual, using a DE scheme, the mutation phase and crossover operation;
4. Evaluating the trial individual;
5. Selecting for the next generation the individual with the best (smallest) fitness value between the target and the trial individual;
6. If the stopping criterion has not been reached, then it will continue to the next generation.

A more detailed explanation about the creation of the trial vector (x_{trial}) and selection of the target vector x_{target} is available in the previous subsection, where every intrinsic control parameter of DE was described.

The complete pseudocode for DE is given in Algorithm 1 that includes a detailed explanation on how to create the child vector x_{trial} in every iteration of the algorithm.

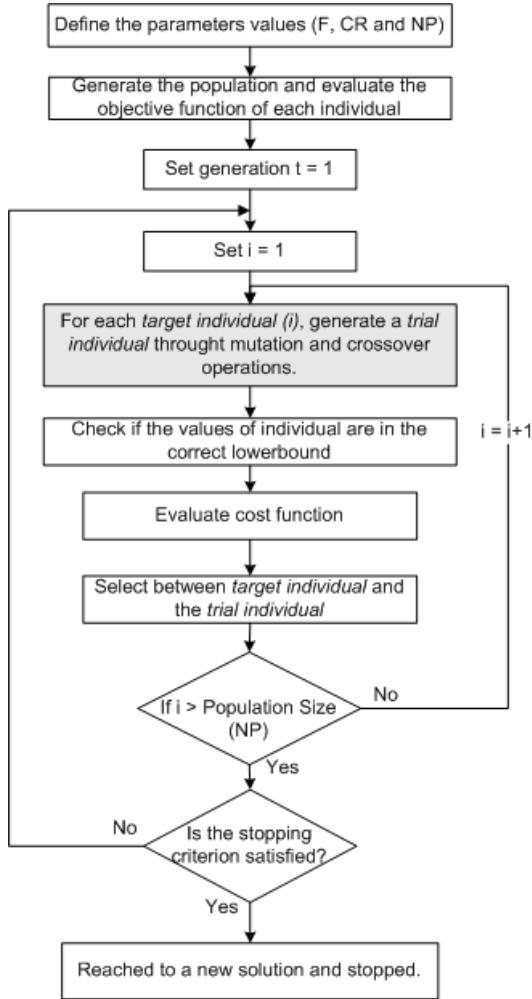


Figure 4.2: Classic DE algorithm.

4.3.4 Hybrid Differential Evolution Approach

To the classic DE algorithm we have proposed some additional improvements that use information about the FAP domain knowledge. The new approach that we have designed is a hybrid DE algorithm that tries to optimize the results obtained for FAP (it tries to optimize the frequency plan of a mobile network).

In our hybrid DE version we have added to its classic structure two additional features. The pseudocode for the proposed hybrid version of DE algorithm is presented in Algorithm 2.

The new steps include the incorporation of a Local Search method (line 7 of Algorithm 2) and avoid that in every generated solution the most severe penalties do not occur. Basically, the optimizations that we propose in DE can be described

Algorithm 1 Pseudocode for the Differential Evolution algorithm.

```

1:  $t \leftarrow 0$                                      ▷ initial generation
2: population  $P^t \leftarrow \text{CreatePopulation}(NP)$           ▷ NP=population size
3: Evaluate( $P^t$ )           ▷ evaluates fitness of every individual in the population
4: while stopping criterion is not reached do
5:   for  $i = 1$  to  $NP$  do
6:      $x_{target} \leftarrow x_i$ 
7:      $r_0 \leftarrow \text{selectRandomIndividual}(P^t)$ 
8:      $r_1 \leftarrow \text{selectRandomIndividual}(P^t)$ 
9:      $r_2 \leftarrow \text{selectRandomIndividual}(P^t)$ 
10:    ▷ Begin creation of trial individual
11:    for  $j = 1$  to  $D$  do           ▷ D=dimension of the problem
12:      if ( $rand[0, 1] < CR$ ) OR ( $j = j_{rand}$ ) then
13:         $x_{trial,j}^t \leftarrow x_{r0,j}^t + F * (x_{r1,j}^t - x_{r2,j}^t)^t$           ▷ DE scheme
14:      else
15:         $x_{trial,j}^t \leftarrow x_{target,j}^t$ 
16:      end if
17:    end for
18:    ▷ End of creation of trial individual
19:    Evaluate( $x_{trial}^t$ )
20:    if  $f(x_{trial}^t) < f(x_{target}^t)$  then          ▷  $x_{trial}$  is better than x  $x_{target}$ 
21:       $x_i^{t+1} \leftarrow x_{trial}^t$ 
22:    else [ $x_{trial}$  is worse than  $x_{target}$ ]
23:       $x_i^{t+1} \leftarrow x_{target}^t$ 
24:    end if
25:  end for
26: end while
27: return best( $x_i, P^t$ )          ▷ represents the best individual

```

as:

Method 1 The first modification consists in guaranteeing that for all the TRXs inside the same sector will not be assigned the same or an adjacent frequency value, because it will provoke severe penalties. These *co-channel* and *adjacent-channel* interferences are the highest-cost interferences, as shown in Equation 2.3.

Method 2 The second optimization consists in applying a Local Search (LS) method, which was adapted specifically to this FAP formulation [93]. As described above, this method permits to optimize the assignment of frequencies to every TRX in each sector of a given solution. A more detailed explanation about the specific LS features applied to the FAP problem can be consulted in the reference [93].

Algorithm 2 Pseudocode for Hybrid Differential Evolution algorithm.

```

1:  $t \leftarrow 0$                                       $\triangleright$  initial generation
2: population  $P_t \leftarrow \text{CreateEfficientPopulation(NP)}$        $\triangleright$  NP=population size
3: EvaluatePopulation( $P_t$ )
4: while not time-limit do
5:   for  $i = 1$  to  $NP$  do
6:      $x_{trial} \leftarrow \text{CreateTrialIndividual}(x_{target})$            $\triangleright x_{trial} \leftarrow i$ 
7:      $x_{trial} \leftarrow \text{ApplyLocalSearch}(x_{trial})$ 
8:     EvaluateIndividual( $x_{trial}$ )
9:     if  $f(x_{trial}) < f(x_i^t)$  then
10:        $x_i^{t+1} \leftarrow x_{trial}$ 
11:     else [ $x_{trial}$  is worse than  $x_i$ ]
12:        $x_i^{t+1} \leftarrow x_i$ 
13:     end if
14:   end for
15: end while

```

The first optimization is used when generating the initial population (line 2, Algorithm 2) and every time a new solution is created. Therefore, in DE it occurs every time some changes are made to the genes of a given solution. Thus, it happens every time *trial* individuals are created (line 6, Algorithm 2).

We have also tried to optimize the implementation process; therefore we have incorporated in every solution a dynamic list containing all the available frequencies for every sector. With the LS method we also have the guarantee that the same frequency value or an adjacent frequency is not assigned to two TRXs within the same sector. Thus, any time changes are made in a solution it is assured that the co-channel and the adjacent-channel interferences are not violated within the same sector. Incorporating that restriction in the structure of the algorithm requires that every time a change is made in the frequency assigned to a TRX, the list of available frequencies needs to be updated to reflect that alteration in the set of frequencies that from now on will be unavailable (each sector of the network has its own list of available frequencies). This strategy is described in more detail in Section 4.2.

The main advantage of using an extra list containing the range of available frequencies for every sector is that it is no longer necessary to verify, for each sector, what the frequencies are that are not being used. All that information is immediately available for a sector because it is constantly being updated.

The second optimization made in the classic DE to tackle FAP consists in applying an additional improving Local Search method to an individual. Having this goal in mind, an additional question rises because several alternative points can be thought-out. Indeed, the open question was: What is the best place in

the classic DE structure to incorporate the LS feature? Two hypotheses can be considered:

- The first one, identified as *LS1*, applies the local search heuristic to all the individuals belonging to the initial population and also in every step of the algorithm where a new x_{trial} individual is created.
- In the second hypothesis, named *LS2*, the local search heuristic is only applied after creating a x_{trial} individual. Therefore, it excludes the initial improvement made to all the solutions that form the initial population.

Both hypotheses were designed and adapted to be included in our hybrid DE algorithm. They were implemented and then tested to identify which the best approach was to be followed in our hybrid version. From the results obtained in the experiments scenario, we have identified that the one with the highest performance is the *LS2* approach. As a result, in our hybrid DE algorithm we propose the incorporation of the LS heuristic only after being applied the differential mutation operation to the vectors that are generated to an iteration of the algorithm.

In Section 6.4.1 the results that validate the choice that we made regarding the incorporation of LS method in the structure of DE algorithm are shown.

4.4 Variable Neighborhood Search Algorithm

Variable neighborhood search (VNS) is based on a local search developed by Hansen and Mladenovic [59, 60]. The VNS algorithm is a metaheuristic based on trajectories that have been used to solve optimization problems, and its main idea is based on systematic changes of neighborhood within a local search. The algorithm increases the size of neighborhood when the search does not move forward.

In this algorithm the mutation operator changes the environment when the obtained solution (after applying the Local Search) is worse than the current solution. The basic model for the VNS algorithm is presented in Algorithm 3.

An optimization problem can be formulated as a $\min f(x) | x \in X, X \subseteq S$, where S, X, x, f are the solution space, feasible set, feasible solution and a real valued function, respectively. A computational problem rises if S has a possible solution in a finite, but also large, set of possible solutions. The VNS metaheuristic intends to solve an optimization problem as the one described above. The FAP problem can also be represented in this context.

More formally [59], it is possible to denote $N_k (k = 1, \dots, k_{max})$ as a finite set of pre-selected neighborhood structures, and $N_k(S)$ as the set of solutions in

the k^{th} neighborhood of S . Several neighborhoods N_k can be produced from the solution space S . The VNS algorithm applies gradually a local search method on a solution S' obtained from a mutation of the current S , conducted according to the type of environment used in each iteration $N_k(S)$. If the last local search was effective, that is, if the solution obtained after it (S'') improved the current solution (S), in the next step of the algorithm the first environment E_1 would be used. Otherwise, it would be used in the next environment ($k \leftarrow k + 1$) the solution used at the beginning of the local search (S'). This choice would cause a biggest disturbance in the search space in which the current S lies.

4.4.1 Algorithm Structure

Besides the main functionalities described above, the VNS algorithm structure is simpler and it basically tries to obtain a new neighborhood solution in every iteration, by applying a mutation operator (line 4, Algorithm 3) which is then improved by a local search method (line 5, Algorithm 3). Different local search procedures can be applied in the VNS algorithm, and we used the method developed to this specific instance of the FAP problem [26, 27, 93, 104]. VNS applies the LS method with S as initial solution and denote with S'' the so obtained local optimum. If the candidate solution S'' is better than initial solution S (line 6, Algorithm 3), it will prevail to the next iteration as the initial solution. Otherwise, until the maximum neighborhood environment is not reached (k_{max} parameter), a new neighborhood candidate solution is analyzed.

Algorithm 3 Pseudocode for Variable Neighbourhood Search algorithm.

```

1:  $S \leftarrow \text{CreateInitialSolution}$                                       $\triangleright$  at random
2:  $k \leftarrow 1$ 
3: while  $k \leq k_{max}$  do
4:    $S' \leftarrow \text{MutationInEnvironment}_k(S)$ 
5:    $S'' \leftarrow \text{LocalSearch}(S')$ 
6:   if  $S''$  is better than  $S$  then
7:      $S \leftarrow S''; k \leftarrow 1$ 
8:   else
9:      $k \leftarrow k + 1$ 
10:  end if
11: end while
12: return  $S$ 

```

The VNS is a descendent, first improving method with randomization. It can be changed to a descendent-ascent method (also considering worse solutions than the best solutions found so far). As a result, several variants for the VNS algorithm have been proposed [58–60, 109]. They include several additional features that are incorporated into the structure of the basic version of VNS. Next we

present another variant using the variable neighborhood search that we have also implemented to solve FAP.

4.5 Skewed Variable Neighborhood Search Algorithm

The Skewed Variable Neighborhood Search (SVNS) algorithm is one of several extensions purposed to the VNS algorithm [58, 59, 109]. Both VNS and SVNS use a local search heuristic to find viable solutions within the search space of the problem. Likewise, the SVNS variation of the VNS algorithm also increases the size of neighborhood when the search does not move forward.

Furthermore, the SVNS also adopts the main concepts of VNS that combines efficient local optimization procedures with the heuristics which can handle the local optimum to escape from the local optima by restructuring the neighborhood. Different neighborhoods provide different candidate solutions, thus it is possible to find out a better solution.

As suggested above, one of the problems of using only an isolated local search algorithm to solve a problem is that it could be lost in a local optimum, and sometimes, this solution could be far from the global optimum. Therefore, with this new approach, using the local search within a more complex algorithm we aim to reduce this limitation in order to be able to obtain better solutions.

4.5.1 Algorithm Structure

The structure of SVNS is extremely similar to the one of the VNS. The main feature consists in restarting the local search, but from another initial solution. It faces the exploration of remote areas. That is, it degenerates, in some sense, in a heuristic multi-boot, in which an iterative local search is applied from solutions generated at random.

The steps required to implement the SVNS algorithm are shown in Algorithm 4.

Distance Function

Definitely the most characteristic feature of the SVNS algorithm is the incorporation of a distance function to make a directional improvement and comparison between neighborhood solutions. The function is used to measure the distance between two solutions that are the current solution S and the local optimum found S'' . The distance function is identified by the $Distance(S'', S)$ function (line 9, Algorithm 4).

In the specific case of FAP, the distance value is calculated by passing over all the TRXs of S'' and S and seeing how many TRX (out of 2 612 possible,

Algorithm 4 Pseudocode for Skewed Variable Neighbourhood Search algorithm.

```

1:  $S \leftarrow \text{CreateInitialSolution}$                                  $\triangleright$  at random
2:  $S^* \leftarrow S$                                                $\triangleright$  so far  $S$  is the optimal solution( $S^*$ )
3:  $k \leftarrow 1$ 
4: while  $k \leq k_{max}$  do
5:    $S' \leftarrow \text{MutationInEnvironment}_k(S)$ 
6:    $S'' \leftarrow \text{LocalSearch}(S')$ 
7:   if  $S''$  is better than  $S^*$  then
8:      $S^* \leftarrow S''$ 
9:   end if
10:  if  $S''$  is better than  $(S + alfa * Distance(S'', S))$  then
11:     $S \leftarrow S''; k \leftarrow 1$ 
12:  else
13:     $k \leftarrow k + 1$ 
14:  end if
15: end while
16: return  $S^*$ 

```

in case of Denver instance) have assigned different frequency values. At start, the distance value is set to 0, but when comparing the two TRX, if they have different values, the distance value is incremented. In the Denver instance case, it can generate a value between 0 and 2 612. The value 2 612 indicates that both solutions being compared have different values for all the TRXs. Otherwise, if the value computed is 0, this means that both solutions are exactly the same (all the TRXs from both solutions have exactly the same frequencies assigned to their TRXs). Additional information about the used instances is given in Chapter 6.1.

The $alfa$ value $\epsilon[0.0, 1.0]$ represents the magnitude of the fitness value. In the worst case, with $alfa=1$ it is permitted a maximum deterioration of the fitness value, that will be 2 612 (number of TRXs of the network instance represented by n). Using an $alfa=0$, the results of the SVNS are very similar to those achieved by the VNS, because the algorithm does not consider the search space surrounded by the distance function.

The influences of the several values that can be set for the $alfa$ parameter in the global performance of the algorithm have been analyzed. The accomplished results are available in Chapter 6 that is devoted to present the experiments results accomplished by metaheuristics investigated.

4.6 Summary

Throughout this chapter we have described particular implementations that were specifically addressed to tackle FAP, which has been previously introduced. Mainly, the concepts approached in this chapter represent a single-objective formulation

of FAP. This version of FAP only uses the interference cost value as fitness function to evaluate the quality of the given candidate solutions.

First, we have started by describing a population-based algorithm and then we detailed two trajectory-based algorithms. Additionally, some considerations about the FAP formulation and their integration in the algorithms features were also given.

In the algorithms descriptions we have integrated their classic structure and also the hybridizations that we have designed to be included in the new structure that we are proposing. Although some methods are specific for FAP, it is possible to generalize the more relevant aspects of each proposed algorithm. Thus, they can be used when addressing other types of optimization problems.

Multiobjective Metaheuristics: New Models

In this chapter, we start by presenting the new multiobjective algorithms that we have designed and adapted to be able to address the real-world FAP. Following the multiobjective formulation of FAP that we have presented in a previous chapter, we now describe the main features of multiobjective metaheuristics, which were proposed, based on our previous study focused on a single-objective version of FAP.

First, we give some global concepts related to FAP and how to evaluate the quality of the accomplished solutions. Next, several multiobjective concepts incorporated in the proposed algorithms are detailed and also formulated. The main content is devoted to explain the algorithms and how they can be tailored to FAP. Although algorithms are presented considering FAP characteristics, they can be generalized to any multiobjective optimization problem.

Besides the new approaches that we propose with the algorithms being described in this chapter, we also approach some of the characteristics of well-known multiobjective algorithms (NSGA-II and SPEA2). Although they are not new algorithms they were included in our work because they were used in the comparison analysis carried out to the algorithms that we have designed.

5.1 Introduction

In multiobjective optimization (MO) the main goal is to optimize the values of several objectives at the same time, but usually there does not exist a single point within the search space where all the objectives reach their individual optimum. Instead, we have a set of solutions that can be seen as optimal solutions. This set of so-called Pareto optimal solutions is called a Pareto optimal set or Pareto Front.

A solution belongs to the Pareto optimal set if none of the objective function values can be improved without impairing the value of, at least, one other objective (see Chapter 3.3).

The remaining question is how to identify which solutions belong to the Pareto Front. All the solutions taking part of the Pareto Front are known as nondominated solutions and they are the ones that are not dominated by any of the remaining known solutions. This is the main concept considered when evaluating the quality of the solutions in multiobjective optimization. In our work, the nondominated solutions are constantly being identified in the algorithms that we proposed. Therefore, it is one of the most relevant concepts of our approach.

A solution representation, the crossover and the mutation operation are key concepts in evolutionary algorithms and they all are problem dependent, but are not affected by multiple objectives. Therefore, our new multiobjective metaheuristics still continue to use the same codification as the one used with the single-objective metaheuristics. The major differences are in the evaluation of the fitness function and in their representation. In our specific formulation of FAP, the followed concepts are explained every time they are required to understand the methodology of the algorithm.

Our multiobjective approach is based on the absence of weights for the objectives being minimized. Therefore, the dominance between solutions is the criterion used to evaluate the fitness value (quality) of the solutions. The accomplished solutions are expressed in the Pareto optimal (nondominated) solutions. The reason for this is that it cannot be said that a solution is better than the other; they are equally good if they are not dominated by any other known solution. In conclusion, all the presented multiobjective algorithms have as outcomes a Pareto Front.

5.2 Multiobjective Optimization: Basic Concepts

Many real world problems require the simultaneous optimization of multiple, often competing, objectives. Usually, the solution to such problems is computed by combining them into a single criterion to be optimized, according to some

fitness function. Other approaches are also possible, specially for situations where transforming all the objective in only one fitness function can be an inefficient process. The alternative is to treat the whole problem as a multiobjective problem with non-commensurable objectives.

Optimization problems involving multiple and conflicting objectives are known as multiobjective optimization problems. Initially, they were often approached by aggregating the several objectives into a scalar function [114]. However, this approach has evolved and it has become more profitable to find a set of optimal trade-offs solutions, the so-called Pareto optimal set. Thus, a number of solutions can be found which provide not a unique solution but instead a set of potential solutions that are known as Pareto Front. Finally, the optimal solutions are chosen among this set of possible solutions.

In this type of optimization problems, some quality measures are frequently applied to compare potential solutions. We have selected frequently used indicators as the hypervolume indicator and the coverage relation. They will be explained in more detail in Chapter 6.3.

5.2.1 Multiobjective Formulation

In a multiobjective optimization problem it is possible to consider two main concepts to compare solutions and relate them regarding the best profitable one. The two possibilities are: a solution either dominates the other solutions, or dominates neither. A solution is known as being dominated (*isDominated*) when there is at least one other solution that improves at least one of the objectives.

Formally, a mathematical formulation for multiobjective optimizations can be defined as: if the problem has n different functions to be minimized:

$$\begin{aligned} \text{Minimize } & y = f(x) = f_1(x), f_2(x), \dots, f_n(x) \\ \text{where } & x = (x_1, \dots, x_m) \in X \\ & y = (y_1, \dots, y_n) \in Y \end{aligned} \tag{5.1}$$

where the decision vector x belonging to the X parameter space and with the objective vector y , from objective space Y .

5.2.2 Dominance Concept

Most well-known MO metaheuristics use the concept of domination. In MO, two solutions are compared on the basis of whether one solution dominates the other or not.

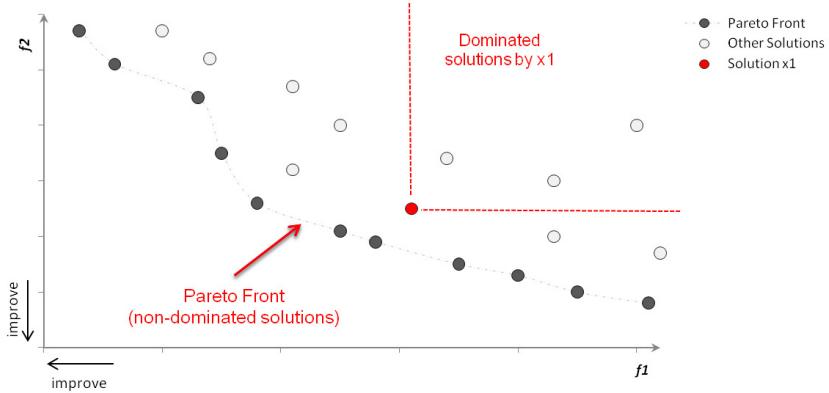


Figure 5.1: The dominance concept in the two-objective case.

In multiobjective optimization, the *dominance* is the base concept. Therefore, dominance is used to compare candidate solutions in order to identify what the more efficient ones are, regarding the objectives that are being considered.

Formally, if we consider that a multiobjective problem has n different functions to be maximized ($f_1(x), f_2(x), \dots, f_n(x)$) - considering that in our FAP scenario we try to minimize two objectives. A decision vector representing a solution $x_1 \in X$ *dominates* a solution $x_2 \in X$ (defined as $x_1 \succ x_2$) iff [150]:

$$\begin{aligned} \forall i \in \{1, 2, \dots, n\} : f_i(x_1) &\geq f_i(x_2) \wedge \\ \exists j \in \{1, 2, \dots, n\} : f_j(x_1) &> f_j(x_2) \end{aligned} \quad (5.2)$$

where x_1 covers x_2 ($x_1 \succeq x_2$) if, and only if $x_1 \succ x_2$ or $f(x_1) = f(x_2)$. It means that solution x_1 dominate solution x_2 and in the opposite direction that x_2 is *Dominated* by solution x_1 .

In other words, a solution x_1 is said to dominate solution x_2 if all objective values of x_1 are at least as good as those of x_2 , with at least one strictly better objective value. This dominance relationship is sometimes referred to as a weak dominance relation [140]. A strong dominance relation can be defined as follows: A solution x_1 strongly dominates a solution x_2 , if solution x_1 is strictly better than solution x_2 in all n objectives.

On the other hand, a solution is nondominated if it is not dominated by any solution. Thus, the nondominated solutions inside a population of individuals are represented by all those solutions that are not dominated by any of the remaining solutions in that same population. These nondominated solutions represent the

Pareto Front. Indeed, a single optimal solution rarely exists in a multiobjective problem; therefore an optimal solution in multiobjective is usually represented by a set of solutions defined by the Pareto Front concept. In MO optimization problems, the concepts of nondominated and Pareto optimal solutions are essential to formulate a MO approach.

As previously mentioned, the dominance concept represents the main idea followed by the metaheuristics that we present in this work. Based on the above definition, the Pareto-optimal solutions are the *nondominated* solutions that can be defined formally as:

1. Let $x_1 \in X$ be a decision vector.
 2. Solution x_1 is said to be nondominated regarding a set $X' \subseteq X$ iff there is no vector in X' which dominates x_1 ; formally:
- $$\nexists x'_1 \in X' : x'_1 \succ x_1 \quad (5.3)$$
3. Solution x_1 belongs to the Pareto-optimal iff x_1 is nondominated regarding X .

Concluding, a solution that belongs to the Pareto optimal set cannot be improved in any objective without degradation of at least one of the other objectives. They represent the best solutions achieved until a given time and they form the Pareto Front.

Figure 5.1 shows the dominance concept visually. Considering, for example, that the scheme is related to a two-objective optimization problem and that solution x_1 is a potential candidate solution of the problem. The dominance of x_1 represents all the solutions that are in the region defined by the area above the axis. In this case, using a two-objective example, the solution x_1 dominates three other solutions. Thus, the solutions dominated by x_1 represent worst solutions for the problem.

The relevance of the dominance concepts in multiobjective optimization, and therefore in this work, is that it permits the identification of the solutions that are nondominated, i.e. the solutions that are not dominated by any other solutions in the solution space. From all the search space of potential solutions generated by a metaheuristic, the ultimate goal is to identify the best solutions, which are the ones belonging to the Pareto Front.

5.3 Evaluating the Quality of Solutions in Multiobjective FAP

Following the main concepts approached previously it is possible to define that in multiobjective the concept of Pareto dominance is commonly used to compare candidate solutions. Thus, by its definitions an optimal solution set means that it will contain all the best solutions. Therefore, following this point of view we assume that until a specific time period there are no other known solutions superior to them in all objectives that are being considered.

During algorithms execution, several potential candidate solutions are generated and consequently to each one the dominance concept is computed. Thus, every time the algorithm needs to take a decision regarding which the best solutions are, the dominance information is the base of that decision process.

In our specific FAP scenario a good solution aims to accomplish a frequency plan that provokes the lowest number of interferences in the network. In practice, when the evaluation regarding the quality of a frequency plan is performed the costs associated with the *interference cost* and the *separation cost* are considered. Besides these intrinsic values that characterize a valid solution in FAP, additional information is computed to help in the evaluation of the accomplished solutions. It implies that for an individual, besides both costs previously mentioned, it is also considered a *fitness* value that represents a scalar value to be minimized.

The *fitness* function that we used is defined by two different terms. The first term considers the computation of the two costs related to the frequency assignment of that individual. Then, the computation of the final fitness value is used and it considers the context of the individual inside the population of solutions from which is part of. Therefore, in our multiobjective scenario the final fitness value is obtained considering the quality of the individual regarding to the remaining individuals belonging to that population.

The multiobjective algorithms that we are proposing use simultaneously the fitness value and the dominance concepts frequently related to multiobjective optimization. Specifying, the quality measurement computes the dominance value as:

$$\text{fitness}^i = NP * N_{IsDominated}^i + N_{Dominates}^i \quad (5.4)$$

where the NP parameter represents the size of the population where the individual i is. The size represents the number of individuals that are competing against the individual for which the fitness value is being computed. The $N_{IsDominated}^i$ is the number of individuals that dominate individual i , while the $N_{Dominates}^i$ is the number of individuals that are dominated by individual i . An

individual with a lower fitness value is preferred instead of others with higher fitness value.

In the fitness value computation, the number of dominating individuals establishes a primary ranking in the population. In case of existing individuals that are dominated by an equal number of individuals, those that dominate fewer individuals are preferred. Following this approach we are promoting individuals from less crowded regions in the objective space. If the complete population consists of nondominated individuals, selection acts as a mere uniform selection [147].

Both NSGA-II and SPEA2 used their own original fitness value. The remainder metaheuristics used the described fitness value.

5.4 Differential Evolution with Pareto Tournaments Algorithm

Multiobjective Differential Evolution with Pareto Tournaments (DEPT) is a new algorithm that is based on classic Differential Evolution [116, 117, 135]. The classic DE algorithm has been successfully applied in many optimization problems, including multiobjective optimization problems since 1994 [116, 117, 135, 136].

In our proposing of DEPT, the base fundament is the incorporation of Pareto Tournaments in the parental selection to address a multiobjective FAP optimization problem. The DE arithmetic operators, such as the selection, crossover and mutation still make part of the structure of the DEPT algorithm. Our enhancements consist in additional mechanisms, developed specifically to support multiobjective optimization problems, which consist in the incorporation of the dominance concept in a Pareto Tournament to perform the selection of the best solutions.

Next, the main steps proposed to the new algorithm are outline in more detail.

5.4.1 Algorithm Structure

The pseudocode for DEPT is given in Algorithm 5 where NP is the population size. DEPT starts by generating the initial population ($P^{t=0}$) randomly, containing NP individuals represented as vectors, where t represents a specific generation.

It can be seen that the algorithm structure looks almost identical to that of classic Differential Evolution given in Chapter 4.3, except that new individual $p^{t+1,i}$ is generated through tournament between $c^{t+1,i}$ and $p^{t,i}$. The index i identifies the individual positions in the population.

Indeed, in DEPT, every individual from the current population, named *target* individual ($p^{t,i}$) serves in turn as a parent, while all the other individuals

Algorithm 5 Pseudocode for DEPT algorithm.

```

1:  $t \leftarrow 0$                                      ▷ initial generation
2: initialize population  $P^t$ 
3: evaluate population  $P^t$ 
4: while time-limit is not reached do
5:   for ( $i = 1$  to  $\in NP$ ) do
6:     create a child  $c^{t+1,i}$            ▷  $c$  is the  $x_{trial}$  individual
7:     evaluate child  $c^{t+1,i}$ 
8:     pareto tournament between  $c^{t+1,i}$  and  $p^{t,i}$  to get individual  $p^{t+1,i}$ 
9:   end for
10:   $t = t + 1$ 
11: end while

```

in the population may randomly participate in the creation of the *trial* individual ($c^{t+1,i}$). A fitness value computed using the two objective values is used to choose between these two potential solutions when performing the selection tournament. The tournament uses the Pareto dominance information of each one of the participating individuals.

In DEPT the creation of the child (*trial*) individual is done by applying a mutation and recombination operator. First of all, it starts by selecting the *target* individual ($p^{t,i}$) and after that, the algorithm chooses at random other three individuals from the current population (where $r_1 \neq r_2 \neq r_3 \neq i$ that are used as indices to index each parent vector).

Following this, the creation of the child individual ($p^{t+1,i}$) is done using a DEPT scheme, that e.g. it starts by building a weighted difference vector with individuals x_{r1}^t and x_{r2}^t . This operation also uses the F parameter that represents the scaling factor, which controls the differential variation to be applied in the child individual. Next, the third randomly chosen individual (x_{r3}) is added to the last result.

A crossover probability (CR parameter) is also used to control the percentage of genes that will be changed in the child individual.

5.4.2 Pareto Tournaments

Pareto tournaments [147] are implemented in DEPT to select the individuals that survive to the next generation. It is based on a ranking strategy that takes into considerations the concept of domination. In DEPT the Pareto Tournament is performed in line 8 of Algorithm 5 to make a selection between two potential solutions.

To every individual, the dominance concept is used to rank the individuals. Thus, it is assigned the fitness value as the one defined by Equation 5.4. The

individual *fitness* value represents the scalar value to be minimized.

As mentioned previously, the fitness value is the base criterion used to select the best profitable individual between the existent alternatives solutions. The fitness value is always computed considering the current solutions that are in the population. This way, selecting the individual with the smallest fitness originates the selection of an individual that, compared to the remaining solutions, is a better solution.

The Equation 5.4 defines the way fitness value is computed and it takes into consideration the existence of a population P^t . The Pareto tournament occurs when it is necessary to select the individual that will prevail to the next population. The solution returned by the tournament is based on a decision that has computed the dominance information to every member of the population.

Formally, to every member of the population P^t is computed the $N_{isDominated}^i$ that is the number of individuals that dominate individual i while $N_{dominates}^i$ is the number of individuals that are dominated by individual i .

Thus, child $c^{t+1,i}$ joins population P^t to compute $N_{isDominated}^i$ and $N_{dominates}^i$, and accordingly *fitness* value, for both $p^{t,i}$ and $c^{t+1,i}$. After computing to every population member their fitness value, a tournament between $p^{t,i}$ and $c^{t+1,i}$ is then implemented based on their *fitness* value.

The Pareto tournament is performed following the next considerations:

1. The individual with the smallest *fitness* wins.
2. In case of a tie, the winner is determined according to their relevant dominance.
3. If $p^{t,i}$ and $c^{t+1,i}$ continue to tie, the winner will be the one having the smallest interference cost.
4. If $p^{t,i}$ and $c^{t+1,i}$ tie again, the winner will be the one having the smallest separation cost.
5. If tie happens again, $p^{t,i}$ survives.

The individual return by the Pareto tournament represents the solution that dominates more solutions. Thus, if it dominates more solutions it means that it has smaller values for both objectives functions that were formulated for the problem we are addressing.

5.5 Multiobjective Variable Neighborhood Search Algorithm

The Multiobjective Variable Neighborhood Search (MO-VNS) algorithm is a new proposed developed by us to solve a multiobjective optimization problem. It implements dominance concept in the standard VNS [58, 60] to select the best solution.

As well as its base algorithm, the MO-VNS starts from a random initial solution and performs several changes in neighborhood. Neighborhood solutions are generated by a mutation operation until no further improvements are possible. MO-VNS also changes the environment when the obtained solution is worse than the current solution.

The complete pseudocode of MO-VNS is given in Algorithm 6 and it is explained in more detail in the next section.

5.5.1 Algorithm Structure

The MO-VNS algorithm gradually applies a mutation operator to generate a solution S' from the current S , conducted according to the type of environment used in each iteration $N_k(S)$, where $N_k(k = 1, \dots, k_{max})$ denotes the potential neighborhood solutions.

If the new candidate solution S' improves the current solution S , the next iteration of the algorithm will start again to use the first neighborhood environment N_1 . Otherwise, the algorithm will go to the next neighborhood environment ($k \leftarrow k + 1$). To handle a multiobjective optimization problem we propose to MO-VNS the incorporation of a set containing the dominating solutions.

The main feature of multiobjective VNS approach is that it uses the dominance concept to select the best solutions created in each iteration of the algorithm.

Algorithm starts from a random initial solution, and neighborhood solutions - generated with the mutation operator (line 10, Algorithm 6), are generated until no further improvements are possible. The algorithm also changes the neighborhood environment when the obtained solution is worse than the current solution (line 19, Algorithm 6).

In the beginning, the set of *pareto solutions* is empty (line 1, Algorithm 6). It will contain all the nondominated solutions that represent the best solutions found so far. The size of the set representing the *pareto solutions* is not always the same since it changes along the run. In every iteration, a comparison between existent *pareto solutions* and the candidate solution is performed to see if it is a better

solution (line 4 and line 11, Algorithm 6). After dominance check operation, in the *pareto solutions* will only prevail the nondominated solutions.

In the MO-VNS when exploring the solutions from neighborhoods environment, if a better solution (using the dominance concept) is found, it is added to the *pareto solutions* set (line 11 to line 14, Algorithm 6). To update the *pareto solutions* set an additional method is implemented. It compares all the solutions inside the set, so it will remove all the solutions that are dominated by the newly found solution. Therefore, the process of adding and removing solutions from the *pareto solutions* set is performed in the beginning of the algorithm and every time the algorithm creates a new candidate solution by using the mutation operator.

Algorithm 6 Pseudocode for the MO-VNS algorithm.

```

1: paretoSolutions  $\leftarrow$  0
2: while time-limit is not reached do
3:    $S \leftarrow$  generateSolution                                 $\triangleright$  at random
4:   if  $S$  dominates any Pareto solution in ParetoSolutions then
5:     add  $S$  into paretoSolutions
6:     remove solutions in paretoSolutions dominated by  $S$ 
7:   end if
8:    $k \leftarrow 1$ 
9:   while  $k \leq k_{max}$  do
10:     $S' \leftarrow$  mutationInEnvironment $_k(S)$ 
11:    if  $S'$  dominates any Pareto solution in ParetoSolutions then
12:      add  $S'$  into paretoSolutions
13:      remove solutions in paretoSolutions dominated by  $S'$ 
14:    end if
15:    if  $S'$  dominates  $S$  then
16:       $S \leftarrow S'$ 
17:       $k \leftarrow 1$ 
18:    else
19:       $k \leftarrow k + 1$ 
20:    end if
21:   end while
22: end while
23: return paretoSolutions

```

The mutation operator used in Algorithm 6 can be replaced by an improvement characteristic that we have designed to be incorporated specifically to address FAP. The originated improved version of MO-VNS is named GMO-VNS that represents the use of the *greedy mutation in environment approach**.

*A more detailed explanation about the greedy mutation operator is presented in Chapter 5.7.

Therefore, both MO-VNS and GMO-VNS algorithms have the same algorithm structure. The only difference lays in the mutation operator that is used. In the base approach the mutation operator ($mutationInEnvironment_k(S)$) function creates a new solution randomly, meaning that it assigns a random frequency value for a TRX considering only the unused frequencies of the sector in which the TRX is installed. The greedy operator uses the $greedyMutationInEnvironment_k(S)$ function instead of the $mutationInEnvironment_k(S)$ function.

5.5.2 Mutation Operator

The MO-VNS algorithm applies a mutation operator to generate new solutions. In the design of the MO-VNS algorithm two approaches of the mutation operator were considered. Both versions of the mutation operator were implemented using specific information from the problem domain.

Non-Greedy version

It assigns a random frequency value for a TRX from unused frequencies of the sector in which the TRX is installed. The set of available frequencies of the specific sector is updated accordingly.

Greedy version

The greedy mutation adopts the dominance concept and considers all the potential neighborhood solutions. The pseudocode for this approach is described in more detail in Section 5.7.

5.6 Multiobjective Skewed Variable Neighborhood Search Algorithm

Our new approach, named Multiobjective Skewed Variable Neighborhood Search (MO-SVNS) algorithm is a multiobjective version of the SVNS [58, 60], that is a known extension of the Variable Neighborhood Search (VNS) algorithm named Skewed Variable Neighborhood Search (SVNS).

The MO-SVNS incorporates the same concepts as the SVNS algorithm but additionally it combines on it the dominance concepts required to be able to address a multiobjective optimization problem. Similarly to its base algorithm, MO-SVNS also increases the size of neighborhood when the search does not move forward. The main feature in the variant MO-SVNS is that it restarts the algorithm, but from other initial solution. This approach permits it to face the exploration of remote areas, but unlike the MO-VNS version that drives the

search more randomly to distant neighborhoods; the MO-SVNS is more receptive to accepting a new solution, even if it is slightly worse than the previous solution.

The concept beyond this algorithm is that it keeps the best found solution and uses a distance function that quantifies the distance between two distinct solutions (S and S').

5.6.1 Algorithm Structure

In MO-SVNS, as shown in Algorithm 7, where $0 \leq \alpha \leq 1.0$ is a quality magnitude parameter, it implements Pareto dominance in SVNS to select the best solution in a multiobjective problem (see Chapter 4.5). MO-SVNS restarts from another initial solution to explore remote areas when it is lost in a local optimum which may be far from the global optimum (line 15, Algorithm 7). However, unlike MO-VNS, it is more receptive to accepting a new solution, even if it is slightly worse than the previous solution. It also increases the size of neighborhood when the search does not move forward (line 18, Algorithm 7).

The concepts behind it are keep the best found solution and quantify the distance between two distinct solutions S and S' (line 15, Algorithm 7) using a bi-dimensional distance function that represents the two different goals to be minimized.

As in MO-VNS, MO-SVNS also uses the *paretoSolutions* set to store the best solutions obtained by every algorithm iterations. Indeed, every time a new solution is found it is compared with the remaining solutions that are inside the Pareto set (line 4 and 11, Algorithm 7). The new solution will prevail only if it is not dominated by any other solutions that are in the Pareto set.

A solution S is better than a solution S' if it dominates, meaning that it is better in both objectives.

5.6.2 Distance Function

As pointed out above, the main difference between MO-SVNS to the variant MO-VNS is that MO-SVNS uses a bi-dimensional distance function that represents the two different goals to be minimized. In our investigation, and because FAP multiobjective formulation considers two minimization functions, these are the dimensions used to quantify the distance value between the two solutions S and S' .

The two dimensions are expressed as independent values (line 15, Algorithm 7). The distance value is obtained computing the difference between the lowest cost value of S and S' and the highest cost value of all solutions that are inside the set of Pareto solutions.

Algorithm 7 Pseudocode for the MO-SVNS algorithm.

```

1: paretoSolutions  $\leftarrow 0$ 
2: while time-limit is not reached do
3:    $S \leftarrow \text{generateSolution}$  ▷ at random
4:   if  $S$  dominates any Pareto solution in paretoSolutions then
5:     add  $S$  into paretoSolutions
6:     remove solutions in paretoSolutions dominated by  $S$ 
7:   end if
8:    $k \leftarrow 1$ 
9:   while  $k \leq k_{max}$  do
10:     $S' \leftarrow \text{mutationInEnvironment}_k(S)$ 
11:    if  $S'$  dominates any Pareto solution in ParetoSolutions then
12:      add  $S'$  into paretoSolutions
13:      remove solutions in paretoSolutions dominated by  $S'$ 
14:    end if
15:    if  $S'$  dominates  $(S + \alpha * \text{distance}(S', S))$  then
16:       $S \leftarrow S'$ ;  $k \leftarrow 1$ 
17:    else
18:       $k \leftarrow k + 1$ 
19:    end if
20:   end while
21: end while return paretoSolutions

```

In FAP case, to compute the *distance* function for the interference cost (*IC*), it is considered the difference between the best interference cost value (from solutions S and S') and the worst interference cost value from all the solutions inside the *paretoSolutions* set. Therefore this distance_{IC} value is expressed by:

$$\text{distance}_{IC} = |\text{Best}_{IC}(S', S) - \text{Worst}_{IC}(\text{FromParetoSolutions})| \quad (5.5)$$

The distance value for the separation cost (*SC*) is expressed identically but using instead the separation cost value as defined by:

$$\text{distance}_{SC} = |\text{Best}_{SC}(S', S) - \text{Worst}_{SC}(\text{FromParetoSolutions})| \quad (5.6)$$

5.7 Greedy Versions of MO-VNS and MO-SVNS Algorithms

In both MO-VNS and MO-SVNS a mutation operator is used to generate new candidate solutions. To improve this operator an enhanced approach was proposed. The enhanced method uses instead of the randomly operator a more greedy approach that is named *greedyMutationInEnvironment*.

5.7.1 Greedy Mutation Operator

The greedy mutation operator considers that in a multiobjective problem, solutions can be gathered applying the dominance concept to generate the final Pareto Front. This method reflects the dominance concept, which is applied considering all the potential neighborhood solutions.

The greedy mutation operator, as shown in Algorithm 8, considers all potential neighborhood solutions. In the beginning, the sector in which a TRX is installed and the corresponding unused frequencies are identified. Then, the best solution in terms of dominance is chosen from all possible solutions by testing all frequencies available to that TRX.

In the beginning of the greedy mutation method, for every TRX in the solution, it starts by identifying in which sector the TRX is installed. Then, it gets the range of unused frequencies for that specific sector. In the next step all the possible solutions are analyzed and then it chooses the one with the best fitness value. It means that it will test for a specific TRX, one at a time, all the available frequencies of that TRX. To choose the best frequency it computes and then compares the dominance concept for all these potential solutions. The selected frequency value will be the one that dominates more solutions. This procedure will return the best neighborhood solution.

This method uses a temporary population that only contains the candidate neighborhood solutions. The individuals inside are eliminated right after the mutation operation.

The greedy mutation returns to the main algorithm the best solution found within the possible candidate solutions.

Algorithm 8 Pseudocode for *greedy mutation in environment* function.

```

1: function GREEDYMUTATIONINENVIRONMENT( $S, k$ )
2:    $k_{probability} \leftarrow k/10$                                  $\triangleright$  mutation probability
3:   for  $TRX_{id} = 0$  to  $allTRXinNetwork(S)$  do
4:     if  $rand \leq k_{probability}$  then       $\triangleright$  rand is a random number  $\in [0, 1]$ 
5:        $sector \leftarrow getSector(TRX_{id})$ 
6:        $unusedFreq \leftarrow getUnusedFreq4Sector(sector)$ 
7:        $population \leftarrow generateSolutions(unusedFreq)$ 
8:        $computeDominance(population)$ 
9:        $best \leftarrow selectBestSolution(population)$ 
10:    end if
11:   end for
12:   return  $best$ 
13: end function

```

Shortly, the *GreedyMutationInEnvironment* function is a newly proposed variant for the mutation operator that we have implemented into MO-VNS and MO-SVNS. The two new algorithms are identified by using the G denomination (GMO-VNS and GMO-SVNS), which are extensions of their base algorithm, namely MO-VNS and MO-SVNS, respectively.

5.8 Multiobjective Artificial Bee Colony

The proposed Multiobjective Artificial Bee Colony (MO-ABC) algorithm is a new metaheuristic designed by us to address the multiobjective formulation of FAP. It is based on the standard Artificial Bee Colony (ABC) algorithm [71, 73] which was inspired by the behavior of real bees. The MO-ABC algorithm main characteristic is the incorporation of Pareto Tournaments (PT) to address the multiobjective FAP formulation.

The main goal when solving a multiobjective problem is to obtain the Pareto optimal set and, consequently, the Pareto Front. When metaheuristics are applied, the goal becomes to obtain an approximation of the Pareto optimal set having two properties: convergence to the Pareto optimal front and uniform diversity. The first property ensures the generation of near-optimal Pareto solutions, while the second property indicates a good distribution of the obtained solutions around the Pareto optimal front, so that no valuable information is lost [140]. The proposed MO-ABC algorithm incorporates a crowding distance function in the selection phase, which is performed in the Pareto tournament and it permits to obtain spread Pareto Front solutions.

5.8.1 Standard ABC Algorithm

Standard ABC algorithm [71–74] was proposed by Karaboga in 2005 and simulates the behavior of a honey bee swarm. Based on structure of ABC in which there is a colony of bees, grouped under three types: employed bees, onlookers and scout bees, we have enhanced the basic features of the algorithm to address a multiobjective formulation. ABC is a population-based algorithm, in which the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source represents its quality (fitness) value.

Half of the colony is composed by employed bees and the other half by onlooker bees. The number of employed bees in the colony is equal to the number of used food sources around the hive. The colony size is one of the algorithm parameters. In ABC, an employed bee becomes a scout bee and starts to research a new food source (neighboring solution), when the previous one has been exhausted by all the bees.

The main steps of the classic algorithm are given below [71]:

1. Initial food sources are produced for all employed bees
2. REPEAT
 - a) Employed bees move onto their food sources and determine their nectar amounts.
 - b) Onlookers bees move onto the food sources and determine their nectar amounts.
 - c) Abandoned food sources are determined and are replaced with the new food sources discovered by scouts.
 - d) The best food source found so far is memorized.
3. UNTIL (requirements are met)

5.8.2 Pseudocode of the MO-ABC Algorithm

Algorithm 9 Detailed pseudocode for Multiobjective Artificial Bee Colony (MO-ABC) algorithm.

- 1: Initialize the population (P) of employed bees $x_i, i = 1, \dots, NB/2$
 - 2: Evaluate the Population
 - 3: $t \leftarrow 0$ ▷ initial cycle
 - 4: Initialize *Pareto Front* solutions ▷ at the beginning it is empty
 - 5: **while** not *time-limit* **do**
 - 6: Produce new neighboring solutions v_i for employed bees and evaluate them
 - 7: Apply *Pareto Tournament* for employed bees (between x_i and v_i)
 - 8: Calculate probability values P_i for solution x_i
 - 9: Produce new neighboring solutions v_i for onlookers from solution x_i selected depending on P_i and evaluate them
 - 10: Apply *Pareto Tournament* for onlookers bees (between x_i and v_i)
 - 11: Determine the abandoned solutions for the scouts, if exist, and replace everyone with a new produced solution x_i
 - 12: Update the current Pareto Front
 - 13: $t \leftarrow t + 1$ ▷ go to next cycle
 - 14: **end while**
-

In the multiobjective MO-ABC metaheuristic a Pareto Tournament was incorporated to select the food sources (solutions) to remain in the population (colony of bees). The Pareto Front evolves at every algorithm iteration and it only contains the nondominated food sources. An outline of the algorithm is shown in Algorithm 9.

The MO-ABC starts with an initial population of employed bees ($P_{t=0}$) of $NB/2$ solutions - food source positions (line 1, Algorithm 9). Next, the population is subjected to repeated cycles in the search process of the employed, onlooker, and scout bees, respectively. In each algorithm cycle, an employed bee produces a modification on the food source position in her memory and discovers a new neighboring solution (line 6, Algorithm 9). The new neighboring solution $v_{i,j}$ of solution $x_{i,j}$ for the employed bees is obtained using a mutation function. Algorithm 10 describes how the new neighboring solution is generated, which is similar to the one implemented by GMO-VNS and GMO-SVNS (see Section 5.7).

Algorithm 10 Pseudocode for the mutation operator of MO-ABC.

```

1: for all the TRX in the network do
2:   if TRX is to be changed then      ▷ based on a mutation probability
3:     identify the TRX sector
4:     get the set of unused frequencies
5:     generate a new solution for every available frequency
6:     compute the dominance for the new generated solutions
7:     select the solution with the best dominance
8:   end if
9: end for
return best found solution

```

The mutation is applied to every elements (TRXs) of the solution, based on a probability of mutation (PM parameter). For every TRXs, the greedy mutation procedure consists in: all possible *unused frequencies* inside the sector of the TRX being tested, in order to identify what is the best frequency. The selection of the winning neighboring solution (the new frequency to be assigned to the TRX) is performed by using the dominance concept. This way, the new frequency value to be assigned to the TRX will be the one that dominates more solutions. The *unused frequencies* inside a sector are all the frequencies not assigned to the TRXs, considering also their adjacent frequencies, e.g, when assigning a frequency value f to the TRX also the $f - 1$ and the $f + 1$ frequencies become unavailable.

In the next step of MO-ABC, a Pareto Tournament (line 7, Algorithm 9) is performed to select the remaining solutions (see Section 5.8.3). In the Pareto Tournament, if the new neighboring solution is a better solution (it has a smaller multiobjective fitness value) than that of the previous food source, the bee memorizes the new neighboring solution and forgets the old one. Otherwise she keeps the original solution in her memory.

After all employed bees complete the search process they share the position information of the food sources with the onlooker bees. In the next step, onlooker bees choose a new neighboring solution depending on its probability value (P_i) - line 8, Algorithm 9. To compute the probability value, all bees are sorted

as: first, by the Pareto Front number and secondly by the crowding distance. Then, a linear bias function is used to compute a (b_i) that is assigned to the i -th ranked element, where $b_i = 1/i$. The probability that the i -th ranked solution is selected is $P_i = b_i / \sum_{j=1}^{|NB|} b_j$, where NB is the number of bees. Every onlooker bee evaluates the nectar information taken from all employed bees and then chooses the solutions depending on their nectar amounts (multiobjective fitness value). Again, the selection is performed applying a PT between the solution x_i and the new neighboring solution v_i produced by the onlooker bees (line 10, Algorithm 9).

Next, the abandoned solutions are determined and new neighboring solutions are produced to replace the abandoned ones by scouts. The new neighboring solution is generated using the greedy mutation method. A solution is abandoned (line 11, Algorithm 9) if it cannot be further improved. This criterion is based on a predetermined number of cycles (*limit* parameter). After employed bees, onlooker bees and scout bees have performed their work the best solutions are memorized, updating the Pareto Front (line 12, Algorithm 9).

Pareto Front MO-ABC is a multiobjective algorithm, therefore solutions are represented by a Pareto Front (to be maintained and built in every algorithm iteration). Initially the Pareto Front is empty (line 4, Algorithm 9). Then, after finishing the first iteration, for all the bees (employed, onlooker and scout bees) the nondominated solutions are calculated. They represent the first Pareto Front. Next, at every iteration the nondominated bees are computed for all the bees in the colony. It represents the current Pareto Front. This process is repeated iteration after iteration, until the stop condition is met. So the end result of the algorithm is a complete Pareto Front defined by all the nondominated bees.

5.8.3 Pareto Tournament

The Pareto Tournament (PT) is performed in MO-ABC to select the solutions that will survive to the next iteration of the algorithm. A solution (food source) has a multiobjective fitness value that is a scalar defined as:

$$\text{fitness}^i = NB * N_{IsDominated}^i + N_{Dominates}^i \quad (5.7)$$

where NB is the number of bees in the population, $N_{IsDominated}^i$ is the number of individuals that dominate individual i while $N_{Dominates}^i$ is the number of individuals that are dominated by individual i . An individual with a lower fitness value is more desirable.

Every time a bee produces a new solution a selection is performed using the Pareto Tournament technique. Therefore, a solution v_i joins bee colony to compute $N_{Isdominated}^i$ and $N_{Dominates}^i$ and accordingly fitness , for both x_i^t and v_i^t . The tournament between both solutions is performed based on the fitness value:

1. The solution with the smallest fitness value is chosen.
2. In case of tie, the winner is determined according to their relevant dominance.
3. If x_i^t and v_i^t continue to tie, the winner will be the one with the highest crowding distance.
4. If tie happens again, x_i^t survives, when PT is being applied in the employers bees case. Otherwise, the v_i^t survives when it is being applied in the onlooker bees case.

PT aims to select the best food sources and it is mainly based on the dominance fitness function.

Crowding distance The crowding distance operation is performed when a tie happens in the Pareto Tournament. If the fitness values of both solutions in terms of nondominance information are equal, the crowding criterion is used to select the individual that gives the largest crowding distance. Using the crowding concept to untie solutions implies that solutions with higher crowding distance are considered better solutions, as it introduces more diversity in the population (colony of bees).

Formally, the crowding distance is defined as the circumference of the rectangle defined by its left and right neighbors, and infinity if there is no neighbor [119]. Therefore, when different solutions have the same fitness value it is chosen the one that introduces more diversity in the bees colony.

The crowding distance operation is performed similarly as in the well-known multiobjective algorithm NSGA-II [32]. To successfully apply the crowding operator all bees in the colony must be sorted out in different Pareto Fronts. The first Pareto Front is the one that contains the individuals that are not dominated. To calculate the second Pareto Front, the individuals of the first front are ignored, and it is recalculated (without taking into account those individuals) which individuals are not dominated by anyone. To calculate the third Pareto front, we eliminate the individuals in the first two fronts, and recalculate the new head of Pareto Fronts (nondominated solutions), and so on. In summary, the individuals in the population can be grouped into a set of N different Pareto Fronts.

In our algorithm, if two solutions have equal fitness that means they are in the same Pareto Front (within all possible). Therefore, the crowding distance has to be calculated using only points of that Pareto Front (and not all individuals of the population).

5.9 Other Multiobjective Metaheuristics: Comparative Overview

Besides the metaheuristics proposed by us, which were previously presented, other approaches were implemented to address FAP. Due to the enormous range of available multiobjective metaheuristics, we have decided to use in the comparison phase of our metaheuristics other metaheuristics with higher relevance in the area. As a result, we have selected some of the most efficient algorithms for multiobjective optimization that have demonstrated their performance on a number of benchmark problems. These algorithms are known as NSGA-II and SPEA2 and a brief description of their structures is given next.

5.9.1 Nondominated Sorting Genetic Algorithm - II

The Nondominated Sorting Genetic Algorithm II (NSGA-II) was presented by [32] and it has become very popular in multiobjective optimization. NSGA-II algorithm is an improvement of its previous version, which required more complexity at a computational level and also the lack of elitism and the need of an optimal value for the sharing parameter. This new approach, from previous NSGA, has been very effective to address multiobjective problems and some of their main characteristics are the incorporation of elitism and the fact that no sharing parameter needs to be chosen a priori. Because of its popularity in multiobjective optimization problems, we have implemented it to address FAP. Its performance was compared with our multiobjective algorithms that were developed to address FAP (see Sections 5.4, Section 5.5 and Section 5.6).

The NSGA-II algorithm starts with the creation of the initial population $P_{t=0}$ (line 1, Algorithm 11). Then the population is sorted based on non-domination into each front. This sort procedure creates a first front containing only nondominant solutions. In the second front there are the individuals dominated by the individuals in the first front only, and the fronts go on so. The NSGA-II uses the dominance to create a ranking and classify the population into a number of layers (fronts). To every individual it is also assigned a *fitness* value that represents the ranking value based on the layer it belongs to (value of 1 to the first front, value of 2 to the second front, and so on).

The first layer represents the best layer in the population and therefore they are the best solutions.

To an individual, besides the assigned ranking value (*fitness* value), a *crowding distance* is also computed. This value measures how close an individual is to its neighbors. The crowding distance of an individual p is the Euclidean distance between each individual in a front based on their number of objectives. In our case it represents a two-dimensional hyper space. A large average crowding distance

will result in better diversity in the population. Eventually, this could permit the algorithm to obtain better results.

The selection of the individuals is based on the crowded comparison operator (\prec_n) that uses the rank and the crowding distance. Parents are selected from the population by using a binary tournament. An individual is selected if its rank is smaller than the other or if the crowding distance is greater than the other. The crowding distance is used if for both individuals there is a tie in the rank value. This selection permits the creation of a new offspring in the population that is sorted again based on nondominance. Then, only the best N (population size) individuals are selected.

Thus, the main operations required by the NSGA-II are the following:

1. Initialize population.
2. Nondominated sort.
3. Crowding distance.

Being p an individual, the n_p represents the number of individuals that are dominated by p and the S_p contains all the individuals that are dominated by p . In NSGA-II, after creating the initial population, it is sorted based on the nondominated individuals. The nondominated sort function uses the information about the set that an individual p dominates S_p and the number of individuals that dominate individual n_p . After the nondominated sort, it is applied the crowding distance function, that assigns to every individual a crowding distance value and it will be used with the rank value as criterion to select individuals.

In the NSGA-II algorithm pseudocode [32], the P_t , R_t and Q_t represent a population. After the creation of initial population, a combined population is generated by using the parent and the offspring population. The resulting combined R_t population has a size of $2 * N$. Next, the resulting population R_t is sorted according to the nondominated solutions (line 2, Algorithm 11). The \mathcal{F} represents the several nondominated fronts of R_t . Then, until the parent population is filled (line 5, Algorithm 11) it is computed the crowding-distance in front \mathcal{F}_i , and the i^{th} nondominated front is included in the parent population (line 7, Algorithm 11). The next front will also be checked for inclusion. When filled the parent population, it is sorted using (\prec_n) (line 10, Algorithm 11) and then the first $(N - |P_{t+1}|)$ elements in \mathcal{F}_i are chosen. Finally, a new population Q_{t+1} is created using the selection, crossover and mutation operators (line 12, Algorithm 11).

The sort of the nondominated solutions in the population (line 2, Algorithm 11) and the computation of the *crowding distance* (line 6, Algorithm 11)

Algorithm 11 Pseudocode for the main function of NSGA-II algorithm.

```

1:  $R_t \leftarrow P_t \cup Q_t$ 
2:  $\mathcal{F} \leftarrow \text{fastNondominatedSort } (R_t)$ 
3:  $P_{t+1} \leftarrow 0$ 
4:  $i \leftarrow 0$ 
5: while  $|P_{t+1}| + |\mathcal{F}_i| \leq N$  do
6:    $\text{crowdingDistanceAssignment}(\mathcal{F}_i)$ 
7:    $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i$ 
8:    $i \leftarrow i + 1$ 
9: end while
10:  $\text{Sort}(\mathcal{F}_i, \prec_n)$ 
11:  $P_{t+1} \leftarrow P_{t+1} \cup \mathcal{F}_i[1 : (N - |P_{t+1}|)]$ 
12:  $Q_{t+1} \leftarrow \text{makeNewPopulation}(P_{t+1})$ 
13:  $t \leftarrow t + i$ 

```

represent complex operations and they are central characteristics of the algorithm. A more detailed explanation about the algorithm structure and how these two operations are implemented is available in [32].

5.9.2 Strength Pareto Evolutionary Algorithm 2

Strength Pareto Evolutionary Algorithm 2 (SPEA2) [152] is an elitist alternative of Strength Pareto Evolutionary Algorithm (SPEA), which is a multiobjective optimization algorithm proposed by Zitzler & Thiele [151]. The main power of SPEA2, compared to its ancestor, lies in the elitism preserved operation. Like NSGA-II, SPEA2 is one of the most popular multiobjective evolutionary algorithms that use elitism approach.

NSGA-II and SPEA2 [18] are very similar in the way they use the elitism. SPEA2 uses a fixed population size and an external archive (A) that also has a fixed size (N is the archive size), which is normally set equal to the population size (NP). This archive is created to store the nondominated solutions. It is combined with the current population to form the next archive that is then used to create the offspring for the next generation. When filling solutions in the archive, two special scenarios can occur (line 5, Algorithm 12): (i) the number of nondominated solution is smaller than the archive size or (ii) the number of nondominated solutions exceeds the archive size.

In the first situation, the archive (A_{t+1}) includes the nondominated solutions of P_t and then it is filled up by the remainder dominated solutions of the population (line 9, Algorithm 12). This selection is performed according to a fitness value, which considers for each individual how many individuals it dominates and it is dominated by, plus a density value.

In the second case (line 7, Algorithm 12) a truncation operation is performed. For that operation the k^{th} nearest distance method is used. It means that the solution which has the smallest distance to the other solution is the one that will be removed from the archive. If solutions have the same minimum distance, the second nearest distance will be considered, and so forth.

Next, a number of solutions are selected as parents using binary tournament selection with replacement (line 11, Algorithm 12). Finally, recombination and mutation operations (line 12, Algorithm 12) are applied to generate a number of offspring solutions, which become a new population that must be evaluated and merged into the archive. More details on SPEA2 can be found in the original publication [152].

Algorithm 12 Pseudocode for SPEA2 algorithm

```

1:  $t \leftarrow 0$ 
2: Generate initial population  $P_t$  and empty archive  $A_0$ 
3: while  $t < T$  do
4:   Calculate fitness value of all individuals in  $P_t$  and  $A_t$ 
5:   Set  $A_{t+1} \leftarrow$  nondominated individuals in  $P_t$  and  $A_t$ 
6:   if  $A_{t+1} > N$  then
7:     Reduce  $A_{t+1}$ 
8:   else
9:     Fill  $A_{t+1}$ 
10:  end if
11:  Perform binary tournament selection with replacement on  $A_{t+1}$  to fill
    mating pool
12:  Apply recombination and mutation operators to the mating pool
13:  Set  $P_{t+1}$  to the resulting population
14:   $t \leftarrow t + 1$ 
15: end while

```

5.10 Multiobjective Metaheuristics Classification

In Section 4.1 relevant concepts about metaheuristics were presented. Indeed, there are several ways of classifying and describing metaheuristic algorithms. Therefore, with multiobjective metaheuristics we have adopted the same classification to organize the different approaches: (1) trajectory-based and (2) population-based metaheuristics. .

The main difference between the two classifications resides in the number of solutions used at the same time: Does the algorithm work on a population or on a single solution at any time? Algorithms that work on single solutions are known as trajectory methods and encompass local search-based metaheuristics. They all share the property of describing a trajectory in the search space during the

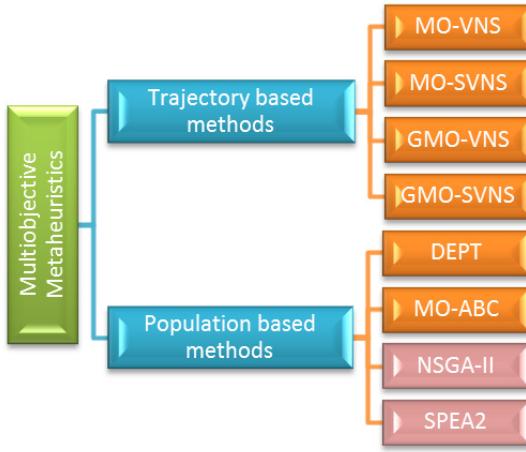


Figure 5.2: Classification of our multiobjective metaheuristics under trajectory-based metaheuristics and population-based metaheuristics.

search process. Population-based metaheuristics, on the contrary, perform search processes which describe the evolution of a set of points in the search space.

The metaheuristics considered in the multiobjective research are dominance-based approaches. They are characterized by the use of the concept of dominance in the fitness assignment, contrary to the other approaches that use a scalarization function or treat the various objectives separately. This idea was introduced initially into GAs by Goldberg [55].

In Figure 5.2 we tried to classify the multiobjective metaheuristics presented in this section considering the above characteristics, as shown in the diagram.

5.11 Summary

In this chapter, we have presented various new multiobjective algorithms that we have developed under some previously studied algorithms. The former single-objective algorithms were used as reference to design the new multiobjective approaches. They were modeled considering the multiobjective formulation of FAP. A detailed explanation about each algorithm is made, focusing on some of their basic concepts and given as an example the specific scenario of applying them to address FAP. Indeed, we have outlined the most relevant aspects that need to be considered when solving the multiobjective formulation of FAP.

Under our point of view, these new approaches can indeed be generalized to be used in other multiobjective optimization problems. Therefore, some of the main concepts used to evaluate the quality of the solutions are set apart in the beginning of the chapter. As a matter of fact, to generalize the proposed

algorithm to other problem domains, the main difference resides in the method used to compute the several goals.

Finally, since with our proposed algorithms we are going to conduct several experiment scenarios to assess their efficiency to solve FAP, we have also compared the metaheuristics with some of the most popular and well-known algorithms in multiobjective optimization, named NSGA-II and SPEA2. In the next chapter experimental results are presented to every one of the presented metaheuristics.

Experimental Results

In this chapter, we describe the experiments designed to assess the efficiency of every algorithm in tackling the real-world instances of FAP that we used. Thus, we present the results obtained during the tuning phase and also the final results observed during the test phase. In the experiments scenario we used the two formulations of FAP followed by us: the single-objective version and a multiobjective version.

Results emphasize the most important characteristics of every class of algorithm. Indeed, we intent to compare the population-based algorithms against the trajectory-based algorithms. Major differences between behaviors of the several implementations are presented and the most efficient ones are pointed out.

This chapter is organized as follows. In the beginning we describe the real-world instances of FAP used. Then, the configuration engaged to conduct the experiments scenario is also shown. During experiments the same type of steps are always followed. Prior to describing the accomplished results and since our aim is to assess the efficiency of every metaheuristics, an introduction to the base ideas is made and the metrics for the performance indicators followed to analyze the accomplished results by the experiments. Next we show the results obtained in the tuning process of every algorithm, that is then followed by the algorithms final results. A final comparison is also made against the other considered algorithms.

Regarding the experiments scenario itself, the main activity involves the data collected directly from algorithms executions that then is assessed by using the performance indicators. The indicators vary depending on the type of problem being analyzed (the single-objective algorithms or the multiobjective algorithms).

To provide data with statistical confidence and to detect differences of performance between algorithms within short and long time ranges, we have gathered statistical data at every 2 minutes of execution.

6.1 Real-world Problem Instances

In this section we outline the used GSM instances and their main characteristics regarding FAP. First, we start by presenting the technical features of the large-scale real-world GSM instances and then some examples of the configuration files used to represent the network are given. In the end we make some considerations regarding the definition and planning required by configuration files used in the study.

6.1.1 Characteristics of the GSM Instances

Two distinct mobile networks were used to perform the experiments to assess the efficiency of every considered algorithm. These mobile networks represent two large-scale real-world instances of FAP currently operating, which are two different US cities named Seattle and Denver. The instances have different sizes, being the Denver instance the one with the highest area, becoming therefore the most complex to be tackled.

A mobile network has several components 2.2, but we focus on the most important ones to formulate our approach of FAP. Therefore, regarding their configuration, the Denver instance has 2612 TRXs, installed in 711 sectors that are distributed in 334 BTSs. Every TRX has only 18 valid frequencies to be assigned to it (from 134 to 151).

The Seattle instance is a smaller instance and has 970 TRXs, installed in 1180 sectors, which are distributed in 503 BTSs. In this network each TRX has 15 valid frequencies to be assigned to it. In this instance the range of valid frequencies may differ from TRX. As an example, in some TRX the range of valid frequencies varys from 591 to 605, and in others it is from 588 to 602. This information is unpredictable and it is obtained from the instance configuration files.

It is also remarkable that the considered instances are currently operating and therefore they represent a real scenario in the FAP. Figures 6.1 and 6.2 display the network topology for every instance in which every triangle represents a sectorized antenna were the TRXs operate.

Although the instances have small different characteristics, in our algorithms implementations we assume that the range of valid frequencies may vary from TRX and therefore, the complete data of every TRX is loaded into memory from the configuration files of the mobile network.

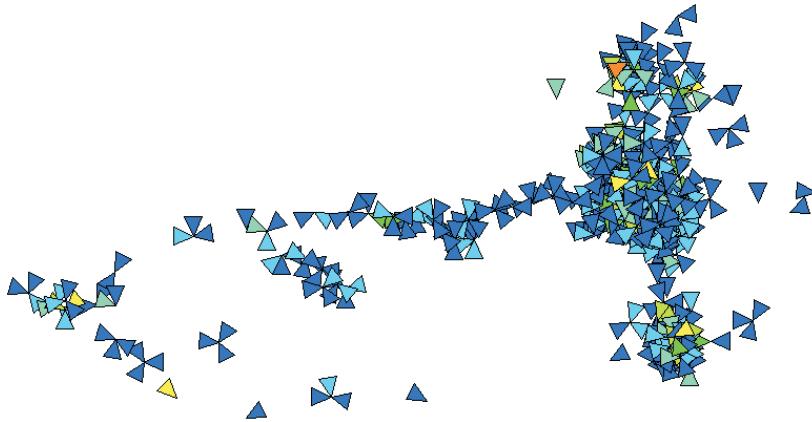


Figure 6.1: Topology of the GSM Denver instance.

To evaluate the quality plans accomplished by the metaheuristics, information to compute the level of interferences needs to be processed. In both instances, the level of interferences denoted by the interference matrix is computed using a $K=100\,000$, $C_{SH}=6$ dB and $C_{ACR}=18$ dB (used by Equation 2.3). The C_{SH} is the minimum quality signaling threshold, that is, if the carrier-to-interference ratio is lower than this threshold there will be a degradation of the communication quality. On the other hand, C_{ACR} is a hardware specific constant that measures the ability of the receiver to receive the wanted signal in the presence of an unwanted signal as an adjacent channel. For more detailed explanation about the mathematical formulation, please refer to [92].

The interference matrix is the same used by [92, 93]. The data source to build the interference matrix is based on the CI (Carrier-to-Interference ratio) probability distribution and uses thousands of Mobile Measurements Reports (MMRs) [84] rather than propagation prediction models. MMRs are a more accurate data source, as they capture the call location pattern in the network and do not rely on predictions. These properties make the GSM problem here addressed a more realistic one than standard benchmarks [39].

Indeed, the most similar approaches are the COST 259 benchmark, but it still does not represent all the realistic concepts that are incorporated by the instances that we use for FAP. Besides the COST 259 benchmark, the first well-known benchmarks were the Philadelphia, CELAR and GRAPH instances [39] but they represent even a more minimalist approach. All these benchmarks are indeed simpler than the realistic instances that we have used.

These instances were provided by our industrial partners and are also used by other research groups. Therefore, some results for these two instances have already been reported in the literature, but only for the single-objective version

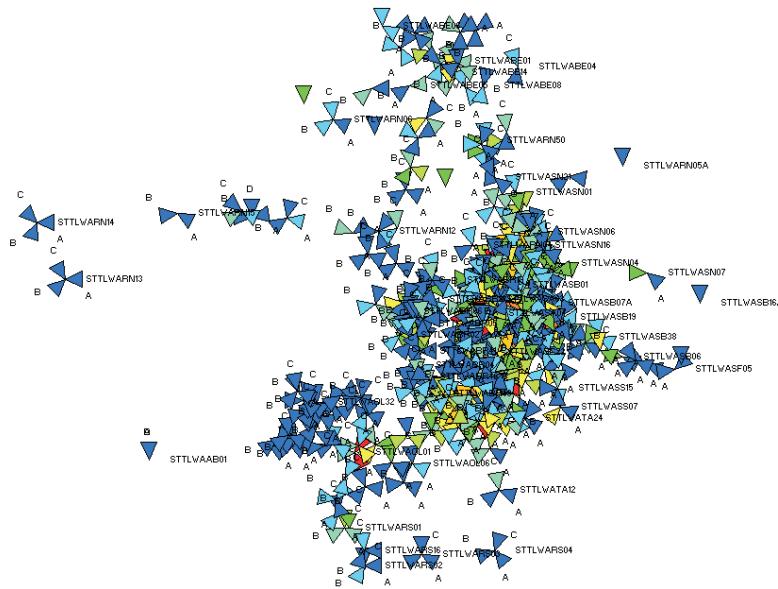


Figure 6.2: Topology of the GSM Seattle instance.

of FAP. To the best of our knowledge we are the ones starting to address the current formulation of FAP for GSM networks in a multiobjective perspective.

6.1.2 Instances Configuration Files

As previously mentioned, the experiments were performed over two large-scale real-world instances of GSM networks, named Denver and Seattle instance. The technical characteristics of every instance are obtained from distinct configuration files. Indeed, to every instance near 8 distinct files are required, each one containing the data that specifies the technical configurations presented in the mobile network instance.

Next we present as an example the structure of the data that algorithms need to process. Therefore, we show some of the content of the most important data from several configuration files. For both instances the data structure followed by the several files is similar, and it is available in one of the most common file formats, which is the text file (.txt file).

<instance>.trx This file contains all the information about the existing transceivers in the network. There are two types of transceivers: one is BCCH and the other is TCH. A transceiver makes part of a sector. Therefore, a sector can have several TRXs. The number of available transceiver in a sector can be configured, but usually each one can have at most 8 TRXs. One is of the type BCCH and the other seven are TCH. These files have information about the type of TRX, its

sector and the site in which it is, respectively. It also contains the number of available frequencies for that TRX and their range of value (see Figure 6.3).

TRX ID, Type, Sector, Site, Number of Valid Frequencies, List of Valid Frequencies									
0	BCCH	0	1	18	134	135	136	137 ...	147
1	TCH	0	1	18	134	135	136	137 ...	147
2	TCH	0	1	18	134	135	136	137 ...	147
3	TCH	0	1	18	134	135	136	137 ...	147
4	BCCH	1	1	18	134	135	136	137 ...	147
...									

Transceiver unique identifier

Transceiver type: BCCH or TCH

Sector and Site IDs Where it's located

Valid frequencies for the TRX.
"first column is the number of valid frequencies"

Figure 6.3: Configuration file containing the data about the TRX.

<instance>.sector The information available in this file is associated to the existing sectors in the network. A sector can have a set of different TRX. This file contains information about the number of the several sectors, the value to be used in its channel separation matrix, the number of existing TRX in that sector and also the unique identification (number) of each TRX (see Figure 6.4).

Sector ID, Channel separation constraint, Number of TRXs, List of TRX IDs									
0	3	4	0	1	2	3			
1	3	4	4	5	6	7			
2	3	4	8	9	10	11			
3	3	4	12	13	14	15			
...									

Sector unique identifier

Constraint separation at sector level

TRX IDs within the sector

Figure 6.4: Configuration file with the sectors data.

<instance>.site It contains all the available sites in the network. In this instance, a site can be compared to a base station. It contains information regarding the identification of the available sites, its channel separation constraint value, the number of sector present in that site and their respective identification (number of the sector). As an example, the data is visible in Figure 6.5.

<instance>.IM contains the values for the several interferences caused by the victim and the interferer sectors. The interference value is given by the means and is a standard derivation of the probability distribution of C/I radio from "interfering sector" in the service area of the "victim sector": the greater the mean, the lower the potential interference (see Figure 6.6).

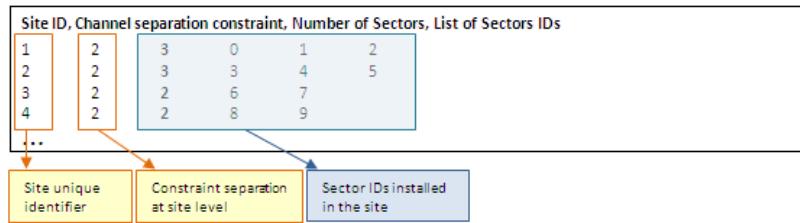


Figure 6.5: Configuration file with the data about the sites.

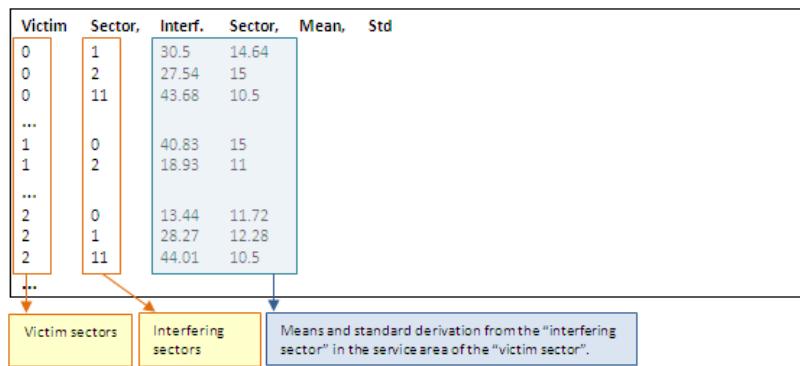


Figure 6.6: Configuration file with the IM data.

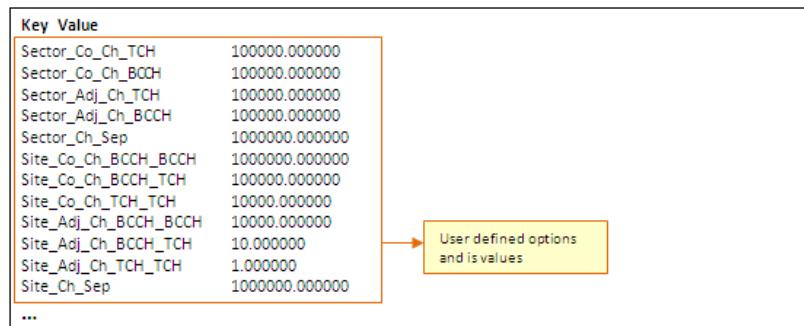


Figure 6.7: Configuration file containing data related to the network options.

<instance>.opt The available information in these files correspond to the user defined options and their values. This information is used to perform the calculation of the fitness value from an individual. It represents the penalty values to be used when a constraint is not satisfied provoking the occurrence of interferences (see Figure 6.7).

Sector ID	ID, Number of Neighbors	Sectors, List of IDs
0	6 318 332	300 295 1 2
1	3 300 2	0
2	7 332 298	284 19 11 1 0
3	8 419 390	364 359 357 352 5 4
4	10 419 398	390 364 359 358 357 352 5 3
...		

↓

Sector ID

Set of neighboring sectors
(first order neighbors)

Figure 6.8: Configuration file about the 1-order neighbors in the network instance.

Sector ID	Number of Sector Neighbors	List of Neighbors
0	40 1 2 10 11 12 19 54 55 56 ...	326 327 332 333 428 598
1	15 0 2 11 19 54 55 56 284 295 298 300 301 302 318 332	
2	36 0 1 10 11 12 17 18 54 55 56 135 ...	332 333 422 425 428 598
...		

↓

Sector ID

Set of second neighboring
for each sector

Figure 6.9: Configuration file about the 2-order neighbors in the network instance.

<instance>.1-hop.neighbors For each sector it contains the number of its first order neighbors sectors (the nearest) and their correspondent unique identification - *id number* (see Figure 6.8).

<instance>.2-hop.neighbors it has the same structured as the previous file, therefore, to each sector it contains the number of its second order neighbors sectors and their correspondent identification - *id number* (see Figure 6.9).

The data contained in all the above files is related to one another, becoming essential to understand their meaning and how they relate to each other. One of our tasks was the interpretation of the data files and their structure, for us to be able to comprehend and manipulate the data of the instances.

The complexity of the available information in the configuration files makes the process of applying an algorithm to solve it very complex. It is necessary to plan a way to represent the information in adequate data structures, and also to project the best way to represent a solution to the problem. In our approach, all the information available in the files is read into memory to be then processed by the considered algorithm.

Since it is necessary to evaluate the quality of a solution, it is important to define the fitness function to be used to evaluate the real-world instance. The required information to perform this task also needs to access and consider all the constraints defined in the several configuration files. The data source is the same

but the way fitness function is computed depends on the class of algorithms being addressed. In single-objective algorithms we use the interference cost value, but in multiobjective algorithms we use the interference cost and the separation cost value.

6.2 Codification of the Problem

In the evolutionary approach that we have developed to address FAP, we adopted an implementation model to reflect some of the technical characteristics of the mobile networks. This approach intents to clearly identify the limitation of the network and the restrictions that need to be satisfied when generating candidate solutions to the problem being tackled.

Since that in FAP a solution represents an assignment of a frequency value to every TRX on the mobile network, an individual has different lengths depending on the size of the instances being used. This happens because every instance has a different number of TRXs. In evolutionary computing, an individual represents a candidate solution. Indeed, a solution is known as a chromosome that is composed of genes. In the specific case of FAP, each gene is located at a particular position on the chromosome and therefore it will represent a TRX of the mobile network. At technical level, a TRX also has information about the sector in which it is installed inside the mobile network.

The intrinsic association made between the TRX and the sector in which it is installed is the main concept to define a frequency plan. This information, along with the range of valid frequencies for every TRXs, represents the data mostly used by algorithms in the optimization of a frequency plan.

A more detailed explanation about the approaches that we have followed in the development process that permitted us to implement the evolutionary algorithms are given next.

6.2.1 Individual Representation

As previously mentioned a solution considering our FAP instance is coded as a vector of integers. This vector represents a frequency plan and as an example, Figure 6.10 shows their representation for the Denver instance. Thus, each gene corresponds to the frequency value assigned to a TRX and each position of the vector represents a TRX of the network. Therefore, to every position of the vector, a specific value needs to be assigned that is in a very limited frequency range of values.

In our experiments scenario, the range of valid frequencies for every TRX can be different. Therefore, for every TRX it is kept a set of valid frequencies and

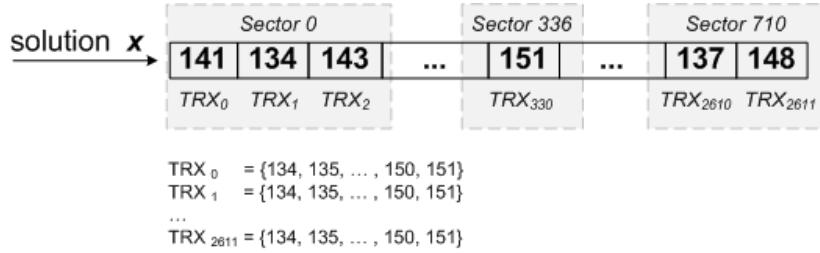


Figure 6.10: Solution encoding.

other containing only the available ones that are the ones that have not been assigned yet.

The range of valid frequencies for every TRX is obtained from the configuration files of the instance. Every TRX contains a set of valid frequencies and since every TRX is installed inside a sector, to every sector it is also keep a list of all the frequencies that are available at a specific moment. The set of the available frequencies is maintained considering all the TRXs in the same sector. All this information is kept in memory to simplify the evolutionary operations required by the algorithms.

In terms of memory consumption, storing all this information becomes an important overhead. Therefore, in the software that we developed to implement the several considered metaheuristics, the performance was always one of the key factors considered during the implementation phase. The data structures selected to store the data read from the mobile network configuration files and the data required to implement the metaheuristics were selected from the most recent data structure available in the C# language under .NET framework [108]. The data structures were also selected considering the ones that had a better performance in terms of efficiency and memory consumption. We have implemented all the software using Object-Oriented Programming (OOP) software architecture. Therefore all the modeling and implementation phase was done using classes. Our approach tried to represent the characteristics being modeled by the problem in every class identified to the problem.

In more detail, regarding the codifications of the problem, the main information is coded in two base classes that represent a solution of the problem: the Individual class and the TRX class. These two main classes structures are visible in Figure 6.11. In the TRX class it is stored all the information required to characterize a TRX inside the mobile network architecture. The information is related to the set of valid frequencies, identification of the sector in which it is installed, the site, etc. The Individual class was set to store all the information that is necessary to compute the quality of a frequency plan, that is characterized by its

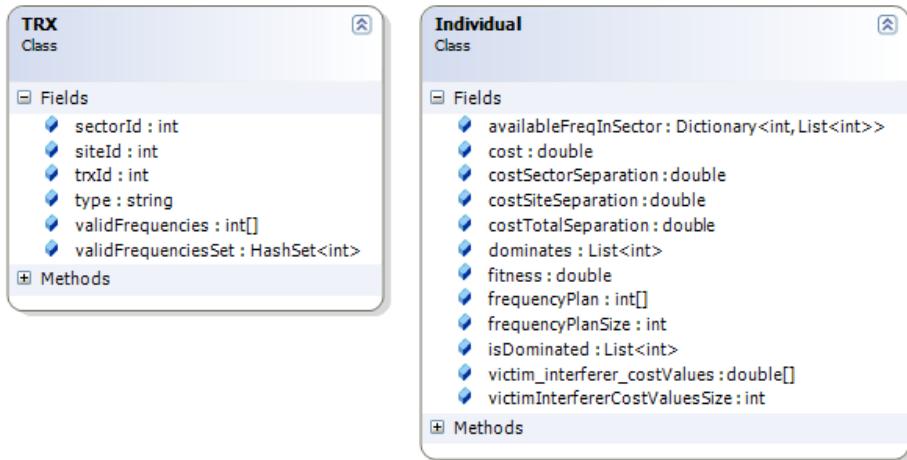


Figure 6.11: Base classes definition to address FAP.

several objective values*. Figure 6.11 shows part of the data defined in two of the most important classes that were modeled to define the physical characteristics of FAP. These two classes are used in together with others that were implemented to define an instance of the problem. They were used with the remaining classes that independently characterize every metaheuristic that we have considered to be implemented for FAP. In Figure 6.11, the attributes that are visible for every class represent the multiobjective formulation of FAP. Additional notes regarding the implementation process are also given in subsection 6.2.3.

Sometimes in the software design it has become increasingly complex to manipulate all the distinct data required by FAP. Therefore manageability has become an important concern. Thus, in the design of the software we have developed several packages to group classes with similar characteristics. In practice, all the base classes representing the characteristics of the problem were grouped inside the package named "Problem". The remaining classes specific for every metaheuristics were put inside a different package. Using these design concepts in the software development permitted us a simpler and more organized way to implement future improvements. The outcome was the existence of a more generic platform that is available and that permits an easier way to implement future metaheuristics.

6.2.2 Creating the Initial Solution

Since we are using evolutionary computing techniques and considering that their main characteristic is the intensive use of randomness and genetic-inspired oper-

*Some additional information about other approaches followed during implementation process are available in Appendix A

ations to evolve a set of candidate solutions, several approaches can be followed in the creation of the initial solution(s).

The most common way to create an initial solution is to assign a random value to every gene of the chromosome. Sometimes there may be necessary to follow some constraints. In our case, creating the initial solutions consists in assigning a random frequency value f to every TRX in the mobile network (a TRX is represented by an index in the vector). Considering our structure of classes, every position of the vector identifies an instance of a TRX class.

Due to the high level of randomness associated with this obvious and simplistic approach, we tried to incorporate a more efficient approach in the process of creating a new solution. Thus, when generating a frequency plan (a solution), an improved method was developed that considers some level of technical information related to the constraints of the mobile network. The main feature of this method is that it tries to avoid the cost with the most severe penalty (the cost K). However, besides this restriction in the assignment of a frequency to a TRX, it still uses a random approach when selecting a frequency value from the range of unused frequencies.

As a result, to guarantee that the mobile network constraints are being followed, the approach used to create the solutions was the same used by all the metaheuristics that we have implemented.

More details about the constraints that need to be considered when generating a frequency plan in a mobile network were given in Section 2.2 and Section 4.2. Indeed, the incorporation of problem domain information was one of the key features developed in the hybrid approaches that we have developed.

6.2.3 Considerations Regarding the Software that was Developed

In the metaheuristics implementation to facilitate future changes in the experiments scenario, we have considered the benefits of using a more flexible and at the same time easy and comprehensible approach to define the parameters necessary for every metaheuristic. Almost all the metaheuristics need different parameters values. Indeed, in the experiments phase, besides the parameters tests, it is necessary to perform several runs to grant statistical relevance to the accomplished results. Therefore, the high number of different experiments required for every parameter being tuned and also because of the final set of tests performed with the final configuration of every metaheuristics, it made us project and develop an approach mutable enough to facilitate the several set of tests that needed to be performed. Thus, because of all this high range of different characteristics, a XML file is used to define the sequence of tests to be performed for every set of parameters of a metaheuristic.

```
<?xml version="1.0" encoding="UTF-8"?>
<runs>
  <parameters>
    <n>125</n>
    <seed>5</seed>
    <numberOfRuns>30</numberOfRuns>
    <pc>0.009</pc>
    <pm>0.001</pm>
    <crossoverType>8X</crossoverType>
    <type>spea2</type>
  </parameters>
</runs>
```

(a) XML file with a single configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<runs>
  <parameters>
    <F>0.8</F>
    <CR>0.9</CR>
    <popSize>10</popSize>
    <seed>5</seed>
    <strategy>Rand2Bin</strategy>
    <numberOfRuns>10</numberOfRuns>
    <localSearch>false</localSearch>
  </parameters>
  <parameters>
    <F>0.8</F>
    <CR>0.9</CR>
    <popSize>25</popSize>
    <seed>5</seed>
    <strategy>Rand2Bin</strategy>
    <numberOfRuns>10</numberOfRuns>
    <localSearch>false</localSearch>
  </parameters>
  ...
</runs>
```

(b) XML file with multiple configurations.

Figure 6.12: XML configuration files example for the developed algorithms applications.

As an example, the configuration file required by our software is presented next. The example represents the two types of configuration files that can be executed. In this case, there are the final DEPT runs: the first file defines the parameters for a single execution, and the second file contains the information regarding several sequential runs to be executed by DEPT algorithm.

The most important advantage is that using a XML file the values and XML elements included in the configuration file can be easily changed depending on the metaheuristic it is representing. The main function of the file is to define the sequence of tests to be performed and for every sequence the several parameter values being used. For these reasons, by adopting this kind of file, we are permitting an easy and understandable interpretation for the sequence of tests required to analyze the performance of every metaheuristics. Thus, it facilitates all the

enormous amount of tests that are necessary to perform during the exhaustive experiments scenario.

At every execution, statistical output files are generated to a specific output directory. These output files contain the frequency plans obtained with the several distinct parameters tested in every experiment and also the statistical data gathered during execution time.

An additional tool was developed to analyze the accomplished results obtained with the multiobjective metaheuristics. We have developed this software mainly because multiobjective metaheuristics have very specific ways to compare their performance regarding the several accomplished final Pareto fronts.

Indeed, in multiobjective algorithms it is necessary and useful to use additional indicators to evaluate the efficiency and performance of the accomplished solutions. Therefore, an application that compares distinct Pareto sets and analyses the statistical data using the selected indicators was implemented. This tool also includes a graphical representation for the Pareto Fronts. Appendix A contains more information about the tool.

6.3 Methodology and Metrics

In this section, we explore the distinct methodologies used to evaluate the quality of the results obtained by the metaheuristics. A description of the main features followed in the tests phase is presented.

6.3.1 Experimental Conditions

Regarding the computational facilities we used a similar scenario for the implementation of the metaheuristics. The code was developed under the .NET framework 3.5 with C# language through object-oriented programming (OOP) approach.

Experiments were performed in different periods of time; therefore it was necessary to use different computational characteristics. The first experiments, that report results for the single-objective version of FAP, were carried out under a 3.2GHz Pentium 4 processor with a 2GB RAM. With this type of algorithms the experiments were carried out under identical conditions. Regarding the multi-objective metaheuristics, the experiments were performed under PC running a Windows XP operating system with 2.66Ghz Intel Core 2 Duo Pentium processor and 3GB RAM. Experiments with this type of metaheuristics were also being carried out under exactly the same conditions. Compared to multiobjective experiments conditions, with the single-objective experiments we used a PC with less RAM and an inferior CPU.

Regarding the termination conditions and for the process monitoring of the experiments scenario we considered executions of 30 minutes. For every 2 minutes we saved the minimum, average and standard deviations values for the objective values being considered. Additionally, when performing the multiobjective experiments we have also saved to the output files the Pareto fronts, containing all the solutions belonging to it and its correspondent hypervolume.

The confidence building in the experiments is a key factor to assess the performance of the reported results; therefore in order to provide results with statistical confidence, 30 independent runs were implemented for each experiment.

Although, nowadays we have two real-world instances of GSM networks available, initially when we were focused in the single-objective version of FAP, only the Denver instance was available. That is the reason why the results with the single-version of FAP are analyzed only using the Denver instance. Then, to increase the scenario of the experiments, in the multiobjective metaheuristics we have added a new instance to confirm the effectiveness of the proposed approaches. The obtained results reflect the distinct characteristics of the different scenarios. Indeed, the magnitude of values obtained for every instances is quite different, and for that reason the comparatives among algorithms performance take this fact into consideration.

The statistical indicators, as the minimum, maximum, average and standard deviations values were considered at every 2 minutes of execution (for the total of 30 minutes). In the single-objective this statistical data was applied only to the interference cost since it was the only objective being minimized. Regarding the multiobjective algorithms the statistical information was gathered for the interference cost, the separation cost and the final fitness value. Additionally, the performance indicators for the multiobjective results were also enhancement with the use of the hypervolume indicator and the coverage relations value. Both these indicators were applied to the Pareto fronts gathered by the metaheuristics.

6.3.2 Experiments on Tune Metaheuristics Parameters

Evolutionary metaheuristics use several parameters to perform some of common features of that type of metaheuristics. Operations such as the selection method, reproduction, replacement strategies are normally defined by a parameter value. Thus, in all the implemented metaheuristics reported in this study, experiments for every parameter were carried out. This represents another contribution of our work, the correct configuration of all metaheuristics for the specific FAP scenario.

Testing all possible combinations of parameters (with all its possible values) would lead to an excessive number of experiments (almost intractable) and for this reason it was decided to follow the tuning strategy that considers one parameter at a time. The order chosen to evaluate the parameters always tries to first

evaluate the parameters that are considered to be more significant in changing the results.

6.3.3 Multiobjective Performance Indicators

Evaluating the performance of multiobjective algorithms is far from being a trivial task since there are multiple solutions instead of a single good solution. Therefore, a compromise between conflicting objectives is required. To evaluate the degree of quality achieved by the outcomes obtained by the multiobjective algorithms, we incorporate in the analysis two additional quality indicators: hypervolume indicator [42] that defines the volume of the objective space dominated by the Pareto front, and the coverage relation [150] to determine the best Pareto front. Both are applied to all non-dominated solutions as comparison criterion for assessing Pareto optimal solutions. Both metrics give a percentage value and a highest value means that it represents a more desirable frequency plan in the FAP context.

As mentioned in section 6.2.3, in order to apply the performance metrics used to evaluate the several sets of Pareto fronts we have decided to automate the analysis of the results. For that reason, we have developed an application specially for the analysis of Pareto fronts. The software receives as input the data about the Pareto fronts that we need to analyze and compare. It then computes the final dominance against the several solutions and it outputs the final non-dominant solutions and the relative hypervolume indicator value. Besides the hypervolume value, it also computes the coverage relation value. Regarding this indicator, the application receives in distinct files the two sets of Pareto fronts to which the coverage relation value is computed. It outputs the coverage relation values for every Pareto front, identifying in the output file what the solutions dominated by every solutions are. All these data are read and stored in text files.

Hypervolume Indicator

The hypervolume indicator (HV) [42, 150] is truly useful because it rewards convergence towards the Pareto front. It defines the measure of the region simultaneously dominated by every Pareto front point and bounded above by a reference point. In our case, the hypervolume represents the two-dimensional areas dominated by the Pareto front. The Pareto front with the highest hypervolume value is a better configuration plan for FAP.

In the hypervolume it is necessary to define a search space by specifying upper bound and lower bound. In our practice, the upper bound is the reference point. The upper bound was (300 000, 2 500) for the Denver instance and (70 000, 200) for the Seattle instance. The lower bound points were (130 000, 1 200) and (4 000, 15) respectively.

The above limits were defined after analyzing the boundary values that were being accomplished to every objective function. Therefore, all the algorithms have used the same boundary points which allowed a more accurate result comparison around the distinct results accomplished by the metaheuristics.

Regarding the hypervolume indicator, some value considerations are justified:

- The area (as we have only 2 dimensions, it is not really a volume, but an area) of a rectangle and it is simply the multiplication of its base by its height. If $P_1(X_1, Y_1)$ is its bottom-left corner and $P_2(X_2, Y_2)$ is the top-right. The volume is calculated as $V = (X_2 - X_1) * (Y_2 - Y_1)$.
- Moreover, to avoid an objective function to have more weight than the other. Before calculating the hypervolume, the axes are normalized in the range [0,1]. At the time of normalization, we use the larger value for both objective functions (300 000, 2 500) (our reference point). For the smaller value (lower-bound) we use the (130 000, 1 200). This is the scenario considering the Denver instance. In Seattle the process is similar, but using different boundary values.

The procedure followed to calculate the hypervolume for a particular Pareto front set it done as follows:

1. Sort the Pareto front points from high to low Y (Y axis). The Y axis represents the separation cost value. The X axis is the interference cost value.
2. Starting in the higher Y value, through each point in that order, calculate the volume (actually the area) between each point and the *reference point*. After, this area (volume) is accumulated in an accumulator that stores the total cumulative area.
3. For example, in Figure 6.13 we assume that the points are $P_1(X_1, Y_1)$, $P_2(X_2, Y_2)$, $P_3(X_3, Y_3)$ and $P_{ref}(X_{ref}, Y_{ref})$. That is, we have a Pareto front with 3 points, besides the *reference point* P_{ref} . The hypervolume is computed as:

$$HV = (X_{ref} - X_1) * (Y_{ref} - Y_1) + (X_{ref} - X_2) * (Y_1 - Y_2) + (X_{ref} - X_3) * (Y_2 - Y_3) \quad (6.1)$$

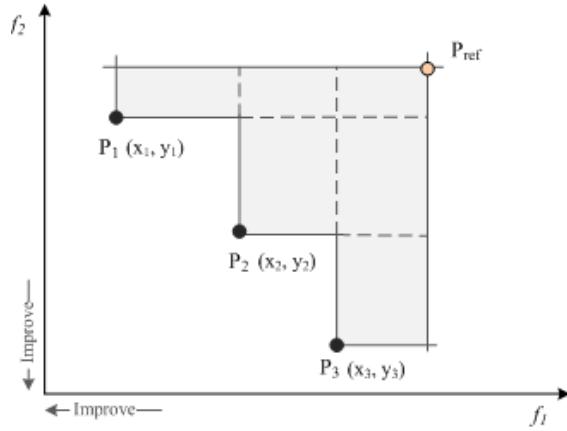


Figure 6.13: The hypervolume representation considering two-objective functions.

Coverage Relation Indicator

Coverage relation (C) [150] is used as an additional performance indicator to assess the performance of multiobjective metaheuristics. In terms of dominance, solution x_1 covers solution x_2 , if x_1 dominates or ties with x_2 . To illustrate the process, given two sets of non-dominated solutions, we compute the fraction of a set of solutions covered by the other set of solutions. Formally, to compare two Pareto sets using the coverage relation ($P \succeq Q$) we compute:

$$C(P, Q) = \frac{|q \in Q; \exists p \in P : p \prec q|}{|Q|} \quad (6.2)$$

where, $p \prec q$ indicates that solution p dominates solution q , belonging p to the Pareto set P and q to the Pareto set Q .

Hence, a higher percentage of coverage means a better solution set. When comparing the C of two Pareto sets $C(P, Q)$ it is required to consider the two distinct situations: $C(P, Q)$ and in the inverse case, where $C(Q, P)$ since it is not necessarily the same. When $C(P, Q) = 1$ means that all the solutions in Q are dominated by the solutions in P , in the opposite side, when $C(P, Q) = 0$ means that none of the solutions in Q are covered by the solutions in the P set.

6.4 Single-objective Metaheuristics Empirical Results

In this section, we present all the results obtained with the metaheuristics that we have considered to address the single-objective version of FAP. We start by presenting the distinct results of every parameter studied during the tuning phase of the algorithms. Then, we present the final results of the several algorithms. We also make additional comparatives between our approaches and other implementations developed by other investigators for the same instance of the problem that we have used. Results presented in this section for the considered single-objective metaheuristics are used to evaluate their effectiveness in solving FAP the interference cost value.

6.4.1 Configuration Parameters: Experimental Study

Next we present the results obtained in the experiments that were performed during tuning phase of the algorithms. For every parameter we found the best control parameter value.

Hybrid DE Tuning Parameters

As the classic DE, our hybrid version of DE algorithm involves the tuning of several intrinsic control parameters. To illustrate the difference and implication of every DE parameter, several experiments were performed, each one aiming to find the best parameter value. All the experiments were executed sequentially. It starts from DE/Rand/2/Bin with CR=0.4, F=0.5 and then it results in the tuning phase of a parameter that is going to influence the parameters to be used for the next experiments.

The tuning of DE parameters is performed in the following order:

1. Local search method.
2. Population size.
3. Crossover probability.
4. Mutation operator.
5. Differential evolution strategies.

At this stage of our investigation only the Denver instance was available. Therefore, results discussed with this algorithm are only referring to the Denver instance. For every parameter we have shown the statistical results that were obtained with the 10 independent runs at every 2 minutes of execution. To provide results with statistical confidence, the interpretation and analysis of the

Table 6.1: Best, average fitness and standard deviation of 10 runs of DE with the different LS algorithm approaches (Denver instance).

	LS1			LS2		
	Best	Avg.	Std.	Best	Avg.	Std.
120 seconds	96 348.0	98 523.8	1 271.1	97 569.4	98 818.2	592.9
600 seconds	94 211.1	95 092.0	575.3	94 192.7	94 784.8	582.3
1800 seconds	92 202.7	93 079.9	633.1	91 946.2	92 734.6	571.3

results was performed considering three different time limits, 120, 600 and 1800 seconds.

Applying the Local Search method The hybrid version of DE uses a Local Search (LS) method that was specially designed to address our version of FAP. The LS makes possible to optimize the assignment of frequencies to TRXs in a given sector, without changing the remaining network assignments [93].

The experiments performed with the LS method aim to determine which the best way is to apply the LS method in our hybrid DE version. Two hypotheses were considered. The first one, named LS1, consists in applying the LS to all individuals in the initial population and also after creating the x_{trial} individual. In the second hypothesis, named LS2, the local search is only applied after creating the x_{trial} individual.

Both scenarios were tested under the same conditions. The parameter settings used in this first experiment were NP=10, CR=0.5, F=0.4 and DE scheme DE/Rand/2/Bin. These values were chosen based on some previous tests that we have already performed with DE in the Denver instance.

In the LS experiments, the favorable results were obtained with the second approach (LS2). Indeed, although they have presented similar results, with the second approach the algorithm shows a more consistent performance, having the smallest value for the standard deviation value (see Table 6.1). It is also further noticed that it also has the best cost function value. Therefore, the approach selected to be incorporated in DE was the LS2, which was then followed in the next experiments.

The conducted experiments showed that using a Local Search method within DE algorithm has a very important role in the performance of the algorithm.

Population Size In the DE the population size is represented by the NP value, which is a fixed value through all the search process of the algorithm. This experiment aims to determine which the best NP value to be used in our problem is. From the previous experiment, the LS2 approach was added to the classic DE algorithm which is responsible for applying to the x_{trial} individual a local search

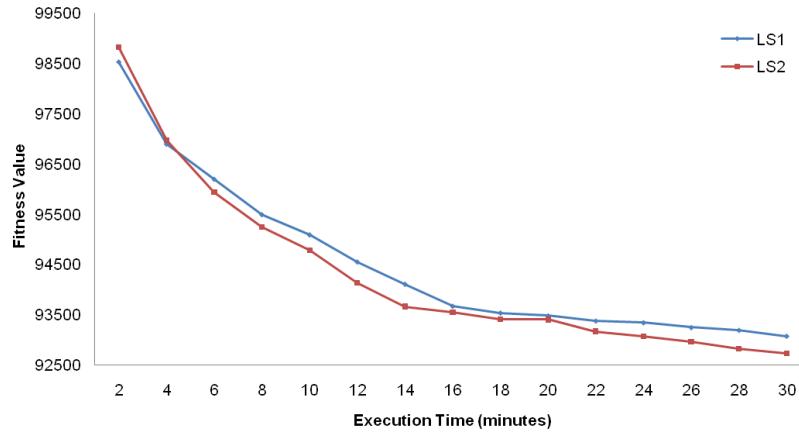


Figure 6.14: Evolution results from 2 to 30 minutes for both LS approaches analyzed obtained among Denver instance (in average values).

method that improves the current solution. The other parameter settings were: CR=0.5, F=0.4, DE scheme=DE/Rand/2/Bin.

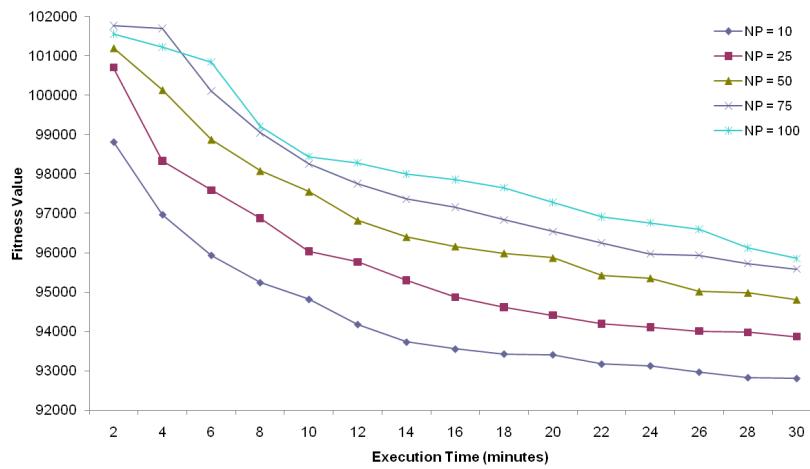


Figure 6.15: Results for the NP values obtained with Denver instance.

Statistical results with the several NP values are tabulated in Table 6.2. The average interference cost obtained with every NP value tested are shown in Figure 6.15. After achieving the results presented in Figure 6.15, the next step consisted in verifying if by using values near 10 (smaller or greater values) it would be possible to obtain better results. This scenario did not occur, the best result was still a population size of 10 individuals. Therefore, this was the value used in the next experiments.

Table 6.2: Results obtained for the different NP values tested. We present the best, the average and the standard deviation fitness of 30 independent runs in three different time limits (Denver instance).

NP value	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
10	97 569.4	98 818.2	592.9	93 896.2	94 819.2	646.9	91 946.2	92 808.8	520.2
25	98 328.5	100 705.2	1 105.4	94 515.6	96 036.8	919.8	93 184.4	93 864.7	421.2
50	99 099.8	101 203.7	1 075.5	96 011.7	97 557.3	824.0	94 303.9	94 808.0	322.0
75	100 927.4	101 773.2	625.4	96 757.1	98 254.5	765.0	93 819.3	95 584.0	821.2
100	100 317.9	101 559.4	807.5	97 074.7	98 444.3	1 017.8	94 519.1	95 854.9	743.8

Crossover Probability Crossover probability is represented by the $CR \in [0, 1]$. In DE this value determines if the newly generated individual (x_{trial}) is to be recombined. To test the CR value we started from a lower limit of 0.1 up to a satisfactory range that appears to be within 0.9-1.0 (1.0 means that all the TRXs will have its assigned frequency changed). The first range of values tested was 0.1, 0.25, 0.5, 0.75 and 0.9. The rest of parameter settings were NP=10, F=0.4, DE scheme=DE/Rand/2/Bin and LS2 approach.

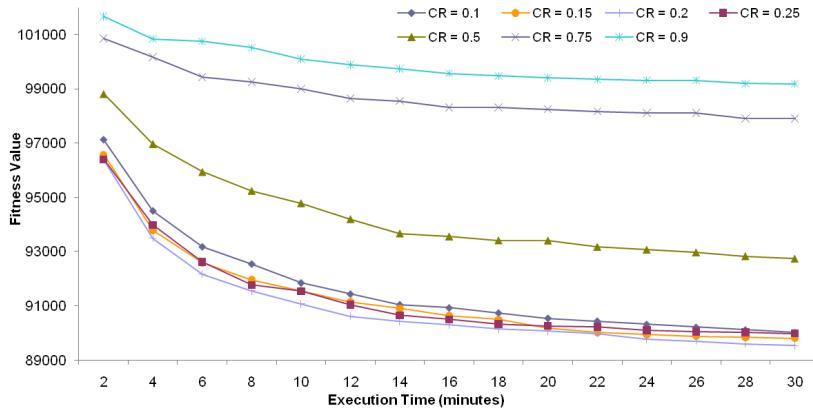


Figure 6.16: Evolution results for the CR values obtained with Denver instance.

With the first range of values, it was possible to determine that the best results were obtained with small values for CR. Therefore, the next step was the decision to verify the evolution results for values near to 0.1 and 0.25. Figure 6.16 only shows part of the tested values, representing the most significant ones. All the results for the several CR values tested are presented in Table 6.3. From these results, we have concluded that the best cost value is obtained using a CR value of 0.2. This means that only a significant small number of frequencies assigned to the TRX will be changed, within the network solution being computed.

Mutation Operator Other intrinsic DE operator is the mutation intensity, which is implemented in DE as a difference vector. The mutation operator is one of

Table 6.3: Results obtained for the different CR values tested. We present the best, the average and the standard deviation fitness of 30 independent runs in three different time limits (Denver instance).

CR value	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
0.1	94 331.3	97 131.6	1 181.0	90 821.7	91 849.6	582.3	89 200.6	90 020.6	563.1
0.15	93 843.8	96 573.5	1 181.9	89 177.8	91 545.2	1 036.6	88 718.0	89 800.6	607.9
0.2	93 251.1	96 390.2	1 193.7	89 827.9	91 068.1	652.1	88 297.4	89 539.3	749.2
0.25	94 767.8	96 398.1	705.4	90 283.8	91 538.7	639.8	89 277.6	89 984.7	527.7
0.5	97 569.4	98 818.2	592.9	94 192.7	94 784.8	582.3	91 946.2	92 734.2	571.3
0.75	98 454.1	100 855.0	1 392.5	96 681.7	99 000.1	1 099.6	96 366.2	97 909.5	845.1
0.9	99 999.6	101 669.0	1 047.5	98 890.5	100 098.0	759.1	98 097.3	99 186.0	598.9

Table 6.4: Results obtained for the different F values tested. We present the best, the average and the standard deviation fitness of 30 independent runs in three different time limits (Denver instance).

F value	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
0.1	95 942.6	97 211.0	948.7	90 010.8	91 418.1	799.0	88 737.4	89 435.6	498.6
0.25	95 973.3	97 106.4	676.5	91 133.8	91 905.6	625.9	89 209.5	89 884.6	500.6
0.5	95 582.4	97 199.2	793.9	90 375.9	91 594.1	737.3	89 174.6	89 848.0	433.6
0.75	95 466.4	96 564.9	860.5	90 182.6	91 827.1	845.3	89 042.5	90 147.4	632.8
0.9	95 139.2	96 649.8	926.6	89 967.5	91 297.9	656.7	87 823.3	89 691.4	730.1
0.05	95 557.2	97 143.5	748.5	90 437.5	91 425.8	572.8	89 117.4	89 847.0	487.9
0.125	95 742.7	96 722.8	705.3	90 049.6	91 671.0	651.8	88 987.7	90 052.5	642.1
0.15	94 015.9	96 647.2	1 299.3	90 162.6	91 315.7	688.2	88 428.4	89 558.8	438.3
0.2	95 455.8	97 034.4	727.5	90 374.8	91 345.5	558.5	88 125.1	89 445.6	713.8
0.85	95 511.0	96 675.8	1 096.3	89 909.2	91 182.5	859.8	88 102.3	89 456.8	979.6
0.95	94 826.0	96 789.8	1 123.9	90 057.7	91 239.7	734.0	89 032.5	89 671.3	449.7

the main characteristics of DE algorithm. Mutation is achieved by adding the weighted difference of two randomly selected vectors to a third vector that will be the new solution (x_{trial}). Normally, the mutation uses a constant scaling factor F in the range $[0, 1]$. Therefore, the range of values was the same as the one tested in the previous experiment. The F values tested were initially the 0.1, 0.25, 0.5, 0.75 and 1. Inspired by the findings, we further investigated some additional values that are near the ones that had the best performance.

The parameter settings followed in these experiments were a NP=10, CR=0.2, DE scheme=DE/Rand/2/Bin and LS2 approach (representing the best of the parameter settings tested so far).

The results for the several F values are visible in Figure 6.17. Considering the average values obtained in the several runs that are tabulated in Table 6.4, the smallest cost value was obtained with an F of 0.1. Therefore, this was the F value to be used in the process of determining which is the best DE scheme.

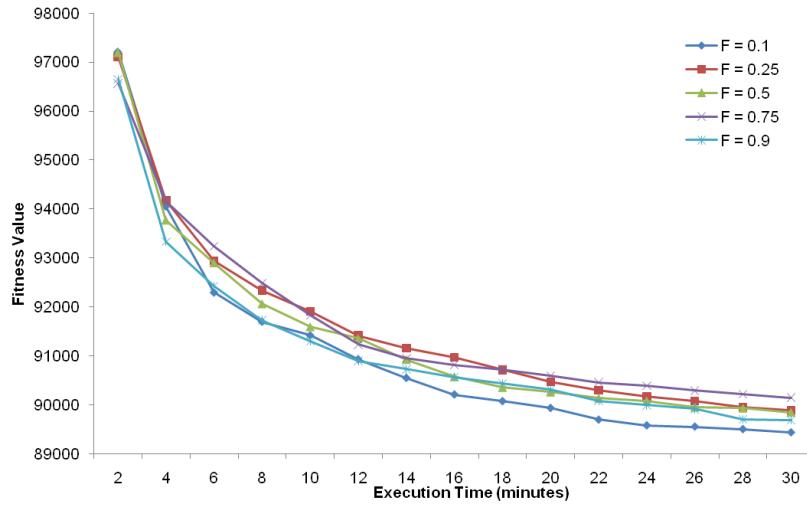


Figure 6.17: Evolution results for the CR values obtained with Denver instance.

DE Schemes Several schemes can be applied by DE in the creation of a new candidate solution. DE schemes differ themselves in terms of base of differential mutation, number of different vectors, and crossover schemes. The most obvious distinction is the fact that they used use different vector pairs to compute the new candidate solution.

In this experiment the different schemes available for DE shown in Table 4.1 were tested. The DE schemes are grouped according to the implemented crossover scheme. Results for the DE schemes using a binomial crossover are shown in Table 6.5 while results applying the exponential crossover scheme are shown in Table 6.6.

Graphically, the schemes with exponential crossover and the binomial crossover are also presented separately to facilitate the evaluation of the accomplished results. All graphs are showing average values that are presented from 2 to 30 minutes.

By analyzing the behavior of the two crossover approaches, it is possible to verify that with all the schemes using a binomial crossover the best results were obtained when compared with its exponential version. The binomial crossover schemes are the most profitable ones. These results are available in the following figures. Graphically, we have compared the strategies that use the best-so-far selection approach against the strategies that used the random approach. These results are shown in Figure 6.18 that shows the best-so-far schemes and the strategies using the random selection are shown in Figure 6.19. Additionally, in Figure 6.20 we compare a more restrict number of strategies, which are the strategies that had the best performance.

By assessing the performance of every strategy, it was possible to determine that a binomial crossover always obtains best solutions. Indeed, with exponential crossovers it was possible to determine, by analysis of the execution output files, that during the execution time almost all strategies have been executed on average during 300 generations. Regarding the binomial crossover, the algorithm runs on average until the 70th generation. Thus, although it uses less generations, it can achieve more favorable results. As a conclusion, schemes incorporating a binomial crossover obtain better results with a clearly smaller amount of generations.

From Figure 6.20 showing the comparison among schemes that have accomplished the best results, it is permitted to conclude that the strategies with the best performance and that outperform all the other approaches were the DE/Rand/2/Bin.

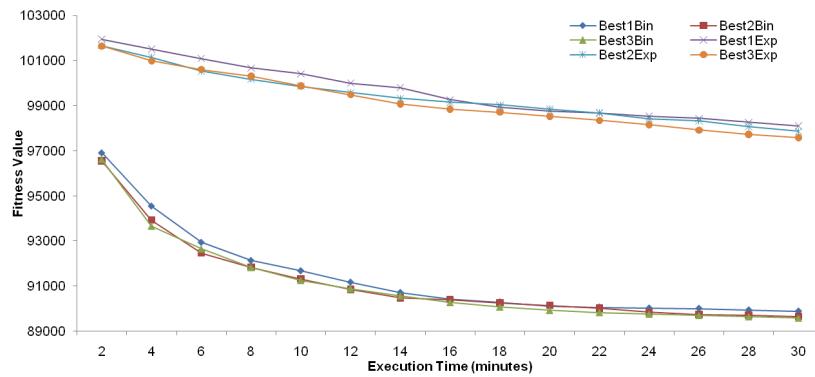


Figure 6.18: Evolution results using the best-so-far selection strategies obtained with Denver instance.

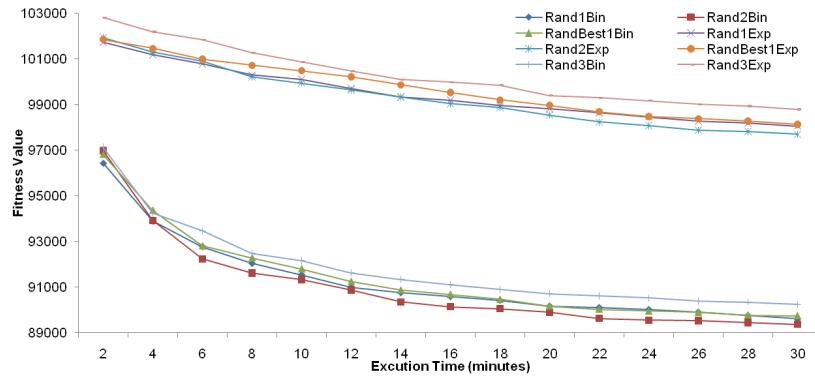


Figure 6.19: Evolution results using the Random selection strategies obtained with Denver instance.

Findings According to the above tuning, the local search method selected was the LS2, that only applies the improving method when a new trial individual is

Table 6.5: Results obtained for strategies using binomial crossover. We present the best, the average and the standard deviation fitness of 30 independent runs in three different time limits (Denver instance).

Binomial strategies	120 seconds			600 seconds			1800 seconds		
	best	avg	std	best	avg	std	best	avg	std
<i>Rand/1/bin</i>	94441.3	96439.8	1060.0	90348.8	91548.5	934.6	88253.2	89613.7	719.6
<i>Rand/2/bin</i>	95942.6	96980.6	715.8	90010.8	91341.1	726.1	88626.1	89379.2	543.1
<i>Best/1/bin</i>	95759.5	96915.0	816.0	90705.9	91672.8	591.4	88685.7	89881.4	575.2
<i>Best/2/bin</i>	94265.3	96547.5	930.5	90503.6	91315.3	498.2	88749.1	89653.9	534.9
<i>Best/3/bin</i>	94832.1	96615.4	883.6	90327.1	91247.9	573.0	88596.9	89578.6	757.2
<i>BestToRand/1/bin</i>	95440.1	96829.2	722.6	90598.2	91804.2	790.3	88885.9	89751.1	642.0
best	94265.3	96439.8	715.8	90010.8	91247.9	498.2	88253.2	89379.2	534.9
avg.	95113.5	96721.2	854.7	90415.7	91488.3	685.6	88632.8	89643.0	628.7
std.	640.5	198.8	120.4	224.2	202.8	147.9	194.0	154.5	85.6

Table 6.6: Results obtained for strategies using exponential crossover. We present the best, the average and the standard deviation fitness of 30 independent runs in three different time limits (Denver instance).

Exponential strategies	120 seconds			600 seconds			1800 seconds		
	best	avg	std	best	avg	std	best	avg	std
<i>Rand/1/exp</i>	100 699.4	101 718.9	827.8	98 700.6	100 115.5	939.8	97 088.3	98 058.0	1 008.7
<i>Rand/2/exp</i>	99 042.6	101 946.2	1 099.3	98 139.1	99 943.2	1 029.0	96 530.9	97 694.2	764.8
<i>Best/1/exp</i>	100 548.3	101 953.2	931.3	97 631.6	100 407.9	1 215.1	96 784.4	98 098.4	849.3
<i>Best/2/exp</i>	99 439.6	101 662.6	1 342.2	98 161.7	99 861.2	1 028.0	95 729.1	97 859.7	1 287.7
<i>Best/3/exp</i>	100 146.9	101 647.6	797.2	98 734.1	99 874.1	742.6	95 904.5	97 580.4	752.9
<i>Best-to-Rand/1/exp</i>	98 811.8	101 855.6	1 184.8	97 026.1	100 490.6	1 317.9	94 744.2	98 129.7	1 243.7
best	98 811.8	101 647.6	797.2	97 026.1	99 861.2	742.6	94 744.2	97 580.4	752.9
avg.	99 781.4	101 797.3	1 030.4	98 065.5	100 115.4	1 045.4	96 130.3	97 903.4	984.5
std.	726.5	126.8	196.2	596.3	251.3	185.5	778.8	209.4	216.0

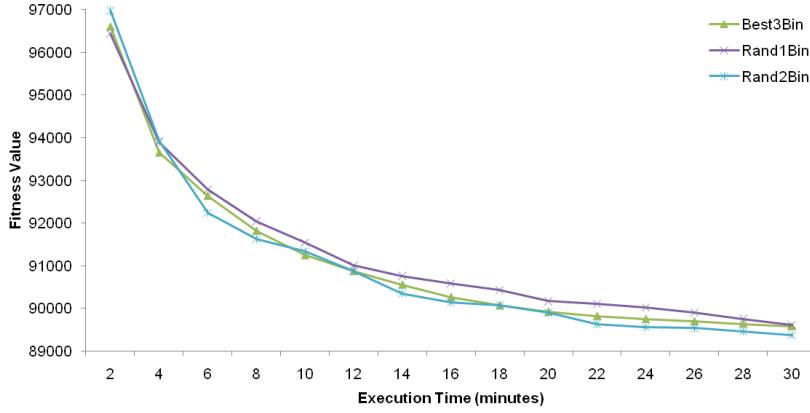


Figure 6.20: Evolution results for the three more efficient strategies using the Denver instance.

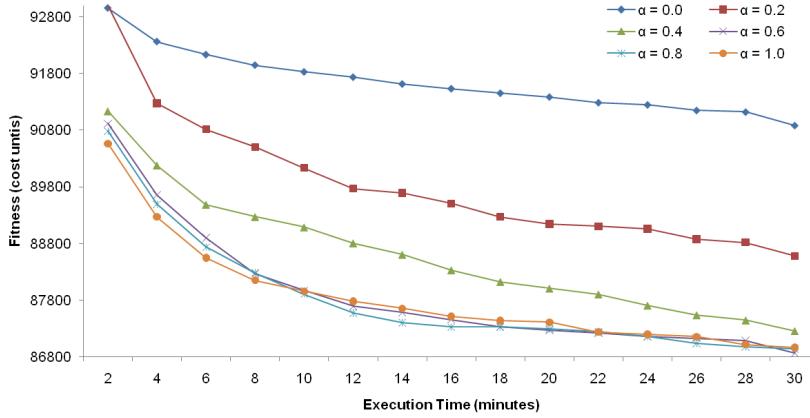


Figure 6.21: α experiment results from the SVNS algorithm (Denver instance).

generated. The chosen strategy was the DE/Rand/2/Bin. The corresponding optimal intrinsic control parameter values are: NP=10, CR=0.2 and F=0.1.

Tuning SVNS α Parameter

Another metaheuristic used in our study was the SVNS algorithm, which is a variant of the VNS algorithm. The SVNS is a trajectory-based algorithm that requires a α (*alfa*) parameter that represents the magnitude of the fitness value and $\in [0.0,1.0]$. In the worst case ($\alpha=1$), a maximum deterioration of the fitness value is permitted, that in the Denver instance is 2612. With a $\alpha=0$, the results of the SVNS are the same as is VNS algorithm.

The α parameters tested are the 0.0, 0.2, 0.4, 0.6, 0.8 and 1. In Figure 6.21 we summarize the results obtained with the several experiments. It has permitted us

Table 6.7: Results obtained for the different α values tested. We present the best, the average and the standard deviation fitness of 30 independent runs in three different time limits (Denver instance).

α value	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
0.0	91 654.2	92 950.1	748.0	90 220.1	91 828.2	1 057.2	89 258.2	90 884.8	1 087.1
0.2	90 865.5	93 004.2	1 624.7	88 602.9	90 133.5	888.6	87 764.5	88 579.4	674.1
0.4	87 604.6	91 135.3	1 810.0	87 093.3	89 088.9	1 182.4	86 156.2	87 257.8	786.7
0.6	86 917.5	90 917.3	1 969.7	86 078.1	87 968.8	1 464.1	86 019.2	86 869.5	881.9
0.8	89 152.8	90 782.1	1 536.7	86 135.2	87 907.0	1 207.3	85 757.9	86 939.8	675.9
1.0	88 510.9	90 564.4	1 480.3	86 355.9	87 956.9	854.9	86 085.7	86 963.7	740.3

to assess that the α parameters have indeed a significant influence on the results accomplished by the SVNS algorithm.

In Table 6.7 it is possible to observe the precise cost results (interference cost) in three specific periods of time. Best results were obtained with the 0.6, 0.8 and 1 values, although they all were very similar. Hence, with the α value of 0.8 it was possible to find the best solution by SVNS (85 757.9 cost units). Furthermore, this was the α value that had a smaller standard deviation (675.9 cost units). Therefore, this was the configuration used to carry out the remaining experiments.

6.4.2 Parameterization

In this subsection, we intent to make a summary of the several parameters that were used by the metaheuristics, after the conclusion of the tuning phase. These parameters were used by the algorithms used for the final experiments performed by the algorithms. We also present some improving features developed for our metaheuristics.

Considerations About our Metaheuristics

In our study, every time it was necessary to incorporate a local search method to improve our algorithms convergence, we have used the same local search method. This local search method was specifically designed to face FAP. The local search method was incorporated in the hybrid DE and also in the VNS and is variant SVNS.

In the population-based metaheuristic, the initial population is created randomly, but incorporating problem domain information. The trajectory-based metaheuristics initial solution is created using the same principle. A more profitable solution is therefore generated. The improvement considers that when assigning a frequency to the TRX, the occurrence of the most costly interferences (co-channel and adjacent-channel) in the same sector is avoided.

The configuration values used by our metaheuristics were:

- Hybrid **DE**: Population Size (NP) = 10, crossover probability (CR) = 0.2, mutation probability (F) = 0.1, using the DE/Rand/2/bin strategy and applying a LS method only after the creation of the x_{trial} solution.
- **VNS**: $k_{max}=9$.
- **SVNS**: $k_{max}=9$, $\alpha=0.8$.

Other Metaheuristics Configurations

To validate our results a comparison with other algorithms developed for the same FAP instances is made. Their configuration parameters were presented in [27, 92, 93]. The stop criterion and the instances used were the same for all the experiments. The differences are in PC characteristics and in the software, because other authors have used Linux as their system operator.

To compare algorithms performance we selected some algorithms that have already been implemented to the instances of the problem that we are addressing. In the comparison we have used several types of metaheuristics. The Steady-State Genetic Algorithm (ssGA) and also the Ant Colony Optimization (ACO) algorithm [92], which was the algorithm that initially have reported the best results for the problem. It was also chosen the (1+2) EA algorithm is a $(\mu + \alpha)$ Evolutionary Algorithm approach which uses a mutation operator specially designed to face FAP [93]. Finally, the Scatter Search (SS) and a Local Search with Heuristic Restarts (LSHR) [93] were also considered.

- **ssGA**: population size=10, uniform crossover with $p_c=1.0$, random mutation with $p_m=0.2$, binary tournament in the selection phase.
- **SS**: population size=40, RefSet size=10, uniform crossover in the solution combination method.
- **(1+2)EA**: $\mu=1$, $\alpha=2$.
- **LSHR**: learning rate $f_r=0.1$.

As a remark, the parameter values presented above for the other authors metaheuristics were also selected after experiments carried out to find the best set of parameters to every algorithm.

Our work has followed the same methodology. It consists in performing extensive experiments to look for the best parameters to every algorithm we implement to solve FAP. The final outcomes of every algorithm are presented and analyzed next.

Table 6.8: Results obtained for the three metaheuristics used with three different time limits. We present the best, the average and the standard deviation fitness of 30 independent runs (Denver instance).

	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
DE*	92 145.8	95 414.2	1 080.4	89 386.4	90 587.2	682.3	87 845.9	89 116.8	563.8
VNS	91 134.2	94 175.3	1 972.8	89 330.1	92 306.5	1 611.4	87 764.9	91 169.7	1 519.3
SVNS	88 125.2	90 842.7	1 598.7	85 935.0	88 008.1	874.7	85 515.2	87 021.9	796.6

6.4.3 Empirical Results: Comparisons Among Algorithms

Concluded the parameter tuning of every algorithm, the next phase tries to assess the performance of the metaheuristics that we have considered to address FAP. A comparative is performed using population-based metaheuristics and trajectory-based metaheuristics.

We present the results in the same order followed in the implementation of the algorithms. Experiments have started with a classic DE algorithm. Then our next goal was to improve the results obtained so far. Therefore, we have developed a new hybrid version of DE algorithm and experiments were done to validate the results.

The conducted experiments showed that using the Local Search (LS) method within the classic DE algorithm had a very important role in its performance. Results presented for the DE (identified as DE*) were obtained with the hybrid version of the DE algorithm described in Section 4.3. The best result obtained was 87 845.9 cost units (after 30 independent runs, each one with 30 minutes). Without the hybridization made in DE, results were much worse. Therefore, a significant conclusion relies on the importance of the hybridization performed with the local search method that can yield improvements regarding the reached frequency plans obtained by the metaheuristics.

Following, we have focused on another metaheuristic named VNS that was implemented as described in Section 4.4. The VNS is a trajectory-based metaheuristic. In VNS we have implemented a common characteristic that was the incorporation of the same local search (LS) method that we have used in the hybridization of DE.

Comparing the results between the DE and the VNS, shown in Table 6.8, the best cost value (87 764.9) was obtained by the VNS algorithm, but in average values (considering the 30 independent executions) the DE shows a better performance. Therefore, between both of these algorithms the DE continues to be the best approach, because the FAP costs of the plans generated by DE are lower considering all the time limits.

After analyzing the performance of above algorithms, a different variant of the VNS algorithm was tested. We have considered the SVNS algorithm, which

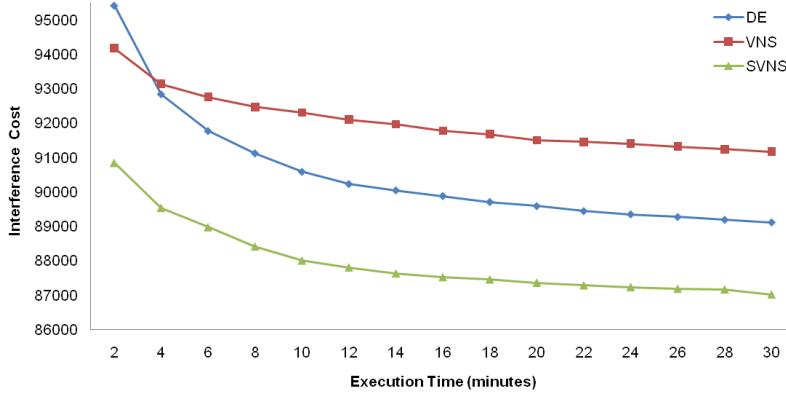


Figure 6.22: Results evolution from 2 to 30 min. with average values of the interference cost for the three metaheuristics used (Denver instance).

was described in Section 4.5. After the experiments performed to define the best value for its α parameter in solving our realistic instances of FAP, the final experiments were executed following the same scenario used previously by the other algorithms. Table 6.8 summarizes all the results obtained with the experiments and Figure 6.22 shows the comparison results for every 2 minutes between the metaheuristics that we have considered in our approach.

Taking into account the final results of every algorithm, we start by comparing the SVNS algorithm with the DE. It permits us to confirm that the SVNS can achieve a better solution (best result of 85 515.2 cost units). It has improved in 2 330.7 units the best solution from DE. Although SVNS clearly beats DE, both algorithms have very good results. In average values, only VNS presents the worst solution, but it still improves the results in 2 650.7 cost units when compared with the results from the SS implementation [26], presented in Table 6.9.

The results previously presented, as we can see, show that the evolution on the accomplished results is very positive. Furthermore, the SVNS shows that it is the best performing algorithm and is capable of solving efficiently the FAP problem. Therefore, one of our conclusions is that for the Denver instance, the trajectory-based metaheuristic gives lower interference plans than the population-based for all time ranges.

Comparing With Other Applied Algorithms

Using the same instances and the same time ranges, we have made a comparative among several types of metaheuristics, in order to assess which the most appropriate for FAP are.

Following our previous conclusions based on our DE algorithm that was mod-

ified to incorporate a Local Search method (i.e., optimizing the assignment of frequencies to every TRX in each sector of the solution), our first results have shown that this hybrid version of the DE algorithm, with the additional features incorporated, clearly makes possible to obtain a viable solution, when compared with other algorithms already proposed by other authors. Additionally, after implementing the VNS and its variant SVNS, we have obtained much more promising results. Indeed our best solution was obtained by the SVNS algorithm, which was an improvement in 2 330.7 cost units from the previous results).

In Table 6.9 we summarize a comparison among the most relevant metaheuristics found in the literature. Results are presented for all time ranges (120, 600 and 1800 seconds) with (1+2) EA that is an Evolutionary Algorithm and a Local Search with Heuristic Restarts (LSHR) [93], the Ant Colony Optimization algorithm (ACO) and the Scatter Search (SS) [26, 93]. All the compared metaheuristics have used the same local-search method. Therefore, the resulting frequency plans computed by the considered algorithms can be properly compared.

The fastest convergence (120 seconds time-limit) algorithms are the SVNS and the ssGA. In long term, our three (DE, VNS and SVNS) perform equally well, considering the difference between the initial solutions and the final frequency plans computed by the algorithms. Taking into account only the algorithms implemented by us, the SVNS was the algorithm that had a better performance for all the time limits.

As we can see, our new approach, the SVNS gets the best value, with 85 515.2 cost units. In an extended analysis of these results, we can conclude that hybrid DE obtains the best results for 600 and 1800 seconds, although it is not the algorithm with the best start. Furthermore, with our approach it was also possible to improve the results obtained in [26, 92] (where other metaheuristics were used: (1, 10) Evolutionary Algorithm, PBIL-Population Based Incremental Learning, etc.).

In conclusion, considering the results that are presented in Table 6.9 in which our results are compared with other different metaheuristics to the same real-world instance of FAP, the main outcomes show that our approach can effectively face the problem being addressed.

In fact, our SVNS approach is among the best algorithms that permitted the achievement of better frequency plans in all the time limits (2, 10 and 30 minutes). The algorithm that can statistically achieve higher results is the ssGA, which is a population-based algorithm. In terms of trajectory-based algorithm, SVNS is indeed the algorithm with the highest convergence and performance.

A second conclusion is that our trajectory-based metaheuristic can indeed handle FAP successfully, as shown by the SVNS algorithm. Actually, our population-based algorithm, such as DE can also get comparable results, but is penalized

in short time ranges (120 seconds). Regarding the population-based algorithms, the ssGA was the algorithm that obtained the best frequency plans. In our case, both DE and SVNS are among the algorithms that had the smallest standard deviation cost at the final time range of 30 minutes.

Table 6.9: Empirical results comparison of the metaheuristics for three different time limits. We present the best, the average and the standard deviation fitness of 30 independent runs (Denver instance).

	120 seconds			600 seconds			1800 seconds		
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.
ACO	90 736.3	93 439.5	1 318.9	89 946.3	92 325.4	1 092.8	89 305.9	90 649.9	727.5
SS	91 216.7	94 199.6	1 172.3	91 069.8	93 953.9	1 178.6	91 069.8	93 820.4	1 192.3
ssGA	87 477.3	89 540.4	991.1	86 755.7	87 850.8	573.6	85 720.3	86 908.9	379.8
DE*	92 145.8	95 414.2	1 080.4	89 386.4	90 587.2	682.3	87 845.9	89 116.8	563.8
VNS	91 134.2	94 175.3	1 972.8	89 330.1	92 306.5	1 611.4	87 764.9	91 169.7	1 519.3
SVNS	88 125.2	90 842.7	1 598.7	85 935.0	88 008.1	874.7	85 515.2	87 021.9	796.6
(1+2)EA	87 763.9	92 294.0	1 407.6	86 064.8	89 669.8	1 164.8	85 607.3	88 574.3	1 100.3
LSHR	88 543.4	92 061.7	585.3	88 031.0	89 430.9	704.2	87 743.0	88 550.3	497.0

6.5 Multiobjective Metaheuristics Empirical Results

This section is devoted to presenting and comparing our multiobjective algorithms results against other well-known multiobjective algorithms.

6.5.1 Configuration Parameters: Experimental Study

Preliminary experiments were carried out to find the best set of parameter values for the proposed multiobjective metaheuristics. The results are shown and analyzed next. In the tuning tests, 10 instead of 30 independent runs were implemented for every parameter.

DEPT Tuning Parameters

DEPT has many variants and involves three intrinsic control parameters. The tuning process aims to find the best variant and its corresponding optimal intrinsic control parameter values. It starts from DE/Rand/2/Bin strategy with a CR=0.2 and F=0.1. These are the best parameters based on the previous work with a single-objective version of FAP.

The tuning is performed in the following order:

1. Identify the best population size (N) value.
2. Choose the best crossover probability (PC) value.
3. Choose the best mutation intensity (PM) value.
4. Choose the most efficient differential evolution strategy.

Population size Population sizes tested are 10, 25, 50, 75 and 100. Figure 6.23 and Figure 6.24 show the corresponding Pareto Fronts. In general, higher NP values lead to better hypervolume. The NP value set for 75 is the value that works best for both instances, with 24.8% of hypervolume for Denver instance and 23.8% of hypervolume for Seattle instance.

Crossover Probability The next experiment was focused on crossover probability (CR). Results for the CR parameter are tabulated in Table 6.10. The CR parameter normally uses values between [0.0, 1.0]. Our first set of values was 0.1, 0.25, 0.5, 0.75 and 0.9. The results from this first set of experiments clearly identified that the best solutions were found using a CR value of 0.1. A CR=0.1 has the highest mean hypervolume (27.5% in average values, considering both instances) and smallest standard deviation value (1.1%).

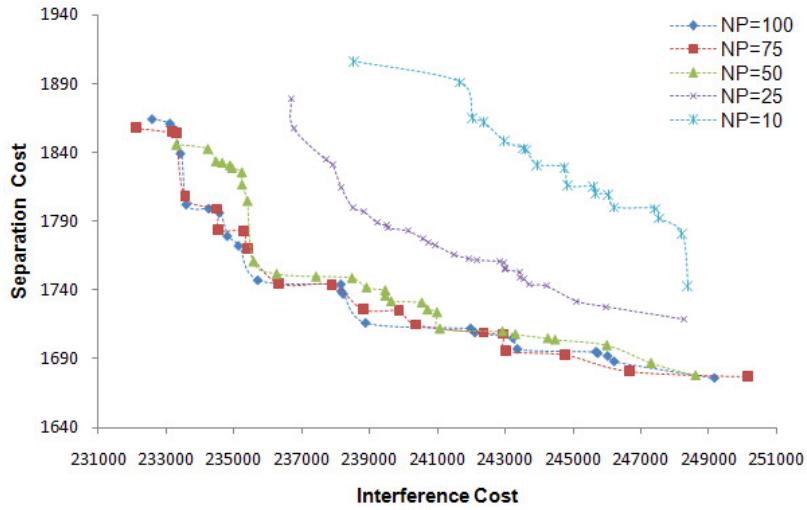


Figure 6.23: Experiments using the Denver instance, showing the Pareto Front for the population size (NP) parameter values.

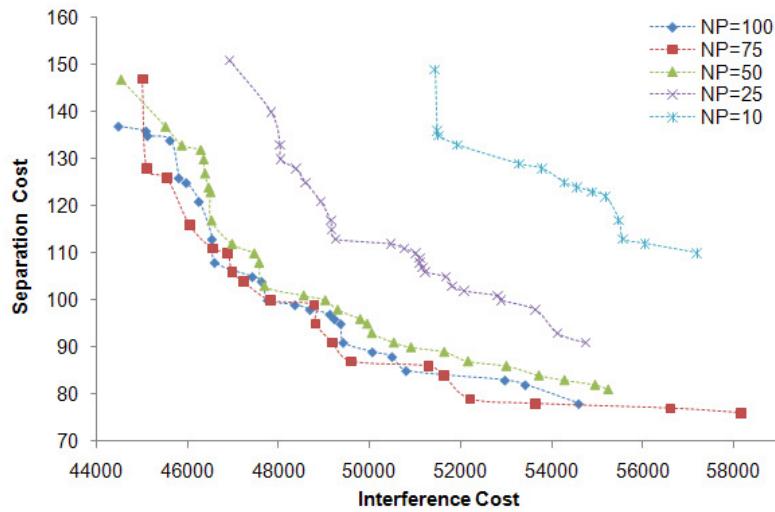


Figure 6.24: Experiments using the Seattle instance, showing the Pareto Front for the population size (NP) parameter values.

Inspired by the above findings, we further investigate if smaller crossover probability values could lead to better results. The second set of crossover probability values 0.01, 0.02, 0.03, 0.05, 0.07 and 0.09 is therefore tested. The 0.01 and the 0.03 were the ones showing the highest hypervolume. This has led us to check if its middle value of 0.02 was the best. In average values with a CR of 0.01 we accomplished the highest hypervolume of 37.3% that represents the best approach so far. Indeed, considering the two cost functions being minimized, the best average value for the interference cost (C_I) and for the separation cost

(C_S) were obtained using the CR=0.01 (for both instances). Precisely, in Denver instance an hypervolume of 28.3% was achieved with an average C_I of 232 924.9 units and 1 713.6 units for C_S . For Seattle instance the hypervolume was 46.4%, with 37 677 units for C_I and 73.8 units for C_S .

Mutation Intensity Normally, the weighting factor F is in the range [0.0, 1.0]. The set of tested mutation values are the 0.1, 0.25, 0.5, 0.75 and 0.9. Table 6.11 shows the results for the several F values. An F value of 0.9 was the value selected as the best configuration; therefore it will be used for the following experiments. By comparing the statistical indicators and also the hypervolume, we could conclude that for DEPT the higher F is more favorable. Indeed, for the Seattle and the Denver instances the best average C_I and C_S values were obtained using an F value of 0.9. Regarding the hypervolume, for Denver instance a hypervolume of 29.2% was obtained and for Seattle instance of 47.6%.

DEPT Strategies For DEPT the final parameter to be tuned represents the scheme to be used. As mentioned in Chapter 5.4 there are several types of schemes that can be applied to generate new solutions. They differ in terms of the base differential mutation, number of difference vectors and crossover schemes. To better assess the performance of every possible scheme, we have grouped them according to the implemented crossover scheme: the schemes using binomial crossover (*bin*) and the schemes using exponential crossover (*exp*).

The way in which the trial vector is generated has an enormous influence in the final performance of the algorithm. The creation of a new solution is directly associated with the scheme that is being used, for example, with the scheme *DE/Best/1/bin*, the best-so-far individual (best) is always selected as the base individual, and added a single (1) scaled vector difference to it. Then it is created a trial vector by uniformly crossing (bin) the resulting mutant with the target individual.

Table 6.12 shows the statistical results obtained by the several schemes implemented in DEPT. It is apparent that DEPT schemes using an exponential crossover selection to generate the trial individual outperform those using a binomial crossover scheme.

Analyzing the hypervolume values for the Seattle instance, in average values, the hypervolume was 46.6% for the binomial strategies and 67.7% for the exponential strategies. The same behavior happened also with the Denver instance, obtaining a 28.8% for the binomial strategy and 43.2% for exponential strategies.

It is further noticed that *DE/Rand-To-Best/1/Exp* scheme is the most profitable for both instances. They have obtained a final hypervolume of 69.4% for Seattle instance and 42.7% for Denver instance. In average values, considering

Table 6.10: Results obtained for the DEPT crossover parameter (CR) using both instances. We present the best, average, standard deviation and the hypervolume value. The last column comprises the hypervolume mean value for each row.

CR	Seattle Instance						Denver Instance						Avg. HV	
	Interference Cost		Separation Cost		HV		Interference Cost		Separation Cost		HV			
Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	HV		
0.1	42306.3	46524.7	3344.8	64	100.0	24.2	28.3%	229511.2	236070.5	6148.3	1644	1750.6	78.6	26.7%
0.25	45582.5	48284.3	1888.3	84	108.4	15.4	22.0%	233629.2	238991.0	3854.8	1695	1758.1	40.0	23.9%
0.5	47664.6	50018.4	1886.2	91	117.0	17.0	18.9%	235176.8	240475.4	3891.8	1719	1773.7	32.8	22.6%
0.75	47516.7	51134.9	1975.1	97	111.6	9.4	18.0%	236098.1	240831.8	3318.2	1729	1784.5	37.9	22.1%
0.9	48379.7	51037.7	2542.1	91	116.9	22.3	18.3%	237040.9	241193.0	3535.5	1722	1776.9	39.2	22.0%
0.01	34882.4	37677.0	2300.1	32	73.8	25.1	46.4%	226637.1	232924.9	3175.0	1634	1713.6	41.1	28.3%
0.02	37774.2	41320.4	2833.5	36	76.9	25.6	40.9%	226734.7	233724.5	4329.7	1625	1717.5	56.0	28.4%
0.03	39066.0	42985.6	3112.7	42	82.3	26.7	37.5%	225774.4	233053.9	5646.1	1606	1702.3	64.3	29.4%
0.05	40693.5	45208.6	3724.5	47	85.5	26.2	33.9%	226621.7	234546.1	5947.0	1612	1712.6	67.3	28.8%
0.07	41746.3	45989.1	3510.9	55	91.0	25.7	31.1%	228018.0	236359.9	6402.0	1628	1712.6	58.8	29.5%
0.09	42604.6	46763.2	3589.9	59	95.1	24.6	29.2%	228558.5	236690.3	5624.7	1640	1728.2	69.6	27.1%

Table 6.11: Results obtained for the DEPT F parameter using both instances. We present the best, average, standard deviation and the hypervolume value. The last column comprises the hypervolume mean values for every distinct F value.

F	Seattle Instance						Denver Instance						Avg. HV		
	Interference Cost			Separation Cost			Interference Cost			Separation Cost					
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.			
0.1	34 704.0	37 737.7	2 384.5	33	72.6	24.2	46.3%	225 994.0	232 824.8	3 370.1	1 628	1 715.4	43.0	28.6%	37.5%
0.25	34 621.4	38 029.6	2 603.3	28	68.2	25.3	47.6%	226 297.3	233 180.0	3 542.1	1 609	1 697.1	47.2	29.1%	38.3%
0.5	34 882.7	38 434.7	2 755.5	27	65.1	23.7	47.5%	225 867.5	233 830.9	3 768.6	1 605	1 700.4	47.2	29.3%	38.4%
0.75	34 654.1	37 772.9	2 363.1	31	73.4	25.9	46.7%	225 872.1	232 124.7	3 627.9	1 611	1 701.4	45.9	29.3%	38.0%
0.9	34 311.6	37 602.5	2 452.8	29	71.7	26.2	47.6%	225 865.6	231 808.6	3 528.9	1 615	1 699.0	45.7	29.2%	38.4%

both instances, the *DE/Rand-To-Best/1/Exp* scheme reaches a hypervolume of 56%.

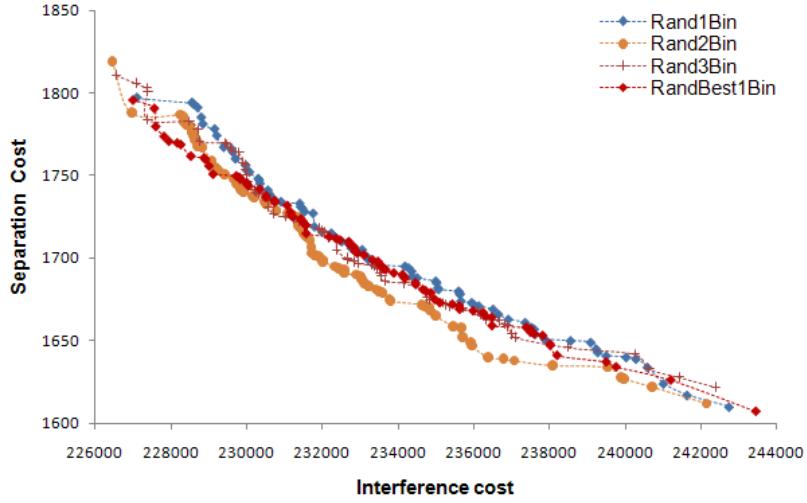


Figure 6.25: Pareto Fronts obtained using *binomial* crossover with *random* base schemes over the Denver instance.

In terms of differential mutation base, it can be seen from Figure 6.25, 6.26, 6.27, 6.28, 6.29, 6.30, 6.31 and Figure 6.32 that random base is more beneficial. These results are much clearer from results by rand/*/bin, because these schemes have a better performance. Indeed, comparing the solutions in the Pareto Front generated using a random scheme, they all have less cost units when considering

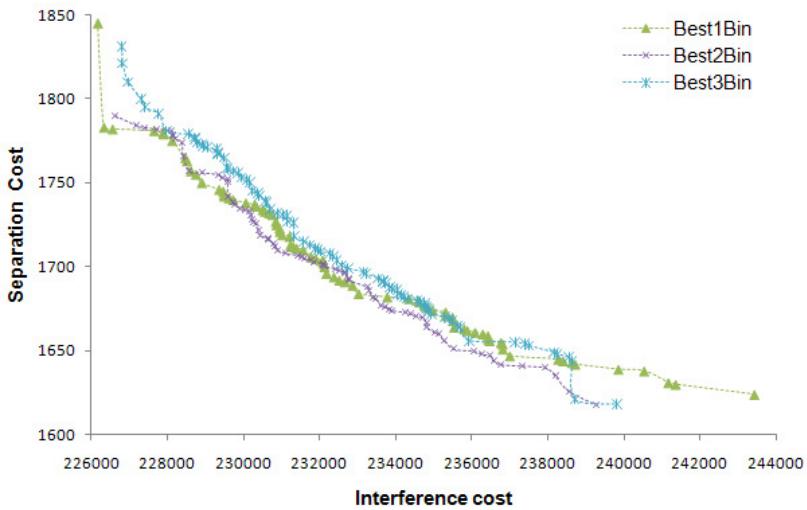


Figure 6.26: Pareto Fronts obtained using *binomial* crossover with *best* base schemes over Denver instance.

Table 6.12: Results obtained for the different DEPT schemes using both instances. We present the best, average, standard deviation and the hypervolume value.

DEPT Strategies	Seattle Instance						Denver Instance						Avg. HV	
	Interference Cost			Separation Cost			Interference Cost			Separation Cost				
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.		
Rand/1/Bin	34.316.6	37.842.2	2.115.5	32	70.1	24.1	46.8%	227.092.5	233.811.1	3791.5	1.610	1.704.6	48.1	
Rand/2/Bin	34.366.2	37.607.5	2.447.2	29	71.6	26.1	47.6%	226.454.4	232.203.9	3.350.2	1.612	1.710.4	48.6	
Rand/3/Bin	34.929.5	37.953.2	2.188.0	33	73.6	24.4	45.9%	226.562.1	232.800.7	3.540.5	1.622	1.711.7	47.2	
Best/1/Bin	35.379.8	38.521.6	2.317.5	29	71.6	25.0	46.2%	26.177.4	232.957.6	3.826.0	1.624	1.702.7	45.9	
Best/2/Bin	34.859.8	38.066.1	2.360.1	32	73.9	26.0	46.1%	226.618.3	232.110.4	3.005.9	1.618	1.705.9	44.2	
Best/3/Bin	35.135.6	38.215.4	2.310.4	31	71.5	24.8	46.2%	226.795.7	232.356.9	3.192.4	1.618	1.716.0	48.7	
Rand-to-Best/1/Bin	34.317.6	37.375.5	2.269.5	30	71.4	26.3	47.6%	226.983.0	233.092.7	3.471.3	1.607	1.705.6	42.2	
Rand/1/Exp	20.034.4	26.083.2	2.179.6	25	71.7	23.6	67.1%	200.990.1	226.315.2	7.531.0	1.427	1.661.2	65.1	
Rand/2/Exp	20.560.0	26.006.0	2.136.4	25	71.2	22.6	66.9%	202.740.2	228.028.3	10.216.0	1.455	1.586.2	73.0	
Rand/3/Exp	19.954.5	26.062.3	2.350.2	22	70.3	23.6	68.2%	203.580.0	226.350.3	6.738.6	1.438	1.661.2	61.3	
Best/1/Exp	20.400.1	26.383.4	2.298.0	27	73.2	23.3	65.9%	201.437.5	226.367.9	6.571.7	1.433	1.659.3	60.0	
Best/2/Exp	19.287.6	26.047.1	2.511.8	21	74.0	25.1	68.7%	201.981.7	226.860.6	6.359.7	1.448	1.654.0	62.4	
Best/3/Exp	20.220.7	26.121.7	2.371.0	23	72.0	24.8	67.5%	202.988.4	225.976.8	6.592.8	1.443	1.661.8	54.9	
Rand-to-Best/1/Exp	18.123.0	23.897.4	2.144.3	27	70.9	23.7	69.4%	203.283.1	227.214.2	6.411.9	1.444	1.661.4	51.6	

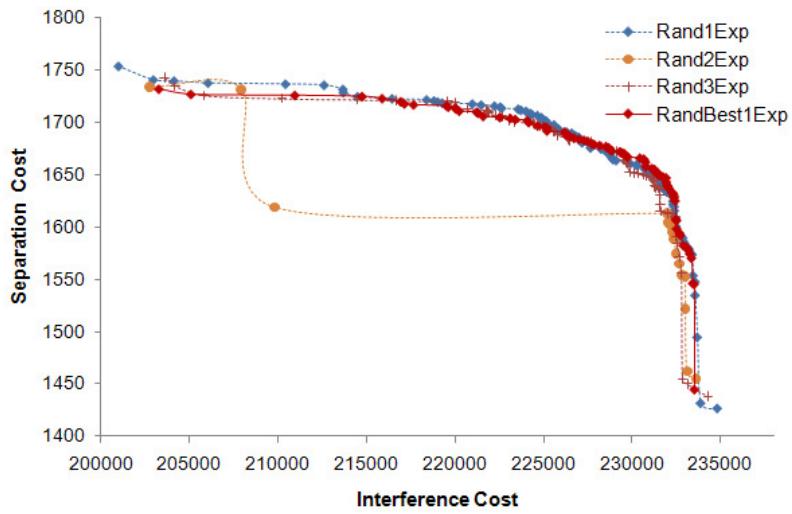


Figure 6.27: Pareto Fronts obtained using *exponential* crossover with *random* base schemes over the Denver instance.

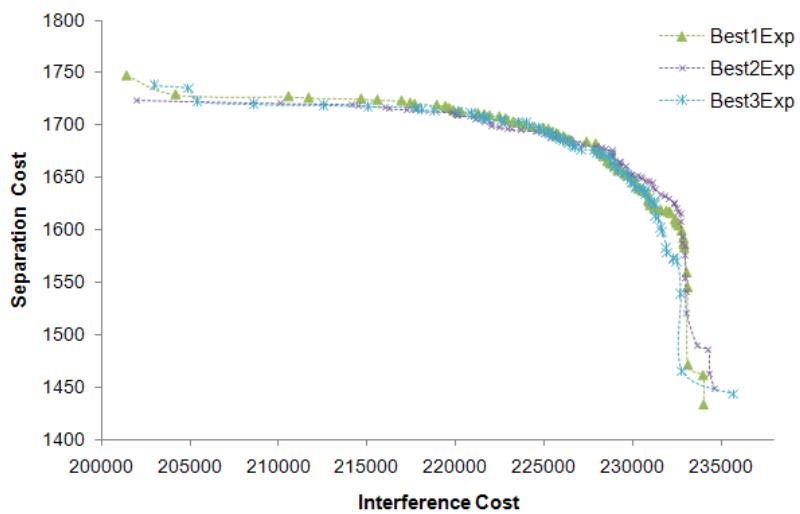


Figure 6.28: Pareto Fronts obtained using *exponential* crossover with *best* base schemes over Denver instance.

the two objective functions being minimized. These results are even more evident in the schemes using the binomial crossover.

However, as shown in Figure 6.31 in the Seattle instance, DE/Rand-to-Best/1/Exp clearly outperforms all other approaches.

The results obtained when comparing the random schemes against the schemes using the best-so-far approach, do not express a so obvious difference between the two differential mutation base approaches. Indeed, in Figure 6.28, 6.27, 6.32 and Figure 6.31 that are showing the Pareto Fronts for the schemes using the *exponential* crossover, it is possible to identify that the differences between the two approaches are small, especially for the Denver instance (Figure 6.28 and 6.27).

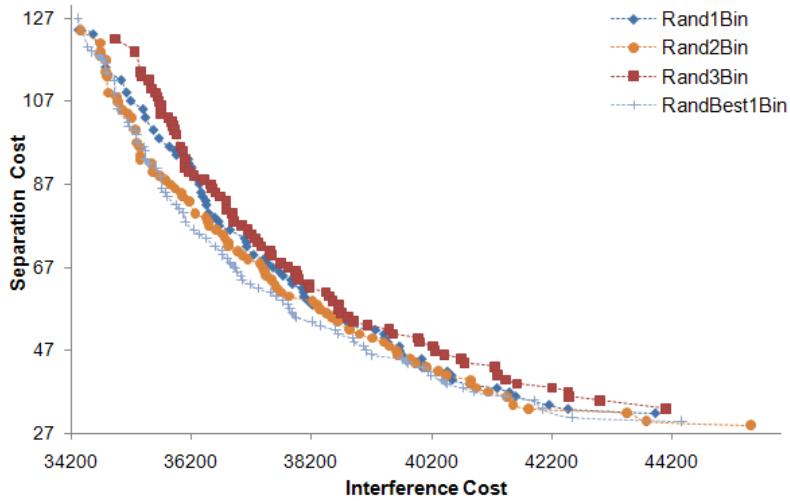


Figure 6.29: Pareto Fronts obtained using *binomial* crossover with *random* base schemes over the Seattle instance.

Tuning the MO-SVNS and GMO-SVNS α Parameter

Both MO-VNS and MO-SVNS use the k parameter to control the neighborhood search space. In our work we use the $k_{max} = 9$. The k parameter starts with the value of 1 and increments until it reaches a k_{max} of 9. It means that when generating a new neighborhood, it will change $k * 10$ (for example, the 90% for $k=9$) of the assigned frequencies to the TRXs (representing a neighborhood with $k * 10$ of different configuration). Besides the k parameter, the MO-SVNS also uses an α parameter value.

Therefore, for the MO-SVNS and the *greedy* MO-SVNS algorithms, we have analyzed the influence of α parameter in their performances. Both algorithms have used the same α value. Table 6.13 summarizes the results for the α parameter using MO-SVNS. It shows the statistical values (minimum, average and

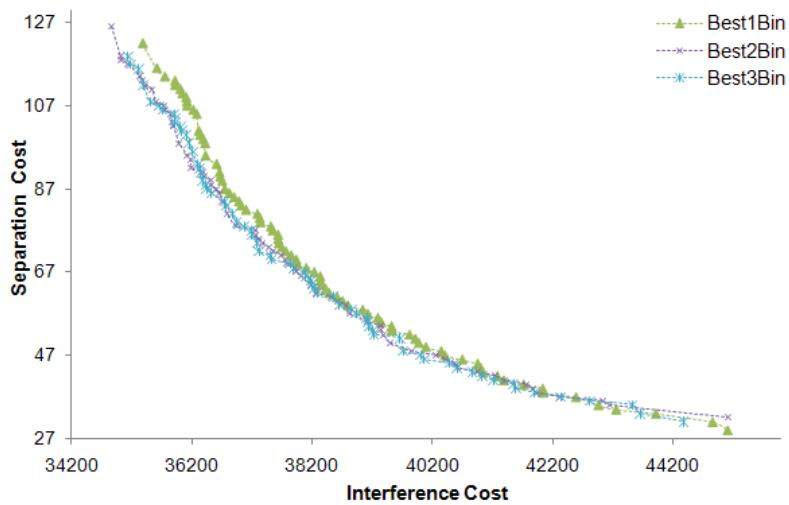


Figure 6.30: Pareto Fronts obtained using *binomial* crossover with *best* base schemes over the Seattle instance.

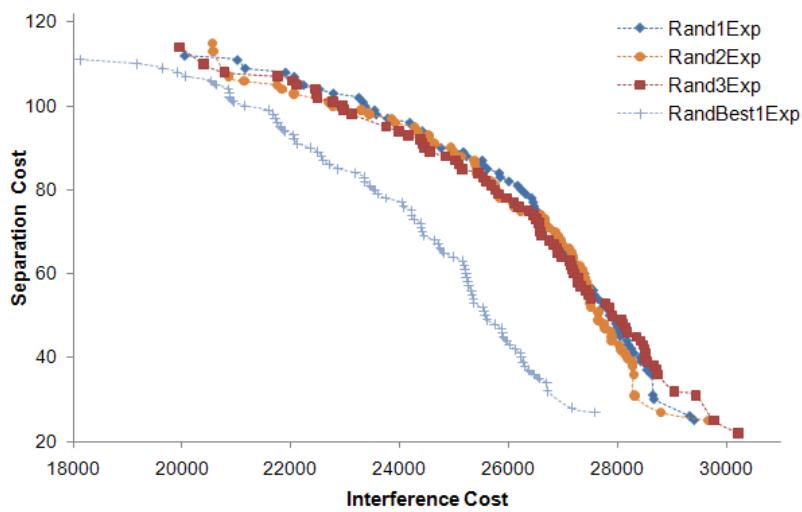


Figure 6.31: Pareto Fronts obtained using *exponential* crossover with *random* base scheme over the Seattle instance.

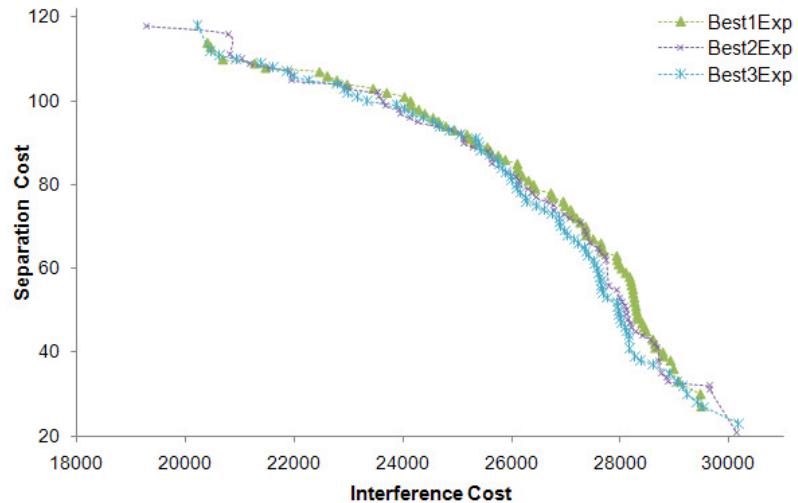


Figure 6.32: Pareto Fronts obtained using *exponential* crossover with *best* base scheme over the Seattle instance.

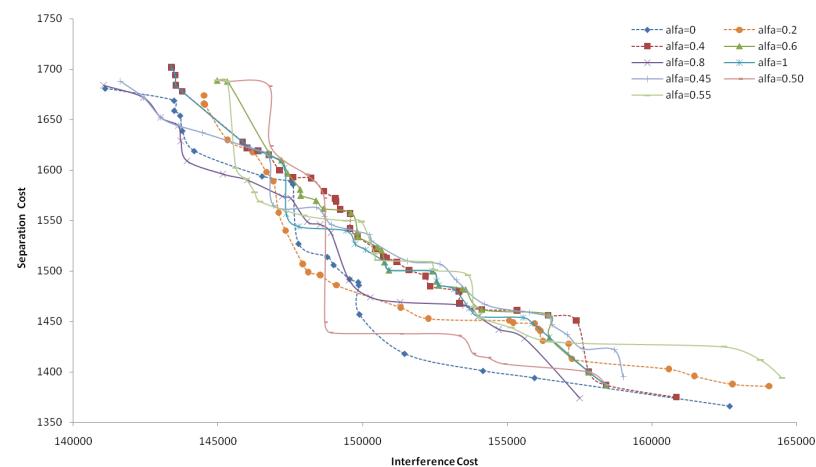


Figure 6.33: Simulation results with α values for Denver instance, showing the Pareto Front of 10 independent runs.

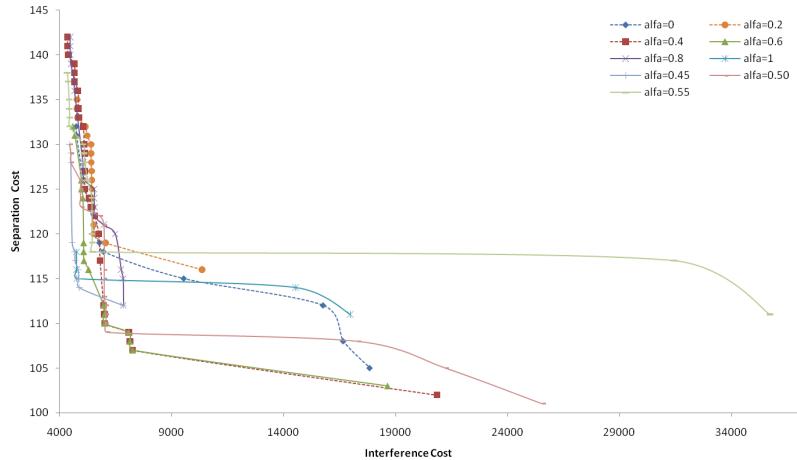


Figure 6.34: Simulation results with α values for Seattle instance, showing the Pareto Front of 10 independent runs.

standard deviation), besides the hypervolume indicator. At first we have analyzed the performance of the following α values: 0, 0.2, 0.4, 0.6, 0.8 and 1. From this initial range of values, we identified that α values between 0.4 and 0.6 had on average (considering the Denver and Seattle instances) better quality indicators.

Inspired by the findings, we further investigated inside the range –the α values of 0.45, 0.5 and 0.55. After analyzing the results obtained with every tested value, we concluded that the best α value was the 0.4. A $\alpha=0.4$ has the higher average hypervolume of 65.1%, considering both instances. Figure 6.33 and Figure 6.34 show the Pareto fronts obtained with the several α values.

MO-ABC Tuning Parameters

The MO-ABC algorithm uses four control parameters: population size (colony size), maximum cycle number, limit and probability of mutation (PM) that are to be predetermined by our first set of experiments. The maximum cycle number in our case corresponds to the 30 minutes of execution. Therefore, this parameter does not need to be tuned. Next, we present the preliminary results obtained in the tuning of the MO-ABC algorithm.

Colony Size The colony size represents the number of individuals that are part of the initial population and it represents the first parameter to be tuned. The set of values tested were: 50, 75, 100, 125, 150, 175, 200, 225, 250, 275 and 300. The initial parameter of MO-ABC was set to a limit of 100, which is the common value of the standard ABC algorithm [74].

Table 6.14 shows the results obtained with the tested colony size values. With the colony size experiment, we have concluded that with a medium colony size

Table 6.13: Results obtained with the α parameter in *greedy* MO-SVNS, using both instances. We present the best, average, standard deviation and the hypervolume value.

α value	Seattle Instance						Denver Instance					
	Interference Cost		Separation Cost		HV	Best	Interference Cost		Separation Cost		HV	Avg. HV
	Best	Avg.	Std.	Best			Best	Avg.	Std.			
0	4739.5	8836.8	5269.1	105	118.7	8.6	49.6%	141 106.6	148 520.0	5024.2	1 366	1537.0
0.2	4779.7	5630.2	1260.7	116	126.6	5.6	44.7%	144 516.3	152 160.4	6084.2	1 386	1500.7
0.4	4356.4	5975.0	3142.5	102	124.9	12.2	51.7%	143 395.7	150 396.1	4497.8	1 375	1541.8
0.6	4595.6	6431.3	3372.0	103	116.8	8.8	51.3%	144 978.0	150 936.7	3693.1	1 387	1529.1
0.8	4483.5	5423.6	847.2	112	127.4	9.5	47.0%	141 049.1	148 615.6	4762.1	1 374	1541.8
1	4758.0	9161.4	6081.8	111	114.8	2.6	47.2%	143 395.7	150 354.4	4571.0	1 387	1535.4
0.45	4448.6	4881.6	707.5	112	119.8	6.7	47.2%	141 626.5	150 967.4	5694.5	1 380	1529.0
0.50	4448.6	7969.0	6152.6	101	117.3	8.8	51.7%	144 978.0	150 632.0	4285.8	1 387	1514.8
0.55	4344.0	8503.0	9829.9	111	126.9	8.0	46.0%	144 978.0	151 984.8	5964.3	1 394	1520.4

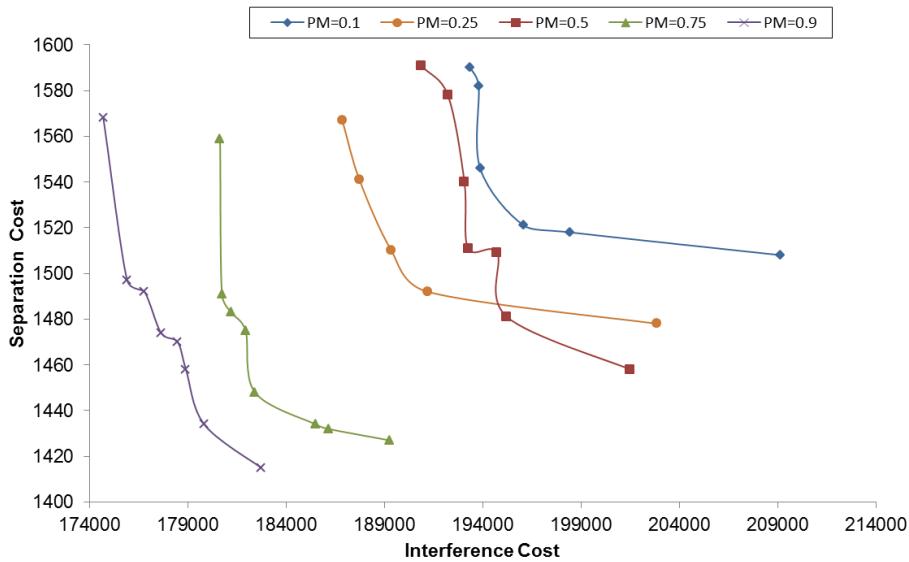


Figure 6.35: MO-ABC Pareto Fronts obtained using the several probabilities of mutation tested over Denver instance.

value it is possible to observe a positive evolution of hypervolume results. On average values, the highest hypervolume (46.4%) was obtained using a colony size value of 100, therefore it was the value selected to be used in the next experiment.

Limit The second experiment intents to determine the best *limit* value. To proceed with the experiment we have fixed the colony size to 100 (from the first experiment) and started to perform the experiments with the set of values 50, 75, 100, 125 and 150. Table 6.15 summarizes the results obtained with the limit values. In average (\bar{x}) values the HV was 47.4% with a limit of 75.

Probability of mutation The third experiment intents to determine the best *probability of mutation* (PM). To proceed with the experiment we have fixed the colony size to 100 (from the first experiment) and the limit to 75 (from the previous experiment). The PM values tested were: 0.1, 0.25, 0.5, 0.75 and 0.9. The experiments permitted to identify that by increasing the PM value, the Pareto Fronts hypervolume also increases. Figures 6.35 and 6.36 show the final Pareto Fronts obtained for the tested PM values, while Table 6.16 summarizes the results obtained. Finishing the experiments with the MO-ABC parameters, the final configuration was defined to a *colony size* of 100, *limit* of 75 and a *probability of mutation* (PM) of 90%.

Table 6.14: MO-ABC results for experiments against *colony size* parameter. We present the average values (\bar{x}) for both interference cost (IC) and separation cost (SC) values and percentage (%) hypervolume (HV).

Colony size	50	75	100	125	150	175	200	225	250	275	300
Seattle	IC \bar{x}	25903.3	41456.4	30319.2	36176.1	39111.9	37402.0	41868.4	42745.6	42283.1	43553.6
	SC \bar{x}	138.9	132.1	126.0	123.3	117.5	127.0	120.0	117.5	124.5	123.8
	HV	34.4	24.4	37.6	32.9	30.4	31.5	27.4	26.9	26.7	25.0
Denver	IC \bar{x}	209239.6	175895.0	180935.6	222575.3	221809.4	195400.0	237122.4	196121.4	206805.6	196145.7
	SC \bar{x}	1552.3	1465.2	1496.0	1534.6	1741.8	1504.7	1593.9	1567.1	1512.3	1645.8
	HV	50.1	59.4	55.2	47.7	33.3	50.1	38.5	48.4	48.2	51.0
\bar{x} HV	42.2	41.9	46.4	40.3	31.8	40.8	32.9	37.7	37.5	38.0	31.4

Table 6.15: MO-ABC results for experiments against *limit* parameter. We present the average values (\bar{x}) for both interference cost (IC) and separation cost (SC) values and percentage (%) hypervolume (HV).

Limit values		50	75	100	125	150
Seattle	IC \bar{x}	30 949.9	30 654.7	30 319.2	30 623.8	30 623.1
	SC \bar{x}	124.9	125.4	126.0	125.5	125.5
	HV	37.6	37.7	37.6	37.6	37.6
Denver	IC \bar{x}	179 383.9	180 836.3	180 935.6	183 055.8	183 055.8
	SC \bar{x}	1 499.1	1 495.3	1 496.0	1 496.4	1 496.4
	HV	55.9	57.1	55.2	56.6	56.6
\bar{x} HV		46.8	47.4	46.4	47.1	47.1

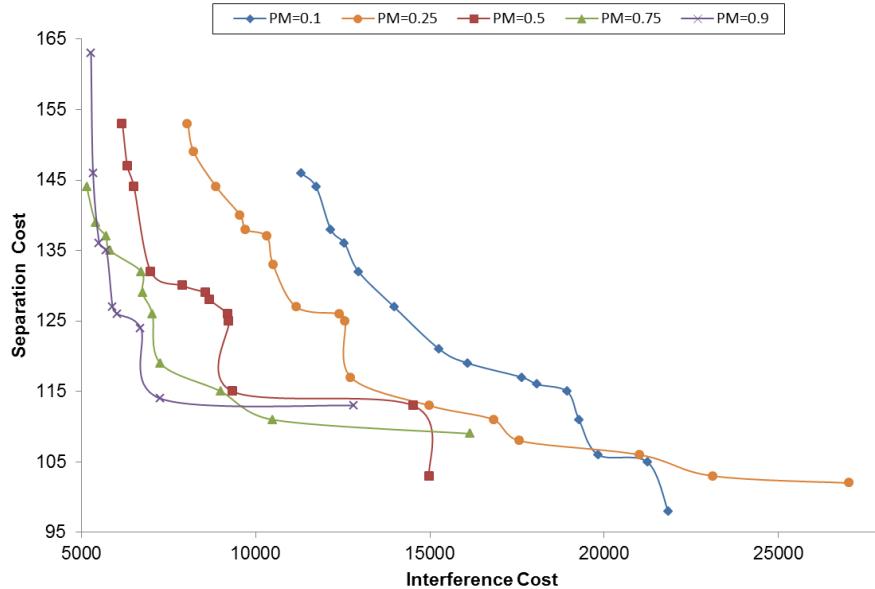


Figure 6.36: MO-ABC Pareto Fronts obtained using the several probabilities of mutation tested over Seattle instance.

NSGA-II Tuning Parameters

The good performance of the NSGA-II algorithm depends on the parameters to be used. Therefore, the first experiment scenario with the algorithm was the tuning of the parameters to address our real-world FAP instances. The experiments with the NSGA-II parameters were done in the following order:

1. Identify the best population size (N) value.
2. Choose the best probability of crossover (PC) value.

Table 6.16: MO-ABC hypervolume (HV) results for experiments against probability of mutation. The HV values are shown in percentage (%).

PM		0.1	0.25	0.5	0.75	0.9
Seattle	IC \bar{x}	16 186.8	13 797.3	9 026.7	7 762.6	6 720.4
	SC \bar{x}	122.1	125.4	128.8	126.9	131.6
	HV	47.0	47.6	49.4	47.6	45.8
Denver	IC \bar{x}	197 429.2	191 581.8	194 395.0	183 468.7	178 118.6
	SC \bar{x}	1 544.2	1 517.6	1 524.0	1 468.6	1 476.0
	HV	47.8	52.1	51.2	57.9	61.3
\bar{x} HV		47.4	49.9	50.3	52.7	53.5

3. Choose the best probability of mutation (PM) value.
4. Choose the most efficient crossover operator (cxOp).

The NSGA-II initial configuration was set to the most common one recommended: $N=100$, $PC=0.9$, $PM=0.2$ and uniform crossover (UX). We have followed a pre-determined sequence of tests. Therefore, the results obtained with the previous parameters were considered to the subsequent parameters experiments.

Population size The set of values tested for the population size parameter was: 25, 50, 75, 100 and 125. From the results obtained with the several N values, a population size of 50 individuals was selected as the best value. Indeed, with $N=50$ the highest hypervolume was achieved (20.6% for Denver instance and a 14.8% for Seattle instance). It represents the highest hypervolumes obtained for both instances. Also, with Denver instance (the most complex instance) with $N=50$ the minimum interference cost and separation cost values were also found, that are 238 346.5 and 1 754 units, respectively.

For all the remain N values, the accomplished average hypervolumes were smaller, despite being relatively near to the values obtained with the best value. Table 6.17 shows the several results obtained in the experiments performed with the population size parameter. Hence, a population size of 50 individuals was the selected value to be used by the subsequent experiments.

Crossover probability In the crossover operator, the both parents used in the generation of the offspring are selected by a binary tournament. Indeed, the crossover probability defines how often will the crossover be performed. If there is no crossover, the offspring is an exact copy of its parents. If there is a crossover, the offspring is created from different parts of its parents.

Table 6.17: Results obtained for the NSGA-II population size (N) parameter using both instances. We present the best, average, standard deviation and the hypervolume value. The last column comprises the hypervolume mean values for every distinct N value.

N value	Seattle Instance						Denver Instance						Avg. HV		
	Interference Cost			Separation Cost			Interference Cost			Separation Cost					
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.			
25	50331.6	52609.1	141.9	119	124.2	5.4	12.8%	240254.1	243617.2	23337.4	1765	1814.1	29.8	19.7%	16.2%
50	50364.3	52074.3	1659.9	105	120.0	13.1	14.8%	238346.5	244171.9	4398.0	1754	1791.6	35.3	20.6%	17.7%
75	49193.8	51701.1	1942.7	111	121.8	8.6	14.6%	241517.2	244257.8	3132.1	1759	1799.6	33.6	19.5%	17.1%
100	50433.0	52377.6	1682.4	114	122.2	11.8	13.5%	239839.1	242767.3	2731.8	1762	1798.7	28.6	20.0%	16.7%
125	50562.1	51902.2	899.7	114	120.0	4.8	13.5%	240671.3	244265.5	3381.7	1764	1792.5	27.7	19.7%	16.6%

If the crossover probability is 100% ($PC = 1$), then every offspring is made by crossover. If it is 0% ($PC = 0$), a whole new individual is made from an exact copy of its parents from the old population.

In the experiments to tune the PC parameter, the set of tested values were $PC=0.1, 0.25, 0.5, 0.75$ and 1 . The best results from the first set of values were obtained using a crossover probability of 10%. Then, we further investigated the influence of other set of values. The new subset of values tested included: $0.01, 0.03, 0.05, 0.07$ and 0.09 . All the results showed that the highest hypervolumes were obtained with a crossover probability value of 7%. In Table 6.18, for both problem instances, the statistical results for both objectives being minimized, and their hypervolumes are shown. On average values, the highest hypervolume (17.4%) was accomplished using a $PC=0.07$. With the most complex instance, which is Denver instance, the highest hypervolume (20.6%) was obtained with a PC of 0.07 , and also for Seattle instance the obtained hypervolume (14.3%) was among the best ones.

Therefore, a crossover probability of 7% was the value chosen to be used by the following experiments to tune the other NSGA-II parameters.

Mutation probability Mutation probability controls how often will parts of individual chromosome be mutated. In FAP it represents the change of a frequency value assigned to a TRX. If there is no mutation, the offspring is taken after crossover without any change. If a mutation is performed, part of the chromosome is changed.

The mutation operator assigns a random frequency value to a TRX from unused frequencies of the sector in which the TRX is installed. The set of available frequencies of the specific sector is updated accordingly. Thus, through that random mutation, the cost K inside the same sector is avoided (interferences inside the same sector). To optimize that operation it is kept an updated list containing all the valid frequencies (the unused frequencies) in every sector.

Several set of values were tested for mutation probability. Initially, the first mutation probabilities were: $0.01, 0.1, 0.2, 0.3, 0.4$ and 0.5 . The scenario used in these experiments followed the same approach as the previous ones: the same parameters were kept unaltered, changing only the PM value.

From the initial set of values, the best values were obtained using smaller PM values. Therefore, we have tested an additional range of values that contained smaller probabilities. The second set was: $0.03, 0.05, 0.07$ and 0.09 . Smaller probability values accomplished higher HV, therefore we have tested even smaller values: $0.001, 0.003, 0.005, 0.007$ and 0.009 . From Figures 6.37 and 6.38 it is clearly observed that the best performance of NSGA-II algorithm was accomplished using the smallest PM value. The mutation probability behavior was similar on

Table 6.18: Results obtained for the NSGA-II crossover parameter (PC) using both instances. We present the best, average, standard deviation and the hypervolume value. The last column comprises the hypervolume mean value for each row.

PC value	Seattle Instance						Denver Instance						Avg. HV	
	Interference Cost			Separation Cost			Interference Cost			Separation Cost				
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.		
0.1	50238.9	52 022.6	2 634.6	104	113.0	9.0	15.1%	241 280.1	244 636.8	3 594.7	1 774	23.3	19.2%	
0.25	50582.5	52 260.0	1 288.7	114	125.5	13.2	13.3%	239 422.6	243 695.2	3 155.8	1 775	1800.5	23.4	19.8%
0.5	49524.7	50 884.5	1 449.8	109	129.0	25.7	14.6%	241 045.0	244 589.6	4 004.4	1 766	1797.0	25.6	19.5%
0.75	49308.3	52 479.2	2 539.1	115	131.1	14.4	13.5%	240 825.7	243 725.3	2 060.9	1 757	1802.1	27.4	19.8%
0.9	50111.1	52 660.4	2 016.2	115	126.7	8.8	13.3%	241 367.4	245 541.3	3 750.5	1 754	1794.2	33.8	19.6%
1.0	50330.0	51 902.7	2 363.9	114	132.2	17.0	13.5%	240 376.6	243 436.1	2 415.9	1 765	1814.5	33.8	19.7%
0.01	50684.5	51 218.6	595.0	108	129.0	18.3	14.3%	240 246.0	244 788.4	3 264.7	1 764	1784.8	15.2	19.8%
0.03	50044.2	51 978.1	1 897.2	115	128.5	14.7	13.5%	239 840.2	241 328.8	2 041.9	1 752	1817.0	66.1	20.3%
0.05	50217.1	51 546.2	1 042.7	113	127.9	10.4	13.7%	240 491.1	243 441.6	2 555.1	1 754	1804.9	35.0	19.9%
0.07	49768.2	52 461.1	2 487.1	110	124.0	13.1	14.3%	238 465.8	244 127.3	3 778.0	1 754	1792.9	23.9	20.6%
0.09	50944.3	52 642.9	1 854.0	112	124.3	9.0	13.3%	240 416.8	243 654.0	4 124.5	1 775	1788.5	17.1	19.5%

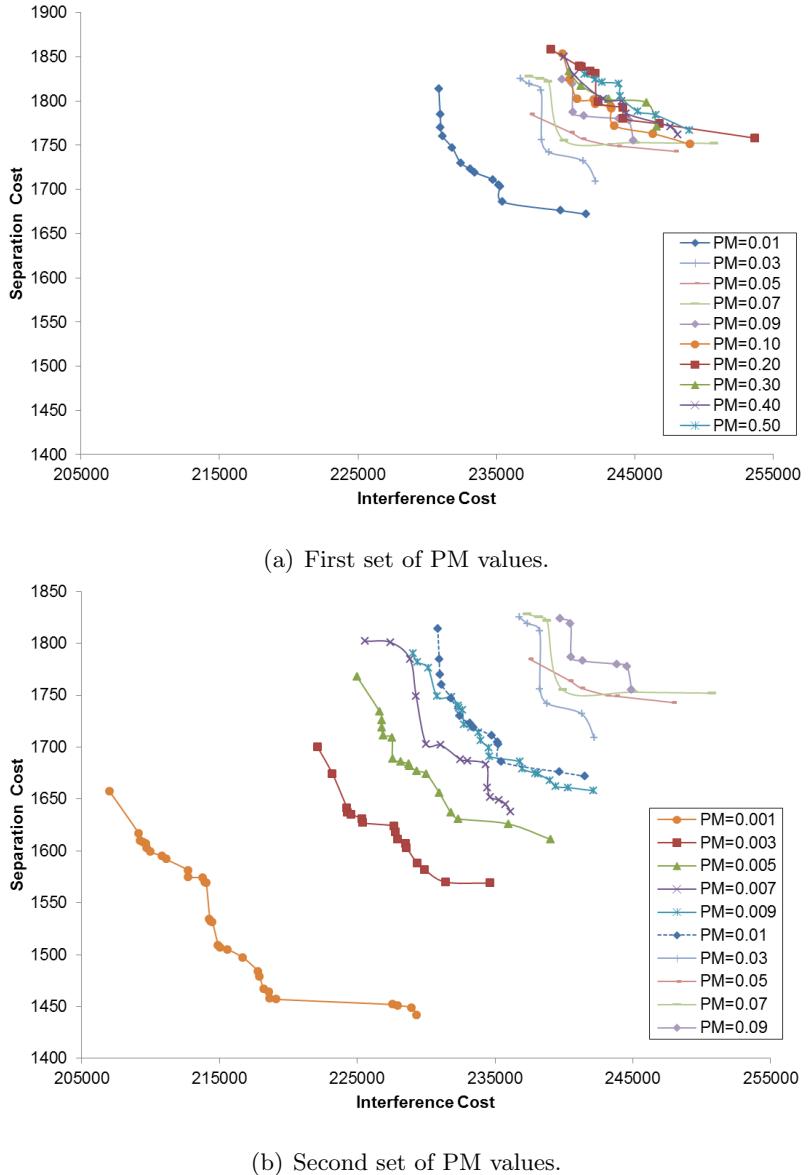


Figure 6.37: NSGA-II Pareto Fronts obtained for the mutation probability (PM) parameter values over the Denver instance.

both instances. Denver instance has obtained an HV value of 43.8% and Seattle instance obtained 62.4%, which was the higher hypervolume for both instances. In average values, the PM=0.0001 obtained a 53.1% of HV.

Thus, the conclusion obtained after all the tested PM values allowed us to clearly identify that the best value to be used in FAP for the probability of mutation is 0.01.

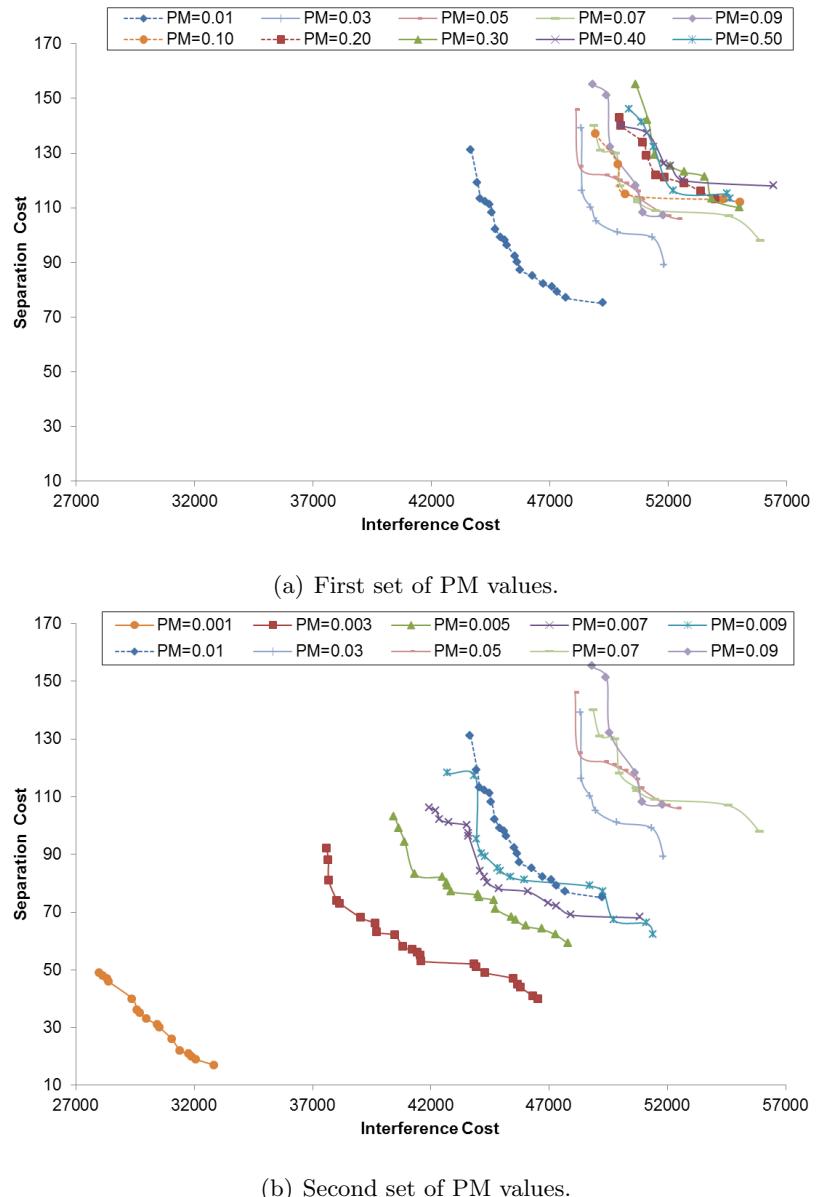


Figure 6.38: NSGA-II Pareto Fronts obtained for the mutation probability (PM) parameter values over the Seattle instance.

Crossover operator The crossover operator used initially in NSGA-II experiments was the uniform crossover (UX). In FAP the uniform crossover operator considers that every frequency of each TRX is chosen randomly from one of the two parents. The UX operator aims to accomplish a population with more diversity.

The crossover operator changes the frequencies assigned to each TRX, considering at first the values from the selected parents. The process of changing the genes is performed taking into consideration that: if the frequency that comes at random is prohibited (cost K), then it is attempted to use the frequency of the other parent. If this frequency is also prohibited, then another frequency is selected randomly within the available ones, avoiding therefore the cost K .

Besides this initial approach, several different alternatives were developed for the crossover operator. Therefore, we also tested their application in the NSGA-II. The traditional approach using the Uniform crossover (UX) was the first approach. Then, other operators that use different slice points were applied to the parents in order to generate new offspring. The several approaches tested for the crossover operator were the one-point (1X) crossover, two-points (2X), four-points (4X) and the eight-points (8X) crossover operator.

The operator showing the highest performance was the uniform crossover (UX), obtaining an average hypervolume of 53.1%. In Denver instance the hypervolume was 43.8% and for Seattle instance it was 62.4%. In Figures 6.39 and 6.40 the several Pareto Fronts are visible. From the results it is possible to visualize that crossover operator at our NSGA-II implementation for FAP did not show a significant difference on the performance regarding the type of crossover operator.

Analyzing the two distinct objectives, and considering that the interference cost is the one with an higher range of values, for both instances, the best value was obtained using the UX crossover operator. Indeed, for Denver instance the minimum value was 207 043.0 units and for Seattle instance it was 27 967.0 units. Regarding the separation cost, the UX operator was also among the best operators, obtaining for Denver instance the separation cost of 1 442 units and with Seattle instance 17 units. Therefore, taking into consideration all these different analysis indicators, the selected crossover operator was the uniform crossover operator (UX).

SPEA2 Tuning Parameters

Similar to the previous algorithm tuning experiments, also with the SPEA2, we have followed the same tuning procedure. Our intent is to assess the performance of SPEA2 and compare it to the NSGA-II. Therefore, the same algorithm oper-

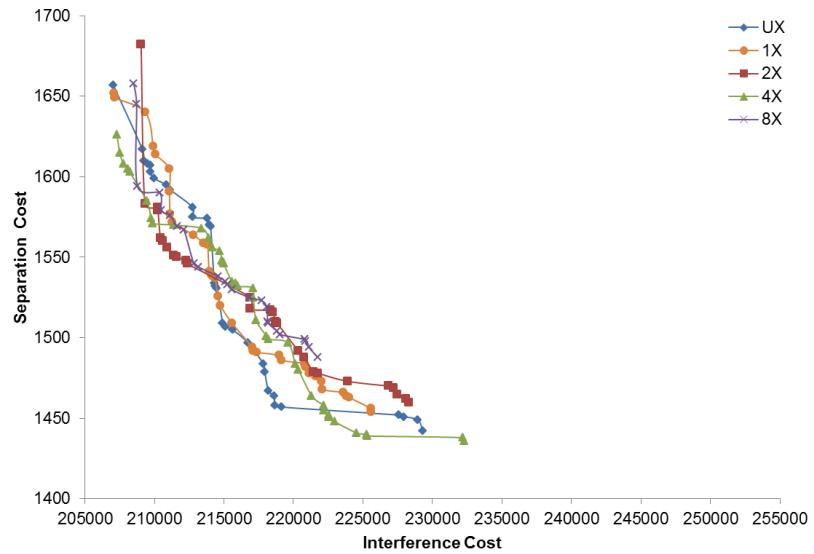


Figure 6.39: NSGA-II Pareto Fronts obtained using the several crossover operators being tested over the Denver instance.

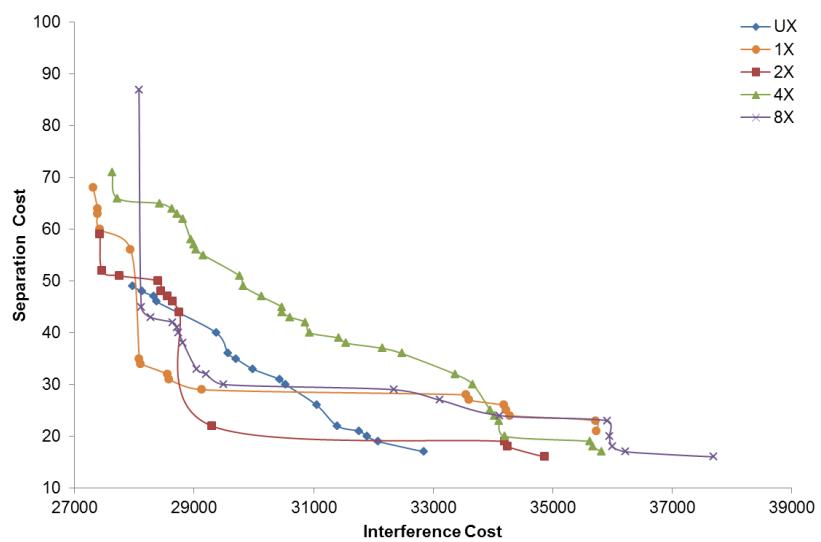


Figure 6.40: NSGA-II Pareto Fronts obtained using the several crossover operators being tested over the Seattle instance.

ators (recombination and mutation) have remained identical for both instances. The experiments with the SPEA2 parameters were done in the following order:

1. Best population size (N) value.
2. Best crossover probability (PC) value.
3. Best mutation probability (PM) value.
4. Select the most efficient crossover operator (cxOp).

SPEA2 initial configuration was set to the same parameter values as NSGA-II, which were N=100, PC=0.9, PM=0.2 and uniform crossover (UX). In SPEA2 the archive size (\bar{N}) was set to the same as population size.

Population size In SPEA2 the population size (N) is represented by a fixed value. The experiments to identify the best N value were set at the following values: 25, 50, 75, 100, 125 and 150. The obtained results for every value are presented in Table 6.19.

The best population size value was obtained by N=125 with an average hypervolume of 16.1%. The N=125 permitted to obtain the highest hypervolume of HV=13.7% for Seattle instance and for the most complex instance, which was Denver instance, it is between the highest values.

Normally, in SPEA2 it is used an archive size (\bar{N}) that is fixed. It has the same size as the population size. Our approach follows the same behavior. For that reason the \bar{N} of SPEA2 was set to 125.

Crossover probability The simulations for the crossover probability (PC) were initially performed over the following set of values: 0.1, 0.25, 0.5, 0.75, 0.9 and 1. In this set of values, results shown in Table 6.20 indicate that higher hypervolume was obtained with a PC=0.75.

Both NSGA-II and SPEA2 have similar parameters. Therefore, since we intend to compare the behavior of these two MO well-known algorithms, we have also added the range of values tested with NSGA-II. Then, the next set of experiments for PC value was set to: 0.01, 0.03, 0.05, 0.07 and 0.09. Surprisingly, the hypervolumes obtained with almost all of these past results were higher. Indeed, results have shown that the highest HV was obtained with a PC=0.01, but since it is an outlier value, we have made a deeper simulation, considering values above that number. Then, the final set of values was: 0.001, 0.003, 0.005, 0.007 and 0.009.

Table 6.19: Empirical results obtained for the SPEA2 population size (N) parameter using both instances. We present the best, average, standard deviation and the hypervolume value. The last column comprises the hypervolume (HV) mean values for every distinct N value.

N value	Seattle Instance						Denver Instance						Avg. HV		
	Interference Cost			Separation Cost			Interference Cost			Separation Cost					
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.			
25	51.993.4	53.511.9	1.091.3	113	132.1	10.9	12.3%	239.35.4	246.035.3	3.318.2	1.808.4	25.3	19.6%	15.9%	
50	51.922.1	52.688.9	666.8	112	135.0	16.8	12.6%	242.572.8	242.921.6	456.1	1.792	1.825.0	46.4	18.4%	15.5%
75	51.250.7	53.320.9	1.404.5	118	125.0	4.8	12.3%	243.042.0	244.760.7	1.998.1	1.796	1.824.8	30.9	18.1%	15.2%
100	51.435.7	52.257.4	871.3	116	129.5	11.5	12.5%	240.827.4	244.398.5	3.107.9	1.790	1.813.3	24.3	18.9%	15.7%
125	51.148.1	51.247.6	140.7	111	121.5	14.8	13.7%	242.783.6	245.168.2	1.760.8	1.782	1.808.9	27.7	18.5%	16.1%
150	50.655.3	52.558.3	1.757.4	120	135.0	12.9	12.2%	242.596.2	243.808.2	1.105.2	1.767	1.796.0	36.5	19.0%	15.6%

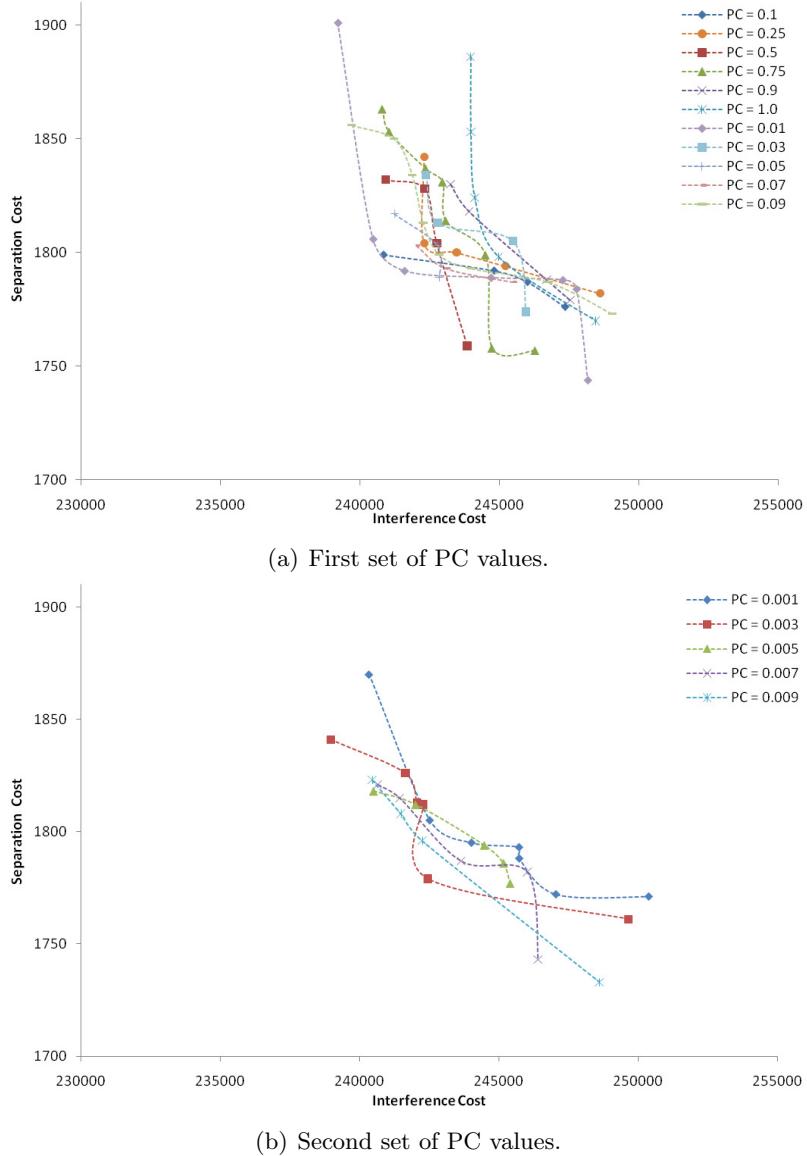


Figure 6.41: SPEA2 Pareto Fronts obtained for the crossover probability (PC) parameter values over the Denver instance.

Final simulations with the range of PC values analyzed for SPEA2 showed that the best value for this parameter is the 0.009. It has accomplished an average HV of 17.4%, with an HV of 14.4% for Seattle instance and 20.4% for Denver instance.

Mutation probability Mutation operator in SPEA2 is identical to the one implemented by NSGA-II. It assigns a random frequency value for a TRX from unused frequencies of the sector in which the TRX is installed. Then, the set of

Table 6.20: Empirical results obtained for the SPEA2 crossover probability (PC) parameter using both instances. We present the best, average, standard deviation and the hypervolume value. The last column comprises the hypervolume (HV) mean values for every distinct PC value.

PC value	Seattle Instance						Denver Instance						Avg. HV		
	Interference Cost			Separation Cost			Interference Cost			Separation Cost					
	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.	Best	Avg.	Std.			
0.1	50349.6	51383.3	1019.9	109	122.0	16.8	14.3%	240861.1	244765.8	2803.1	1776	1788.5	9.7	19.3%	15.9%
0.25	50285.7	53298.0	2747.2	113	127.6	11.8	13.2%	242306.9	244382.8	2646.1	1782	1804.4	22.6	18.7%	16.0%
0.5	50477.0	52029.8	2196.0	122	127.0	7.1	12.2%	240918.4	242461.3	1210.0	1759	1805.8	33.5	19.7%	16.0%
0.75	50571.0	52077.4	1700.3	114	129.0	10.6	13.2%	240804.7	243223.9	1879.6	1757	1814.0	40.3	19.8%	16.5%
0.9	49673.5	52256.9	2158.2	114	129.4	15.6	13.6%	243248.8	245355.7	2091.3	1779	1803.8	24.2	18.4%	16.0%
1.0	50144.6	51148.6	903.3	116	129.7	17.2	13.3%	243975.3	245103.8	1923.4	1770	1826.2	45.4	18.4%	15.9%
0.01	48752.5	52509.1	2,809.2	110	130.0	18.6	14.1%	239208.9	244172.5	3739.2	1744	1800.6	48.2	20.5%	17.3%
0.03	50299.6	50707.4	576.6	112	123.0	15.6	14.1%	242360.9	244147.5	1836.1	1774	1806.5	24.9	18.9%	16.5%
0.05	49481.6	51340.0	1404.0	118	127.8	13.4	13.4%	241249.2	242295.5	906.5	1789	1803.0	14.0	18.9%	16.2%
0.07	50647.2	51836.8	1069.1	115	115.0	8.7	13.2%	242015.7	243528.2	1777.5	1787	1794.3	8.1	18.7%	16.0%
0.09	50831.7	53565.6	2,846.7	114	125.6	14.0	13.0%	239709.6	243466.6	3066.7	1773	1813.1	30.5	19.7%	16.4%
0.001	49570.2	50415.1	940.1	119	128.7	10.0	13.4%	240338.3	245104.7	3236.2	1771	1799.1	33.6	19.5%	16.4%
0.003	50541.9	52564.7	1840.5	114	126.0	9.9	13.3%	238959.0	242837.7	3576.4	1761	1805.3	29.9	20.2%	16.8%
0.005	50461.5	51445.3	868.9	108	134.0	16.6	14.1%	240494.0	243498.0	2150.7	1777	1797.4	17.3	19.4%	16.8%
0.007	50701.8	52360.5	2295.1	111	120.8	8.1	13.7%	240649.8	243626.6	2600.2	1743	1789.6	31.1	20.2%	16.9%
0.009	49719.6	51890.4	1420.4	108	129.1	14.2	14.4%	240448.4	243191.5	3681.0	1733	1790.0	39.6	20.4%	17.4%

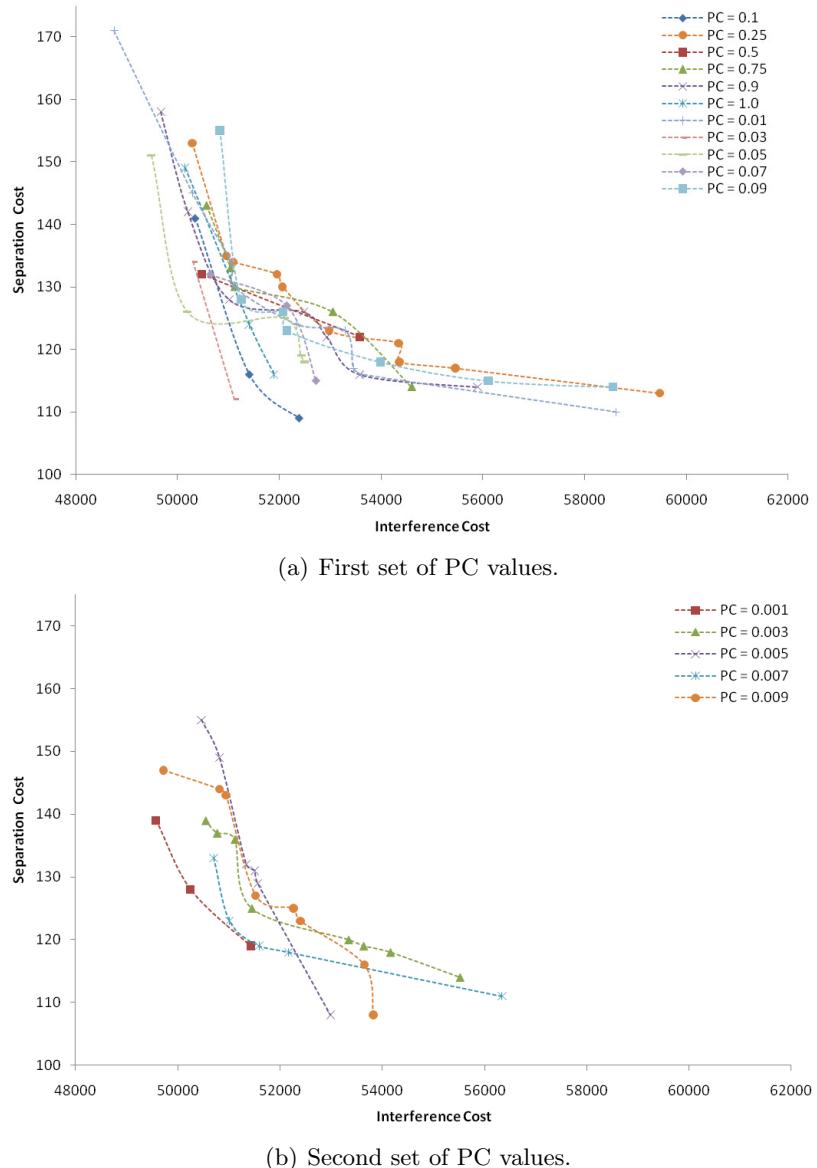


Figure 6.42: SPEA2 Pareto Fronts obtained for the crossover probability (PC) parameter values over the Seattle instance.

available frequencies of the specific sector is updated accordingly. In this context, the mutation probability (PM) controls the amount of frequencies changed in a frequency plan. Commonly, the mutation does not assume large values.

The initial set of values tested for PM was: 0.01, 0.1, 0.2, 0.3, 0.4 and 0.5. The first simulations show that with smaller PM values the higher hypervolumes were obtained. Indeed, a PM=0.01 obtained an average HV of 28.7% (29.7% for Seattle instance and 27.6% for Denver instance). Therefore, additional mutation values were analyzed namely the 0.03, 0.05, 0.07 and 0.09. Results have remained with an identical behavior. As a consequence, then a smaller magnitude of values was considered. The final set of PM values was configured to: 0.001, 0.003, 0.005, 0.007 and 0.009. Figure 6.43 and Figure 6.44 show the several Pareto Fronts obtained for every PM value.

Clearly, the best solutions are obtained using PM=0.001. It accomplished an impressive average hypervolume of 61.9% (49.8% for Denver instance and 74.1% for Seattle instance), which is significantly higher than the one accomplished with the initial tested PM values. Prior to final PM results, in the first set of PM values the highest average HV was 28.7% with PM=0.01 (29.7% for Seattle instance and 27.6% for Denver instance).

Crossover operator In SPEA2 solutions are selected as parents using a binary tournament selection and then recombination is applied to parents in order to generate a number of offspring solutions. For this task, several crossover operators were used in the implemented version of SPEA2. We tested five types of recombination: uniform crossover (UX), one-point crossover (1X), two-point crossover (2X), four-point crossover (4X) and eight-point crossover (8X).

In the uniform crossover operator (UX) every frequency of each TRX can be chosen randomly from one of the two parents. Additional, different operators were designed to FAP, which use different slice points in the generation of the offspring. The one-point crossover (1X) creates one offspring solution by joining a part of the first parent solution from the first position to a crossover point with another part of the second parent from the crossover point to the last position. The other operators use the same approach, but applying the number of parts defined by the number of slice points.

In our experiments we found that results do not differ much for every crossover operator, showing a uniform behavior. As we can see by Figure 6.45 and Figure 6.46, the Pareto Fronts for every operator are very similar. With Denver instance the highest HV was obtained with UX crossover operator (49.8%). In Seattle instance the highest HV was 75.1% using the 8X crossover operator. On average values the crossover operator with the highest HV was the 8X, with an 62.2%.

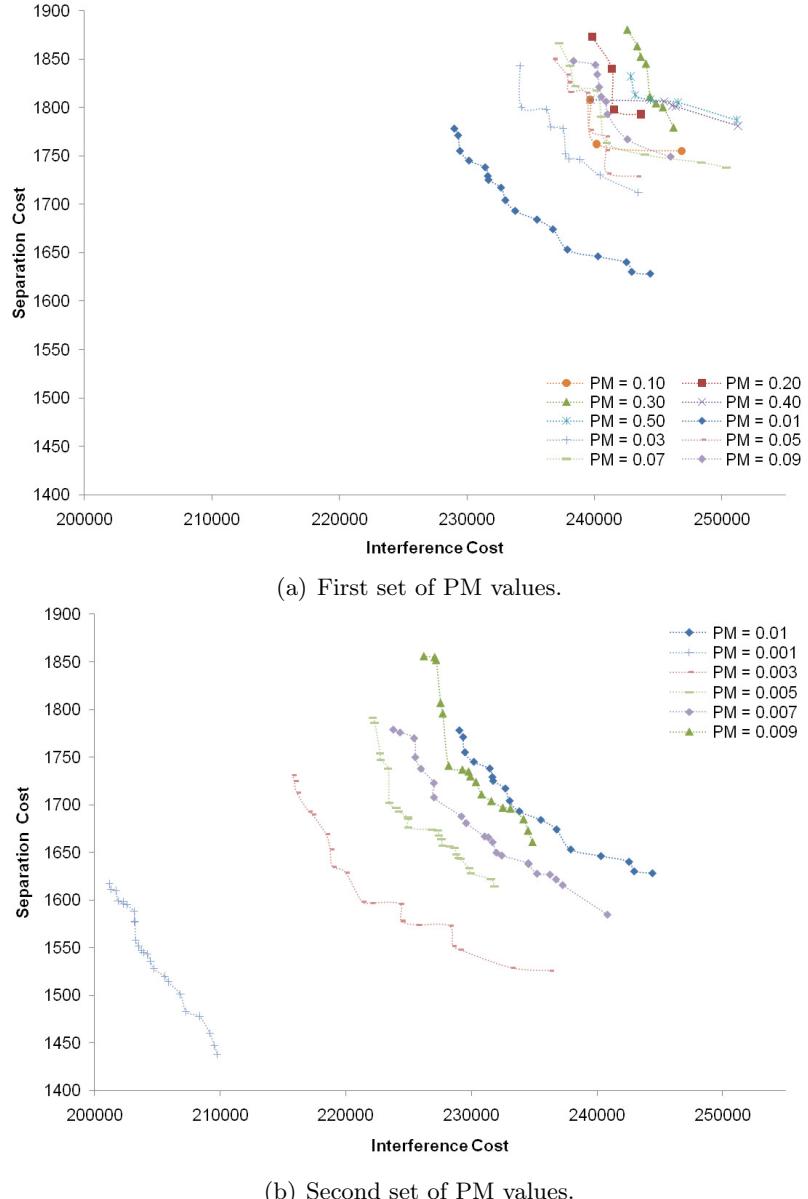


Figure 6.43: SPEA2 Pareto Fronts obtained for the mutation probability (PM) parameter values over the Denver instance.

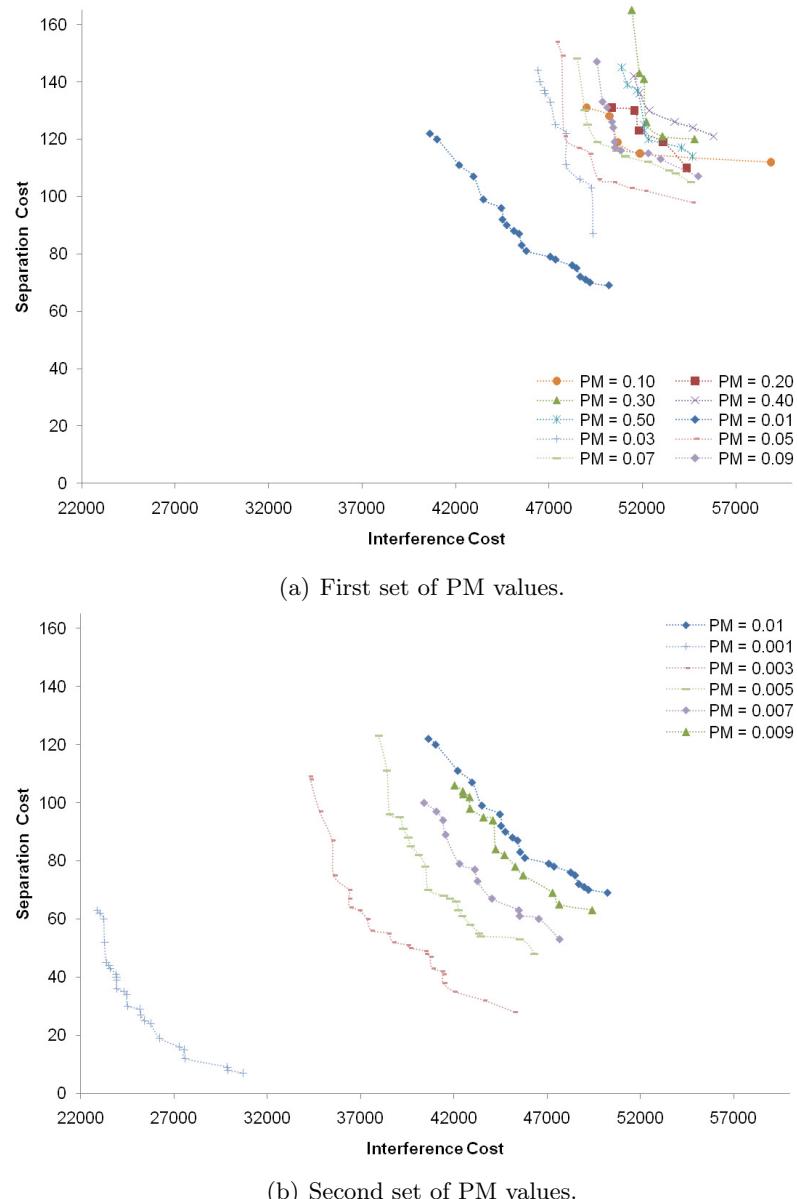


Figure 6.44: SPEA2 Pareto Fronts obtained for the mutation probability (PM) parameter values over the Seattle instance.

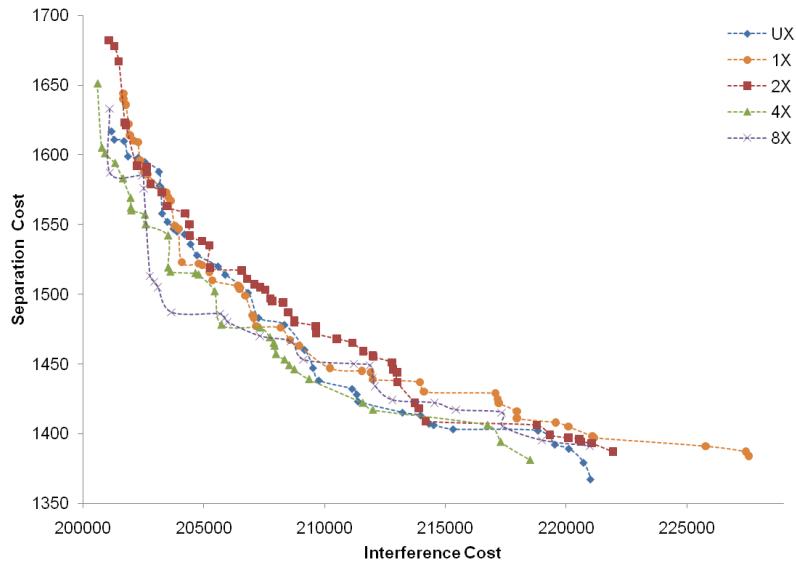


Figure 6.45: SPEA2 Pareto Fronts obtained for the crossover operator (CxOp) parameter values over the Denver instance.

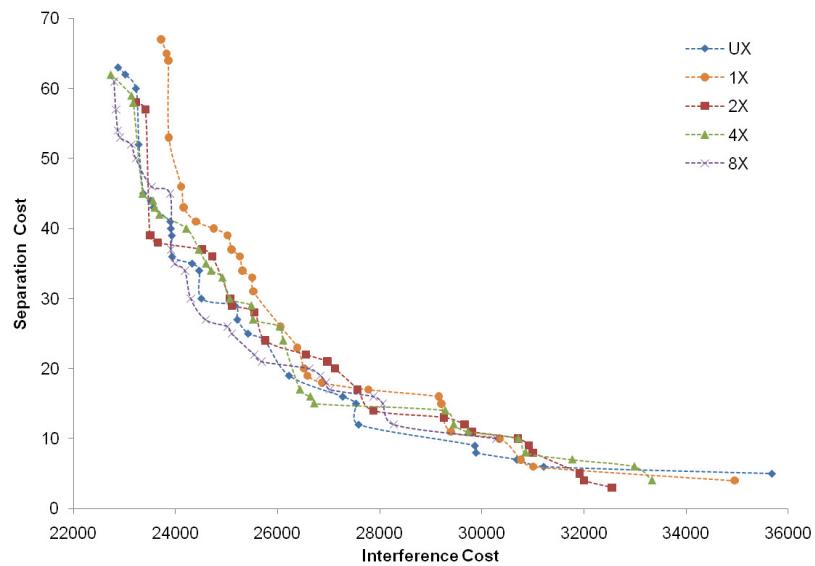


Figure 6.46: SPEA2 Pareto Fronts obtained for the crossover operator (CxOp) parameter values over the Seattle instance.

6.5.2 Parameterization

The finding of the tuning process, which was conducted for the algorithms parameters, was then applied in the final simulations. The configuration used for the algorithms was:

- **DEPT:** Population Size (NP) = 75, crossover probability (CR) = 0.01, mutation probability (F) = 0.9 and DE/Rand-to-Best/1/Exp strategy.
- **MO-VNS and GMO-VNS:** $k_{max}=9$.
- **MO-SVNS and GMO-SVNS:** $k_{max}=9$, $\alpha=0.4$.
- **MO-ABC:** colony size = 100, limit = 75 and mutation probability = 0.9.
- **NSGA-II:** Population Size (N) = 50, probability of crossover (PC) = 0.07, probability of mutation (PM) = 0.001 and uniform crossover operator (UX).
- **SPEA 2:** Population Size (N) = 125, probability of crossover (PC) = 0.009, probability of mutation (PM) = 0.001 and crossover operator (cxOp) = 8X (eight-points crossover) and Archive size (\bar{N}) = N.

6.5.3 Results Comparison Analysis

The current work aims to analyze and identify the best multiobjective metaheuristics to be applied to the multiobjective real-world formulation of FAP. Therefore, after we have designed and adapted several multiobjective metaheuristics, the next step was focused on assessing their performance. There are many versions of FAP, most of them benchmark problems, but our multiobjective formulation has proposed to reach relevant aspects for real-world GSM networks. The statistical study was focused on the two large-scale real-world instances already presented in Section 6.1.

Our contribution was focused on proposing new metaheuristics as the DEPT, the MO-VNS, MO-SVNS (and also their greedy versions - GMO-VNS and GMO-SVNS) and the MO-ABC. To enrich the comparison among our metaheuristics, and to clearly justify the good and promising results accomplished by GMO-SVNS algorithm, we have included a comparison among these algorithm results against the performance of the most well-known algorithm in multiobjective optimization, named NSGA-II and SPEA2.

In order to compare and assess the distinct performance of the above multiobjective optimization algorithms, the final set of experiments were performed using 30 independent runs (in order to assure its statistical relevance). Besides the fact that results are analyzed and discussed using the Hypervolume (HV)

indicator and the Coverage Relation, we also present common statistical analysis (using the mean and the standard deviation values).

Figures 6.47 and 6.48 depict the nondominated individuals of the eight multiobjective algorithms being compared. The final Pareto fronts were obtained using the configuration parameters presented in the subsection 6.5.2.

Next, we started by outlining our findings comparing the performance of the metaheuristics designed by us. Then, comparatives with other well-known algorithms were also considered.

Graphically, it is clearly identified that the worst approaches are the MO-VNS and the MO-SVNS algorithms. Both algorithms have obtained the smallest Pareto Fronts and have the lowest hypervolume values (27.4% for both instances). Indeed, the nondominated solutions belonging to these Pareto fronts are the ones that represent the worst values accomplished for the two objectives being minimized in the problem. The poor performance of both algorithms can be justified because the initial mutation operator is performed totally at random. At this point the trajectory-based algorithms seemed to have a weak performance addressing multiobjective formulation of FAP. Since metaheuristics can profit by incorporating additional information about the problem, we tried to develop new strategies for the trajectory-based algorithms to significantly obtain better results. A new greedy mutation operator was proposed, and results obtained by their GMO-VNS and the GMO-SVNS versions have shown that it was possible to keep improving the solution quality in the two problem instances. The same greedy mutation was also followed by MO-ABC. The values accomplished by the investigated metaheuristics for every instance are available in Table 6.21 and Table 6.22.

In Figure 6.47 and Figure 6.48 the Pareto Fronts show that GMO-SVNS results were the ones showing the best behavior. In fact, the trajectory-based metaheuristics (GMO-VNS and GMO-SVNS) are quite better than population-based metaheuristics (DEPT and MO-ABC). The GMO-SVNS has achieved a hypervolume of 56.9% for Seattle instance and 81.1% for Denver instance. The Denver HV is indeed the highest HV accomplished from all metaheuristics. GMO-SVNS was slightly better than GMO-VNS. Indeed, GMO-SVNS has a better performance in the larger and more complex instance, achieving an impressive hypervolume improvement (81.1% for the Denver instance). Regarding Seattle instance, the highest HV were obtained with the population based metaheuristics.

In the case of population-based metaheuristics, DEPT was not the best, but it is between the best metaheuristics, achieving one of the highest hypervolume (72.3%) in Seattle instance. Our other population-based metaheuristics (MO-ABC) had a similar approach, with an HV of 62.2% for Denver. Indeed,

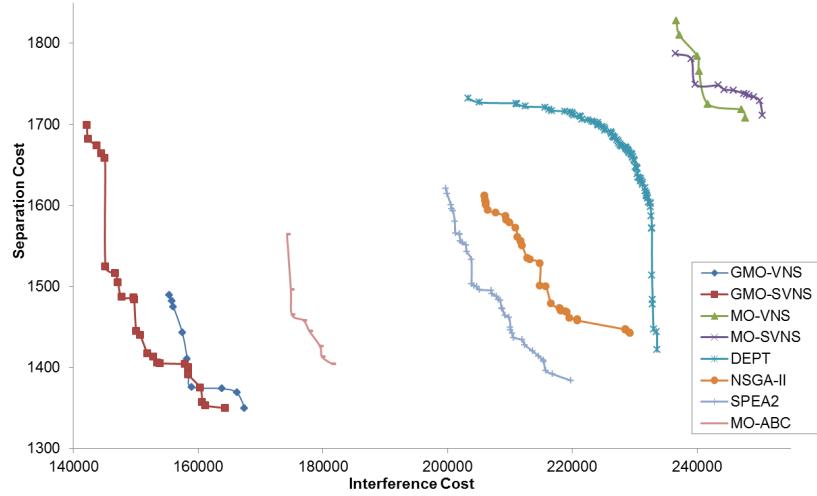


Figure 6.47: Pareto Fronts obtained using various metaheuristics over the Denver instance.

a higher hypervolume was accomplished for Denver than DEPT and the well-known NSGA-II and SPEA2. In Seattle instance the best frequency plan was obtained with SPEA2 with a HV of 74.7%. The worst frequency plan to that instance was accomplished with MO-ABC followed by NSGA-II with an average HV of 50.4% and 63.8% respectively. For Denver instance the scenario does not seem so clear, since those hypervolumes were not the best ones, but still are between the highest hypervolumes, if the trajectory-based metaheuristics were also considered. Indeed, over Denver instance the MO-ABC has surpassed the results obtained by the other population-based metaheuristics.

Overall, our designed trajectory-based algorithm (GMO-SVNS) and the population-based algorithm (DEPT) are effective algorithms to solve both problem instances. We can conclude that from our proposed algorithms, the trajectory-based algorithms were the ones that performed better.

Hypervolume Indicator

Table 6.23 reports the results of the hypervolume indicator for every algorithm solving the two problem instances of FAP.

Regarding the hypervolume indicator for DEPT, we have accomplished a final hypervolume of 72.3% for Seattle instance and 43.4% for Denver instance. DEPT accomplished an average HV of 57.9%, which is the second best HV for population-based metaheuristics. Over Denver instance, our MO-ABC also surpassed the other population-based metaheuristics, obtaining the highest hypervolume (62.2%) and with a final average performance of 56.3% HV. Comparing with the MO-VNS trajectory-based metaheuristics, the hypervolumes were 17.9%

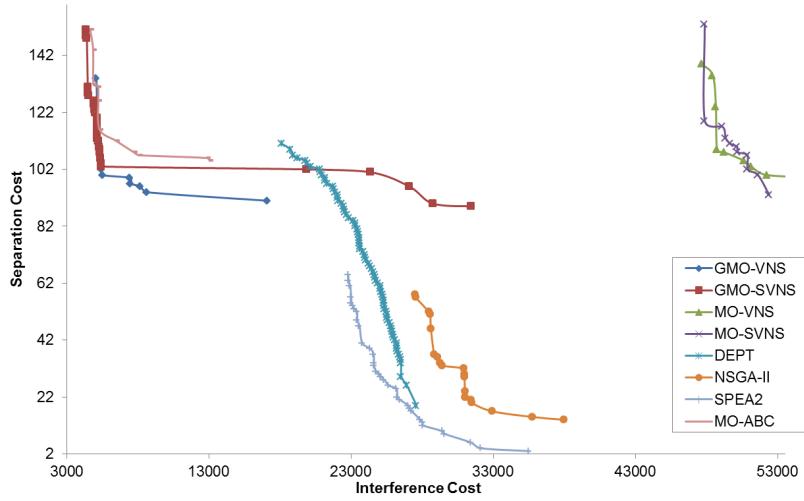


Figure 6.48: Pareto Fronts obtained using the various metaheuristics over the Seattle instance.

Table 6.21: Results obtained with the metaheuristics over the Denver instance. We present the best, average, standard deviation values and the hypervolume value.

Metaheuristics	Interference Cost			Separation Cost			HV
	Best	Avg.	Std.	Best	Avg.	Std.	
MO-VNS	236 680.1	241 531.8	4 389.0	1 708	1 762.6	47.2	22.4%
MO-SVNS	236 507.3	245 179.9	4 642.1	1 711	1 744.5	21.0	22.4%
GMO-VNS	155 359.1	159 759.1	4 420.2	1 350	1 415.9	52.2	75.0%
GMO-SVNS	142 183.4	151 579.4	6 582.2	1 350	1 480.6	113.7	81.1%
DEPT	203 283.1	226 756.2	6 368.4	1 422	1 653.0	65.8	43.4%
MO-ABC	174 231.1	177 473.9	2 688.0	1 404	1 458.9	51.9	62.2%
NSGA-II	205 897.8	214 503.7	7 129.0	1 442	1 527.9	60.7	44.4%
SPEA2	199 729.4	207 273.1	5 431.8	1 384	1 495.2	67.8	49.9%

Table 6.22: Results obtained with the metaheuristics over the Seattle instance. We present the best, average, standard deviation values and the hypervolume value.

Metaheuristics	Interference Cost			Separation Cost			HV
	Best	Avg.	Std.	Best	Avg.	Std.	
MO-VNS	47 681.1	50 424.0	2 880.9	99	113.6	15.2	17.9%
MO-SVNS	47 864.5	49 997.3	1 427.9	93	112.1	15.5	18.8%
GMO-VNS	5 069.7	7 551.1	3 612.6	91	103.0	12.3	57.5%
GMO-SVNS	4 347.8	8 265.8	8 043.7	89	118.2	16.3	56.9%
DEPT	18 123.0	23 828.3	2 191.7	19	69.8	23.5	72.3%
MO-ABC	4 636.6	7 107.5	3 112.5	105	121.6	16.2	50.4%
NSGA-II	27 469.4	30 604.6	2 675.3	14	33.1	14.1	63.8%
SPEA2	22 799.6	25 921.2	3 008.2	3	31.7	18.6	74.7%

and 22.4% for Seattle and Denver instances respectively. The hypervolumes using the MO-SVNS algorithm were 18.8% for Seattle instance and a 22.4% for Denver instance (an average hypervolume of 20.6%). Analyzing the hypervolume among these three approaches clearly shows that the DEPT algorithm has a better performance when compared with the MO-VNS and MO-SVNS algorithms. Considering the remaining algorithms, and comparing the DEPT with the greedy versions of the GMO-VNS and GMO-SVNS algorithms, the highest hypervolumes were obtained with the last two algorithms, representing a significant improvement in the final results (an average hypervolume of 66.3% and 69% respectively).

Overall, despite the fact that NSGA-II is considered one of the best multiobjective algorithms, in our scenario it does not end up being the best choice. Not only it has one of the worst hypervolumes (in average values is 54.1% against the 69.3% of the GMO-SVNS, that represents the best approach), but also regarding the average interference cost and separation cost values, this algorithm shows some worse results than DEPT and the GMO-VNS and GMO-SVNS. Indeed, for Denver instance with the GMO-VNS and the GMO-SVNS better values were obtained for the interference cost (142 183.4 units) and the separation cost (1 350 units). For the Seattle instance, results were also the same, with the best value of 4 347.8 units for the interference cost and 89 units for the separation cost. Regarding the NSGA-II algorithm, worse results were obtained: 205 897.8 for the interference cost of Denver instance and 1 442 for its separation cost. With Seattle the behavior was the same.

On the other hand, SPEA2 was the population-based metaheuristic with the best performance, with a 62.3% of average HV. It accomplished the highest HV for Seattle instance with 74.7% and 49.9% for Denver instance. Regarding Denver instance, the best performance from overall metaheuristics was indeed the trajectory-based metaheuristic GMO-SVNS (81.1%). For population based-metaheuristics, the highest HV was 62.2%, obtained with MO-ABC, followed by SPEA2 (49.9%) and then NSGA-II with 44.4%. For Seattle instance the best population-based metaheuristic was the SPEA2 with a hypervolume of 74.7%, followed by DEPT with 72.3%, NSGA-II with 63.8% and finally with MO-ABC having 50.4%.

Concluding, ranking the final average hypervolume indicator, the best population based metaheuristic was SPEA2, then DEPT with an average HV of 57.9%, followed by MO-ABC with 56.3% and finally, with the worst behavior was NSGA-II with 54.1%. Although DEPT was not the algorithm with the best performance, it has shown a very positive performance. Both greedy versions of the trajectory-based metaheuristics have the highest performance than the best population-based metaheuristics. We have shown that our proposed metaheuris-

Table 6.23: Hypervolume values obtained by the metaheuristics for both instances.

Metaheuristics	Instances		Avg. HV	Std. HV
	Seattle	Denver		
MO-VNS	17.9%	22.4%	20.2%	3.2
MO-SVNS	18.8%	22.4%	20.6%	2.5
GMO-VNS	57.5%	75.0%	66.3%	12.4
GMO-SVNS	56.9%	81.1%	69.0%	17.1
DEPT	72.3%	43.4%	57.9%	20.4
MO-ABC	50.4%	62.2%	56.3%	8.3
NSGA-II	63.8%	44.4%	54.1%	13.7
SPEA2	74.7%	49.9%	62.3%	17.5

tics (DEPT, MO-ABC and GMO-* variants) are very competitive, surpassing in some scenarios the results obtained by well-known NSGA-II and SPEA2. Thus, regarding the hypervolume, the GMO-* variants are indeed the best approach.

Coverage Relation Indicator

The existence of nondominated solutions reflects one of the most common concepts used when tackling a multiobjective optimization problem. Figures 6.47 and 6.48 show respectively the nondominated solutions accomplished by the several algorithms for Denver and Seattle instances. To confirm the conclusions previously presented, it was also used the Coverage Relation (C) metric that considers the percentage of nondominated solutions for each one of the algorithms. We emphasize that Coverage Relation is not a commutative relation, therefore the Coverage (A, B) is not the same as Coverage (B, A). In Table 6.24 several coverage relation values are presented.

Applying the coverage relation indicator, the analysis of the accomplished results clarifies the findings. Regarding the MO-VNS, MO-SVNS and their Greedy versions (GMO-VNS and GMO-SVNS), the algorithms using the greedy methods have become 100% better than the non-Greedy versions. Furthermore, it appears that GMO-SVNS is better than GMO-VNS. On the other hand, if we compare the DEPT algorithm with the GMO-SVNS and the GMO-VNS, we see that both are better than DEPT. DEPT is also surpassed by the MO-ABC in the coverage relation. The average coverage rate of MO-ABC regarding DEPT was 53.2% which was very positive. In Denver instance it has obtained a 100%, although for Seattle instance it only obtained 6.3%. Regarding NSGA-II, it becomes clear that this algorithm is not the appropriate choice, although it performs better than MO-VNS and MO-SVNS, in comparison with all the other algorithms, it is worse. The other well-known multiobjective metaheuristic, SPEA2, was the population based metaheuristic with one of the best performances. SPEA2 covers 100% of the NSGA-II solutions, but only 84.8% of DEPT solutions. Regarding

MO-ABC, the SPEA2 could not surpass the results. Comparatively to GMO-* variants it has a smaller coverage rate.

Table 6.24: Direct comparison of the results achieved by the different multiobjective algorithms implemented for the FAP. Each cell represents the percentage of nondominated solutions evolved from the algorithm Q , which are covered by the nondominated points achieved by the algorithm P in average. The column *mean* includes the mean value for each row.

		Coverage ($P \succeq Q$)		
		Algorithm		
		Denver	Seattle	Avg.
MO-VNS	MO-SVNS	75%	63.6%	69.3%
	GMO-VNS	0%	0%	0%
	GMO-SVNS	0%	0%	0%
	DEPT	0%	0%	0%
	MO-ABC	0%	0%	0%
	NSGA-II	0%	0%	0%
	SPEA2	0%	0%	0%
MO-SVNS	MO-VNS	57.1%	55.6%	56.3%
	GMO-VNS	0%	0%	0%
	GMO-SVNS	0%	0%	0%
	DEPT	0%	0%	0%
	MO-ABC	0%	0%	0%
	NSGA-II	0%	0%	0%
	SPEA2	0%	0%	0%
GMO-VNS	MO-VNS	100%	100%	100%
	MO-SVNS	100%	100%	100%
	GMO-SVNS	4.2%	12.5%	8.3%
	DEPT	100%	24.1%	62%
	MO-ABC	100%	54.5%	77.3%
	NSGA-II	100%	0%	50%
	SPEA2	100%	0%	50%
GMO-SVNS	MO-VNS	100%	100%	100%
	MO-SVNS	100%	100%	100%
	GMO-VNS	80%	20%	50%
	DEPT	100%	10.1%	55.1%
	MO-ABC	100%	100%	100%
	NSGA-II	100%	0%	50%
	SPEA2	100%	0%	50%
DEPT	MO-VNS	100%	100%	100%
	MO-SVNS	100%	100%	100%
	GMO-VNS	0%	0%	0%
	GMO-SVNS	0%	12.5%	6.3%
	MO-ABC	0%	0%	0%

Continued on next page

Table 6.24 – continued from previous page

		Coverage ($P \succeq Q$)		
		Algorithm		
		Problem Instances		
P	Q	Denver	Seattle	Avg.
	NSGA-II	0%	84.2%	42.1%
	SPEA2	0%	0%	0%
MO-ABC	MO-VNS	100%	66.7%	83.3%
	MO-SVNS	100%	72.7%	86.4%
	GMO-VNS	0%	10%	5.0%
	GMO-SVNS	0%	0%	0%
	DEPT	100%	6.3%	53.2%
	NSGA-II	100%	0%	50%
	SPEA2	92.5%	0%	46.3%
NSGA-II	MO-VNS	100%	100%	100%
	MO-SVNS	100%	100%	100%
	GMO-VNS	0%	0%	0%
	GMO-SVNS	0%	6.3%	3.1%
	DEPT	96.5%	0%	48.2%
	MO-ABC	0%	0%	0%
	SPEA2	0%	0%	0%
SPEA2	MO-VNS	100%	100%	100%
	MO-SVNS	100%	100%	100%
	GMO-VNS	0%	0%	0%
	GMO-SVNS	0%	12.5%	6.3%
	DEPT	100%	69.6%	84.8%
	MO-ABC	0%	0%	0%
	NSGA-II	100%	100%	100%

6.5.4 Multiobjective Conclusions

In this section several multiobjective metaheuristics have been analyzed to address a real-world FAP. A comparison between well-known multiobjective algorithms, named NSGA-II and SPEA2, was performed against our recently proposed multiobjective metaheuristics. These approaches were the DEPT algorithm, the GMO-VNS algorithm and its variant GMO-SVNS algorithm (and their non-greedy versions - MO-VNS and MO-SVNS) and finally the MO-ABC algorithm. To assess the performance of the metaheuristics and compare their results, two important indicators have been used: the hypervolume and the coverage relation, under two large-scale real-world GSM networks.

Firstly, a thorough parametric study has been carried out to determine the optimal values for the presented algorithms. After finding the best parameter settings, the results with final algorithms showed that the best metaheuristics are

the greedy versions (GMO-VNS and GMO-SVNS) developed for the MO-VNS and MO-SVNS algorithms. They clearly outperform all the other approaches. The algorithm that obtains the highest hypervolume is the GMO-SVNS, with an average value of 69%. The NSGA-II only obtains 54.1%, MO-ABC 56.3% and SPEA2 62.3%, therefore they have a worse performance. Regarding the coverage relation indicator, the conclusions are the same. Indeed, it was possible to identify that NSGA-II can only get better nondominated solutions than the non-greedy versions of MO-VNS and MO-SVNS algorithms, and for all the other algorithms it has a worse behavior. The poor performance of NSGA-II confirms that GMO-VNS and GMO-SVNS are indeed the best algorithms to tackle the large-scale real-world instances of FAP in GSM networks. SPEA2 has shown a better performance than NSGA-II, but it still does not surpass the excellent performance of GMO-SVNS.

Summarizing all the above, we can conclude that DEPT, MO-ABC, GMO-SVNS and GMO-VNS outperform the NSGA-II algorithm, one of the most common algorithms in multiobjective optimization. On the other hand, when comparing both well-known multiobjective metaheuristics (SPEA2 and NSGA-II), the SPEA2 performs better than NSGA-II and also than our proposed algorithms (DEPT and MO-ABC). Additionally, GMO-SVNS yields better results than GMO-VNS and DEPT and MO-ABC. Results have shown that the best metaheuristics are the two trajectory-based algorithms, named GMO-SVNS and GMO-VNS algorithms. They clearly outperform all the other approaches. As a main conclusion, we can remark that our proposed metaheuristics can be very effective and can be successfully applied in the design of the frequency plan required by a GSM network.

6.6 Summary

Throughout this chapter, results assessing the performance of every algorithm to solve real-world instances of FAP were given. Comparisons about the performance of each one were made. Results were presented considering the two distinct classes of algorithms: the single-objective algorithms and the multiobjective algorithms. Results were validated using statistical indicators appropriated to every type of algorithm.

Normally algorithms may require some serious parameter tuning to achieve the best performance for some problem. In our case, we have conducted a preliminary experiments phase with every metaheuristics to identify the best parameter settings to be used in the final experiments. These results were also given and afterward, the final comparisons were shown and some conclusions were presented.

The assessment of algorithms was based mainly on the evaluation of the statistical performance indicators that we carefully have selected. We followed some of the most common approaches used in the comparison of algorithms performance.

The main results obtained with the metaheuristics applied to FAP instances are the following:

- Through simulations, we found that with a single-objective FAP optimization problem, the best performance was accomplished with the SVNS algorithm, which is a trajectory-based algorithm.
- The incorporation of some domain knowledge is extremely important in the performance of the algorithms. In our case, results could be improved by adding some mechanism that avoids some of the most costly penalties in a frequency plan. The incorporation of problem domain information was a common procedure followed by some of the algorithms that we have designed and adapted. In fact, this approach has led to new hybrid versions of some already known algorithms.
- On the other hand, we also contributed with the formulation of a multi-objective version of FAP. In this area, we have designed and adapted new multiobjective metaheuristics. They were based on some single-objective metaheuristics that have proved to perform well when addressing FAP. We have adapted them to handle multiobjective problems.
- The metaheuristics that we proposed were validated against two well-known multiobjective algorithms. Indeed, by the conducted experiments, we identified that best results were obtained using our proposed multiobjective metaheuristics.

- In the case of the multiobjective version of FAP, the best results were obtained by GMO-SVNS.

The current work has an experimental nature that is based on the accomplished results performed to assess the efficiency of the proposed algorithms; it suggested that trajectory-based algorithms have achieved a better performance than the population-based algorithms. The multiobjective version of FAP also reflects this same conclusion.

Finally, we finishing by standing out that as demonstrated by the results presented in this chapter, the metaheuristics proposed by us to solve a multi-objective formulation of FAP have the best performance, when compared with other already known metaheuristics.

Conclusions and Future Work

In this chapter, we outline the major achievements of our investigation and present some future working areas that can be considered for subsequent investigations in FAP on GSM networks. Finally, a summarized list of scientific production related to the current work is also presented.

7.1 Main Achievements in the Investigation

For NP-hard optimization problems and complex search problems, the metaheuristic methods have proved to be particularly effective. FAP is under this class of problems; therefore, based on this reality we have proposed new metaheuristics, as well as some hybrid versions of already known metaheuristics. Our work was based on two distinct optimization approaches: the single-objective metaheuristics and the multiobjective metaheuristics. Indeed, the new proposed metaheuristics were designed specifically for the scenario using a multiobjective formulation of FAP. Additionally, our contribution addressing FAP was also given by the incorporation of some hybrid approaches on the basic structure of the classic metaheuristics that were used to address FAP in real GSM networks. The hybrid metaheuristics were mainly developed under the single-objective formulation of FAP.

When solving FAP, the common goal of the various algorithms is to achieve a valid frequency plan, provoking a very low interference level in the network, thus having the smallest number of interferences possible. This is the most important issue when evaluating the quality of a frequency plan. Indeed, to a mobile operator, the level of interference influences directly the quality of service provided by

the operator to the end user. Our experiment results have shown that due to the hard computationally complexity of the problem, reaching high quality solutions in short time executions is very hard. The instances complexity is also a factor to be considered because it also influences the final results of the algorithms. Therefore, the adequate tuning of the parameters is also a major concern because it has a direct influence on the performance of the metaheuristics. The accomplishment of frequency plans without any interference is not viable, therefore a high quality plan needs to be considered in long and short term executions. In our study we took into consideration all these important issues. To every proposed metaheuristics two sequential tasks were performed. First, the tuning of the algorithm parameters. Then the final set of experiments.

Driven by the mathematical formulation that makes use of the real information and characteristics of FAP for GSM networks, we have started the research by considering the problem of optimizing FAP only as a single-objective optimization problem. Under this perspective, we have made several experimental implementations of three distinct metaheuristics: Differential Evolution (DE), Variable Neighborhood Search (VNS) and Skewed Variable Neighborhood Search algorithm (SVNS). For these metaheuristics we have enhanced the study by incorporating some hybrid characteristics on their basic structure, which have improved the final performance of the metaheuristics in addressing FAP. To the best of our knowledge, these metaheuristics had not been previously applied to real-world FAP instances, which we have used.

In the single-objective approach of FAP, we made a comparison among the most relevant metaheuristics used to solve this instance of the FAP problem in a GSM Network. As a conclusion, we can say that Skewed Variable Neighborhood Search (SVNS) algorithm is among the algorithms that obtained the best results.

Further, following the single-objective optimization of FAP, our next goal was to tackle the frequency planning as a multiobjective optimization problem. As already mentioned, it corresponds to one of the major contributions of our work to optimize FAP. In this new problem approach we have proposed several new multiobjective metaheuristics and tried to identify the ones that had the best performance.

In this phase of our research, where we have focused on the multiobjective approach of FAP, we also adopted the complex and at the same time realistic approach that makes use of the same realistic and accurate interference information from real-world GSM networks. The new approaches that we proposed consisted in formulating the GSM concepts under a FAP multiobjective perspective. Indeed, one of the main contributions of our research was the multiobjective formulation of FAP for the GSM instances adopted in our study. Nowadays, the

same instances are also used by other research group members and therefore our current work can be considered in potential future research lines.

More precisely, to improve the efficiency of the overall implemented metaheuristics (including the new ones, developed by us) we have incorporated in the optimization process some additional methods to improve the efficiency of the studied algorithms. These new hybrid approaches were mainly focused on the specific characteristics of the FAP instance being used. However, although these techniques are specific for the mathematical formulation that we use, the main concept and general approach can be generalized to other engineering problems.

To summarize, our major contribution when tackling FAP as a multiobjective optimization problem was the presentation and comparison of eight different algorithms: DEPT, MO-VNS, MO-SVNS, GMO-VNS, GMO-SVNS, MO-ABC and the NSGA-II and SPEA2. Indeed, six of these algorithms correspond to new contributions on the metaheuristics field, which were developed by us. More specifically, the Differential Evolution with Pareto Tournaments (DEPT), Multiobjective Variable Neighborhood Search (MO-VNS), Multiobjective Skewed Variable Neighborhood Search (MO-SVNS), Greedy Multiobjective Variable Neighborhood Search (GMO-VNS), Greedy Multiobjective Skewed Variable Neighborhood Search (GMO-SVNS) and the Multiobjective Artificial Bee Colony. The other algorithms, namely the Nondominated Sorting Genetic Algorithm - II (NSGA-II) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2), have been used in order to compare the new developed approaches with well-known algorithms in the multiobjective area. The comparison shows that our new proposed approaches are very competitive and they can obtain very good results. Therefore, solving FAP with a multiobjective formulation has also succeeded.

Furthermore, in order to do a fair comparison, it was also done a deep study of the best configuration parameters for the several studied metaheuristics, which were also performed to this specific problem (Frequency Assignment Problem, FAP). This is another contribution, the correct configuration of the metaheuristics for the FAP in GSM networks. The tuning process of the several metaheuristics was a common approach that was followed both by the single-objective optimization metaheuristics and by the multiobjective optimization metaheuristics.

One major finding of the empirical investigation is the strong influence of parameter tuning of the metaheuristic. Also, the incorporation of local search optimization methods and the inclusion of domain knowledge information on the basic structure of metaheuristics regarding some features of this specific optimization problem have led to a significant improvement in the solutions quality. Comparing the trajectory-based metaheuristics against the population-based metaheuristics, the first ones have shown to perform better with the single-objective and multiobj-

jective optimization. Furthermore, the trajectory-based metaheuristics shown to be more competitive in converging towards the nondominated solutions (Pareto Fronts), obtaining the highest hypervolume and coverage relation rates.

7.2 Future Work

Following the research work described in this thesis, a number of future research lines could be taken up, involving the proposed metaheuristics and also the real formulation of FAP for GSM networks. Indeed, much work can still be done in the optimization of FAP. In this section, we briefly summarize some of the open issues and directions that can be followed in this research field.

Future work includes the possibility of making more comparisons among other multiobjective algorithms. Furthermore, the inclusion of more real-world instances of GSM networks will be a possibility.

In the field of metaheuristics, which is one of our main contributions using the FAP scenario, a possible future line of work includes the development of more metaheuristics to optimize this real-world problem. The main aim will be the study of other evolutionary algorithms, which can be developed under the two classes of metaheuristics: trajectory-based metaheuristics and population-based metaheuristics. Embracing a wider range of distinct metaheuristics will potentiate the accomplishment of a deeper analysis.

Within a multiobjective approach, FAP can also be investigated under different multiobjective formulations.

To assess the performance of metaheuristics it would also be interesting to consider other different performance indicators to investigate the success and ability of the metaheuristics to optimize FAP.

Based on the carried experiments it would also be interesting to continue investigating the possibility of developing specific multiobjective metaheuristics under a possible parallel processing. Using clusters and grid computing in order to speed up all the experiments scenarios can also represent a very interesting approach.

An interesting project could be the development of a framework to incorporate all the metaheuristics that have been recently investigated with the mathematical formulation that we have followed. It could also accommodate a graphical representation of the solution, and export the final frequency plans to a standard format that could then be used directly by the mobile operators.

7.3 Publication Record and Scientific Merits

Several publications related to the subject of this dissertation have been published in international journals and conferences as follows:

Impact Factor Journals

1. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Comparative Analysis of a Hybrid DE Algorithm with the VNS Algorithm and its Variation SVNS to Solve a Real-World Frequency Assignment Problem. *Applied Artificial Intelligence*, Taylor & Francis, 2011, Volume 25, Issue 3, pp:217-234, ISSN:0883-9514. (*Impact factor = 0.580, Quartile = Q3*)
2. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Multiobjective metaheuristics for frequency assignment problem in mobile networks with large-scale real-world instances, 2010. (*Submitted to Engineering Computations Journal, Impact factor = 0.651, Quartile = Q3*)
3. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: A new multiobjective artificial bee colony algorithm to solve a real-world frequency assignment problem, 2011. (*Submitted to Neural Computing and Applications Journal, Impact factor = 0.812, Quartile = Q4*)

International Book Chapters

4. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Application of Differential Evolution to a Multi-objective Real-world Frequency Assignment Problem. *Differential Evolution in Electromagnetics*, Springer Berlin Heidelberg, 2010, 4, 155-176. ISBN: 978-3-642-12869-1, http://dx.doi.org/10.1007/978-3-642-12869-1_7.

LNCS

5. Chaves-González, J.M., Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Comparing Hybrid Versions of SS and DE to Solve a Realistic FAP Problem. Volume 5271/2008 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2008, pp: 257-264.
6. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Parameter Analysis for Differential Evolution with Pareto Tournaments in a Multiobjective FAP. Volume 5788/2009 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2009, pp: 799-806.

IEEE Conferences

7. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Solving the Frequency Assignment Problem with Differential Evolution. Software, Telecommunications and Computer Networks, 2007. Soft-COM 2007. 15th International Conference on (Sept. 2007) 1-5
8. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Analysis of Parameter Settings for Differential Evolution Algorithm to Solve a Real-World Frequency Assignment Problem in GSM Networks. In: The Second International Conference on Advanced Engineering Computing and Applications in Sciences, 2008. ADVCOMP'08. (October 2008) 77-82
9. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: A Hybrid Differential Evolution Algorithm to Solve a Real-World Frequency Assignment Problem. In: International Multiconference on Computer Science and Information Technology (IMCSIT 2008), Polskie Towarzystwo Informatyczne (Poland). Volume 3, IEEE Catalog Number (USA) CFP0864E-CDR (2008) 201-205
10. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Multiobjective Frequency Assignment Problem using the MO-VNS and MO-SVNS algorithms, IEEE (2009) World Congress on Nature and Biologically Inspired Computing (NaBIC'09), pp. 221-226, 9-11 Dec. 2009, Coimbatore, India.

Other International Conferences

11. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: O Algoritmo de Evolucao Diferencial como Abordagem de Resolucao do FAP, Proceedings Conferencia Ibero-Americana InterTIC, M. Muñoz, A. Freitas, P. Cravo (eds.), International Association for the Scientific Knowledge (2007) 173-178

National Conferences

12. Maximiano, M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: El Algoritmo Evolución Diferencial aplicado al Problema de la Asignación de Frecuencias, Actas de las I Jornadas sobre Algoritmos Evolutivos y Metaheurísticas (JAEM07), Thomson (2007) 81-88

7.4 Scientific Events

I have been a reviewer or member of the technical program committee for the following impact factor journals and international conferences:

- The Third International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2009), October 11- 16, 2009 - Sliema, Malta. See <http://www.iaria.org/conferences2009/ComADVCOMP09.html> (Technical Program Committee).
- The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2010), October 25- 30, 2010 - Florence, Italy. See <http://www.iaria.org/conferences2010/ADVCOMP10.html> (Technical Program Committee).
- Optimization Letters Journal, Springer. Impact factor = 0.926, Quartile = Q2. (Reviewer).
- The 11th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL2010). University of the West of Scotland from 1st to 3rd September 2010. See <http://ideal2010.ucc.ie/index.html>. (Program committee of the 11th IDEAL).
- The 12th International Conference on Intelligent Data Engineering and Automated Learning (IDEAL2011). IDEAL 2011, will be held on 7th-9th September 2011 in the University of East Anglia, Norwich, UK. See <http://www.uea.ac.uk/ideal2011>. (Program committee of the 12th IDEAL).
- The Fifth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2011), November 20-25, 2011 - Lisbon, Portugal. See <http://www.iaria.org/conferences2011/ADVCOMP11.html>. (ADVCOMP 2011 Technical Program Committee)

Bibliography

- [1] K. Aardal, A. Hipolito, v. Hoesel, C.P.M., and B. Jansen. A branch-and-cut algorithm for the frequency assignment problem. Research Memoranda 007, Maastricht : METEOR, Maastricht Research School of Economics of Technology and Organization, 1996. URL <http://ideas.repec.org/p/dgr/umamet/1996007.html>. [cited at p. 27]
- [2] K. Aardal, C. van Hoesel, A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for the frequency assignment problem. Technical report, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, Maastricht University,, 2000. URL <citeseer.ist.psu.edu/aardal01models.html>. [cited at p. 11, 14, 25]
- [3] K. I. Aardal, C. P. M. van Hoesel, A. M. C. A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for the frequency assignment problem. *4OR*, 1(4):261–317, 2003. URL <http://fap.zib.de/>. [cited at p. 8, 24]
- [4] E. Alba and J. F. Chicano. Evolutionary algorithms in telecommunication. In *Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference (MELECON 2006)*, Benalmádena, Málaga, Spain, May 16-19 2006. [cited at p. 28, 29]
- [5] S. M. Allen, S. Hurley, D. H. Smith, and S. U. Thiel. Using lower bounds in minimum span frequency assignment. In S. V. et al., editor, *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 191–204. Kluwer, 1999. [cited at p. 26]
- [6] E. Amaldi, A. Capone, F. Malucelli, and C. Mannino. Optimization problems and models for planning cellular networks. In *Handbook of Optimization in Telecommunication*, Eds. M. Resende and P. Pardalos. Springer Science, 2006. URL <http://citeseerkx.ist.psu.edu/viewdoc/summary?doi=10.1.1.80.9986>. [cited at p. 1, 10, 25]
- [7] A. Avenali, C. Mannino, and A. Sassano. Minimizing the span of d -walks to compute optimum frequency assignments. *Mathematical Programming*, 91(2):357–374, 2002. URL <http://link.springer.de/link/service/journals/10107/bibs/2091002/20910357.htm>. previously published as techincal report 04-00, DIS-Università di Roma 'La Sapienza', Rome, Italy. [cited at p. 27]
- [8] B. Babu and A. M. Gujarathi. Multi-objective differential evolution (mode) algorithm for multi-objective optimization: parametric study on benchmark test

- problems. *Journal on Future Engineering and Technology*, 3(1):47–59, 2007. [cited at p. 29]
- [9] B. Babu and M. Jehan. Differential evolution for multi-objective optimization. In *The 2003 Congress on Evolutionary Computation (CEC '03)*, volume 4, pages 2696–2703, Dec. 2003. doi: 10.1109/CEC.2003.1299429. [cited at p. 29]
- [10] T. Bäck, D. Fogel, and Z. Michalewicz. *Handbook of evolutionary computation*. Oxford Univ. Press, 1997. [cited at p. 27]
- [11] R. Battiti, A. Bertossi, and D. Cavallaro. A randomized saturation degree heuristic for channel assignment in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 50(2):364–374, March 2001. [cited at p. 24, 27]
- [12] D. Beckmann and U. Killat. A new strategy for the application of genetic algorithms to the channel-assignment problem. *IEEE Transactions on Vehicular Technology*, 48(4):1261 –1269, July 1999. ISSN 0018-9545. doi: 10.1109/25.775374. [cited at p. 24, 27]
- [13] D. Beckmann and U. Killat. Frequency planning with respect to interference minimization in cellular radio networks. Technical Report TD(99) 032, COST 259, Vienna, Austria, Jan. 1999. [cited at p. 25, 27]
- [14] A. A. Bertossi, C. M. Pinotti, and R. B. Tan. Efficient use of radio spectrum in wireless networks with channel separation between close stations. In *Proceedings of the 4th international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '00, pages 18–27, New York, NY, USA, 2000. ACM. ISBN 1-58113-301-4. doi: <http://doi.acm.org/10.1145/345848.345853>. [cited at p. 25]
- [15] C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.*, 35(3):268–308, 2003. ISSN 0360-0300. doi: <http://doi.acm.org/10.1145/937503.937505>. [cited at p. 34, 35]
- [16] R. Borndörfer, A. Eisenblätter, M. Grötschel, and A. Martin. Frequency assignment in cellular phone networks. *Annals of Operations Research*, 76:73–93, 1998. ISSN 0254-5330. URL <http://dx.doi.org/10.1023/A:1018908907763>. 10.1023/A:1018908907763. [cited at p. 11, 25]
- [17] F. Box. A heuristic technique for assigning frequencies to mobile radio nets. In *IEEE Transactions on Vehicular Technology*, volume 27, pages 57–64, May 1978. [cited at p. 27, 28]
- [18] L. T. Bui, D. Essam, H. A. Abbass, and D. G. Green. Performance analysis of evolution multi-objective optimisation algorithms in noisy environments. *Complexity International*, 11:29–39, 2005. [cited at p. 73]
- [19] B. Cabon, S. D. Givry, L. Lobjois, T. Schiex, and J. P. Warners. Benchmarks problems: Radio link frequency assignment. *Constraints*, 4:79–89, 1999. URL <http://ipsapp008.lwwonline.com/content/getfile/4611/7/1/abstract.htm>. [cited at p. 25]

- [20] V. Cerny. A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. *J. Optimization Theory and Appl.*, 45:41–51, 1985. [cited at p. 34]
- [21] G. Chakraborty. A genetic inspired algorithm with greedy mutation for channel assignment problem in cellular radio networks. In *Proceedings of the IEEE sponsored International Conference on Information Technology (ICIT2000), India*, pages 289–292, December 2000. [cited at p. 27]
- [22] G. Chakraborty. An efficient heuristic algorithm for channel assignment problem in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 50(6):1528–1539, Nov. 2001. ISSN 0018-9545. doi: 10.1109/25.966583. [cited at p. 24, 27]
- [23] G. Chakraborty and B. Chakraborty. A genetic algorithm approach to solve channel assignment problem in cellular radio networks. *IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications*, June 16-18 1999. [cited at p. 27]
- [24] G. Chakraborty and B. Chakraborty. Genetic algorithm to solve channel assignment problem in cellular radio networks. In *Proceedings of the IEEE Workshop on Soft Computing Methods in Industrial Applications, Finland*, pages 34–39, June 1999. [cited at p. 27]
- [25] P. T. H. Chan, M. Palaniswami, and D. Everitt. Neural network-based dynamic channel assignment for cellular mobile communication systems. *IEEE Transactions on Vehicular Technology*, 43(2):279–287, May 1994. ISSN 0018-9545. doi: 10.1109/25.293646. [cited at p. 27]
- [26] J. M. Chaves-González, M. A. Vega-Rodríguez, D. Domínguez-González, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. SS vs PBIL to solve a real-world frequency assignment problem in GSM networks. In *Proceedings of the 2008 conference on Applications of evolutionary computing. EvoWorkshops, Evo'08*, pages 21–30, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 3-540-78760-7, 978-3-540-78760-0. URL <http://portal.acm.org/citation.cfm?id=1787943.1787947>. [cited at p. 25, 29, 34, 47, 107, 108]
- [27] J. M. Chaves-González, M. Maximiano, M. Ángel Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. Comparing hybrid versions of SS and DE to solve a realistic fa problem. In *3th International Workshop on Hybrid Artificial Intelligence*, volume 5271/2008 of *Lecture Notes in Computer Science*, pages 257–264. Springer Berlin / Heidelberg, 2008. ISBN 978-3-540-87655-7. doi: 10.1007/978-3-540-87656-4_32. URL <http://www.springerlink.com/content/k262813394763270>. [cited at p. 25, 29, 34, 47, 105]
- [28] M. o. I. Chiu Y. Ngo and F. o. I. Victor O. K. Li. Fixed channel assignment in cellular radio networks using a modified genetic algorithm. *IEEE Transactions on Vehicular Technology*, 47(1), February 1998. [cited at p. 12, 24, 25, 27]
- [29] W. Cromptson, S. Hurley, and N. M. Stephens. A parallel genetic algorithm for frequency assignment planning. In *Proceedings IMACS/IEEE Conference on Signal Processing, Robotics and Neural Networks*, pages 81–84, Lille, France, 1994. [cited at p. 27, 28]

- [30] M. Cuppini. A genetic algorithm for channel assignment problems. *European Transactions on Telecommunications*, 5(2):285–294, 1994. ISSN 1541-8251. doi: 10.1002/ett.4460050219. URL <http://dx.doi.org/10.1002/ett.4460050219>. [cited at p. 27]
- [31] L. J. de Urries, M. A. D. Guerra, and I. Berberana. Frequency planning using simulated annealing. Technical Report COST 259, TD(00) 054, COST 259, Bergen, Norway, 2000. [cited at p. 25]
- [32] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6 (2):182–197, Apr 2002. ISSN 1089-778X. doi: 10.1109/4235.996017. [cited at p. 70, 71, 72, 73]
- [33] R. Dorne and J.-K. Hao. Constraint handling in evolutionary search - a case study of frequency assignment. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, volume 1141 of *Lecture Notes in Computer Science*, pages 801–810. Springer Berlin/Heidelberg, 1996. URL http://dx.doi.org/10.1007/3-540-61723-X_1043. [cited at p. 9]
- [34] R. Dorne and J. kao Hao. An evolutionary approach for frequency assignment in cellular radio networks. In *IEEE International Conference on Evolutionary Computation (ICEC'95)*, volume 2, pages 539–544, 1995. [cited at p. 11, 27, 38]
- [35] M. Duque-Antón and D. K. amd B. Ruber. Static and dynamic channel assignment using simulated annealing. In B. Yuhas and N. Ansari, editors, *Neural Networks in Telecommunications*, chapter 10, pages 191–210. Kluwer Academic Publishers, Boston, 1994. [cited at p. 27]
- [36] M. Duque-Antón, D. Kunz, and B. Ruber. Channel assignment for cellular radio using simulated annealing. *IEEE Transactions on Vehicular Technology*, 42(1):14–21, February 1993. ISSN 0018-9545. doi: 10.1109/25.192382. [cited at p. 27, 28]
- [37] A. Eisenblätter. A frequency assignment problem in cellular phone networks. Technical report, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, Takustr. 7, D-14195 Berlin, Germany, July 10 1997. (Extended Abstract). [cited at p. 11]
- [38] A. Eisenblätter. *Frequency assignment in GSM networks: Models, heuristics and lower bounds*. Ph.d. thesis, Technische Universität Berlin, June 2001. [cited at p. 8, 14]
- [39] A. Eisenblätter and A. Koster. Official web site of the FAP problem [online]. <http://fap.zib.de/>, Accessed on Jan 2011. [cited at p. 7, 10, 11, 24, 25, 26, 28, 79]
- [40] A. Eisenblättert, M. Grötschel, and A. M. Kstet. Frequency planning and ramifications of coloring. *Discussiones Mathematicae Graph Theory*, 22(1):51–58, 2002. [cited at p. 8]
- [41] H.-Y. Fan and J. Lampinen. A trigonometric mutation operation to differential evolution. *Journal of Global Optimization*, 27(1):105–129, September 2003. doi: 10.1023/A:1024653025686. URL <http://portal.acm.org/citation.cfm?id=846119.846124>. [cited at p. 38]

- [42] C. M. Fonseca, L. Paquete, and M. López-Ibáñez. An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE Congress on Evolutionary Computation, 2006. CEC 2006.*, pages 1157–1163, Vancouver, BC, Canada, 16-21 July 2006. doi: 10.1109/CEC.2006.1688440. [cited at p. 91]
- [43] N. Funabiki and S. Nishikawa. An improved neural network for channel assignment problems in cellular mobile communication systems. *IEICE Transactions on Communications*, E78-B(8):1187–1196, 1995-08-25. ISSN 09168516. URL <http://ci.nii.ac.jp/naid/110003217430/en/>. [cited at p. 27]
- [44] N. Funabiki and Y. Takefuji. A neural network parallel algorithm for channel assignment problems in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 41(4):430–437, November 1992. ISSN 0018-9545. doi: 10.1109/25.182594. [cited at p. 24, 27, 28]
- [45] A. Gamst. Some lower bounds for a class of frequency assignment problems. *IEEE Transactions on Vehicular Technology*, VT-35(1):8–14, February 1986. [cited at p. 26, 27, 28]
- [46] A. Gamst. Application of graph theoretical methods to GSM radio network planning. In *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 2, pages 942–945, Jun 1991. doi: 10.1109/ISCAS.1991.176519. [cited at p. 25, 27]
- [47] A. Gamst and W. Rave. On frequency assignment in mobile automatic telephone systems. In *IEEE Global Communication Conference GLOBECOM'82, Miami, USA*, pages 309–315, November 29-December 2 1982. [cited at p. 8, 9, 11, 26, 27, 28]
- [48] M. J. Geiger. Randomised variable neighbourhood search for multi objective optimisation. *CoRR*, abs/0809.0271, 2008. URL <http://arxiv.org/abs/0809.0271>. [cited at p. 29]
- [49] S. C. Ghosh, B. P. Sinha, and N. Das. A new approach to efficient channel assignment for hexagonal cellular networks. *International Journal of Foundations of Computer Science (IJFCS)*, 14(3):439–463, 2003. doi: 10.1142/S0129054103001832. [cited at p. 24, 27]
- [50] S. C. Ghosh, B. P. Sinha, and N. Das. Coalasced CAP: An improved technique for frequency assigment in cellular networks. *IEEE Transaction on Vehicular Tecnology*, 55(2):640–653, March 2006. ISSN 0018-9545. doi: 10.1109/TVT.2005.86335. [cited at p. 24]
- [51] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sciences*, 8(1):156–166, 1977. ISSN 1540-5915. doi: 10.1111/j.1540-5915.1977.tb01074.x. URL <http://dx.doi.org/10.1111/j.1540-5915.1977.tb01074.x>. [cited at p. 34]
- [52] F. Glover. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, 13(5):533–549, 1986. ISSN 0305-0548. doi: [http://dx.doi.org/10.1016/0305-0548\(86\)90048-1](http://dx.doi.org/10.1016/0305-0548(86)90048-1). URL

- <http://www.sciencedirect.com/science/article/B6VC5-48M2YPX-58/2/5ce703998063685581c57432506cdee2>. [cited at p. 34]
- [53] F. Glover. Scatter search and path relinking. In *New ideas in optimization*, pages 297–316. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999. ISBN 0-07-709506-5. [cited at p. 34]
- [54] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 39:653–684, 2000. [cited at p. 34]
- [55] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0201157675. [cited at p. 75]
- [56] GSM World. Gsm world, Jan 2011. URL http://www.gsmworld.com/newsroom/market-data/market_data_summary.htm. [cited at p. 1, 10, 14]
- [57] W. K. Hale. Frequency assignment: Theory and applications. *Proceedings of the IEEE*, 68:1497–1514, December 1980. URL <http://adsabs.harvard.edu/abs/1980IEEEP..68.1497H>. Provided by the SAO/NASA Astrophysics Data System. [cited at p. 7, 8, 9, 24, 26, 27]
- [58] P. Hansen and N. Mladenovic. Variable neighborhood search: Principles and applications. *European Journal of Operational Research*, 130(3):449–467, May 2001. ISSN 0377-2217. doi: DOI:10.1016/S0377-2217(00)00100-4. URL <http://www.sciencedirect.com/science/article/B6VCT-42FS7FG-1/2/f5892eb143b79d617667bb98eee4e88e>. [cited at p. 34, 47, 48, 60, 62]
- [59] P. Hansen and N. Mladenovic. Variable neighborhood search. In F. Glover and G. A. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 145–184. Springer New York, 2003. ISBN 978-0-306-48056-0. doi: 10.1007/0-306-48056-5_6. URL http://dx.doi.org/10.1007/0-306-48056-5_6. [cited at p. 10, 46, 47, 48]
- [60] P. Hansen, N. Mladenovic, and J. A. M. Pérez. Variable neighbourhood search. *Revista Iberoamericana de Inteligencia Artificial*, (19):77–92, 2003. ISSN 1137-3601. [cited at p. 46, 47, 60, 62]
- [61] J.-K. Hao, R. Dorne, and P. Galinier. Tabu search for frequency assignment in mobile radio networks. *Journal of Heuristics*, 4:47–62, 1998. ISSN 1381-1231. URL <http://dx.doi.org/10.1023/A:1009690321348>. 10.1023/A:1009690321348. [cited at p. 27]
- [62] Z. He, Y. Zhang, C. Wei, and J. Wang. A multistage self-organizing algorithm combined transiently chaotic neural network for cellular channel assignment. *IEEE Transactions on Vehicular Technology*, 51(6):1386 – 1396, Nov 2002. ISSN 0018-9545. doi: 10.1109/TVT.2002.804839. [cited at p. 27]
- [63] M. Hellebrandt and H. Heller. A new heuristic method for frequency assignment. Technical Report TD(00) 003, COST 259, Valencia, Spain, Jan. 2000. [cited at p. 25]

- [64] J. H. Holland. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, MI, USA, 1975. [cited at p. 34]
- [65] S. Hurley, D. H. Smith, and S. U. Thiel. FASoft: A system for discrete channel frequency assignment. *Radio Science*, 32(5):1921–1939, 1997. doi: 10.1029/97RS01866. [cited at p. 26, 27]
- [66] W. C. Jakes. *Microwave Mobile Communications*. Wiley-IEEE Press, May 1994. ISBN 978-0-7803-1069-8. [cited at p. 10]
- [67] J. Kalvenes, J. Kennington, and E. Olinick. Hierarchical cellular network design with channel allocation. *European Journal of Operational Research*, 160(1):3–18, January 2005. [cited at p. 28]
- [68] A. R. Kaminsky. A fuzzy genetic algorithm for automatic channel assignment for tactical HF radio networks. In *HF Radio Systems and Techniques, Seventh International Conference on (Conf. Publ. No. 441)*, volume 411, pages 236–241, July 1997. doi: 10.1049/cp:19970796. [cited at p. 27]
- [69] A. Kapsalis, P. Chardaire, V. J. Rayward-Smith, and G. D. Smith. The radio link frequency assignment problem: A case study using genetic algorithms. In T. Fogarty, editor, *Evolutionary Computing*, volume 993 of *Lecture Notes in Computer Science*, pages 117–131. Springer Berlin / Heidelberg, 1995. URL http://dx.doi.org/10.1007/3-540-60469-3_30. [cited at p. 28]
- [70] A. Kapsalis, V. J. Rayward-Smith, and G. D. Smith. Using genetic algorithms to solve the radio link frequency assignment problem. In D. W. Pearson, N. C. Steele, and R. F. Albrecht, editors, *Proceedings of the Second International Conference on Artificial Neural Networks and Genetic Algorithms*. Springer Verlag, 1995. [cited at p. 27]
- [71] D. Karaboga. An idea based on honey bee swarm for numerical optimization. Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri/Türkiye, October 2005. [cited at p. 66, 67]
- [72] D. Karaboga and B. Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108 – 132, 2009. ISSN 0096-3003. doi: DOI:10.1016/j.amc.2009.03.090. URL <http://www.sciencedirect.com/science/article/B6TY8-4W211Y3-1/2/f92b8a34ede7a7e7a24cb2b0834e7c33>. [cited at p. 66]
- [73] D. Karaboga and B. Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3):459–471, November 2007. ISSN 0925-5001. doi: 10.1007/s10898-007-9149-x. URL <http://portal.acm.org/citation.cfm?id=1295970.1296021>. [cited at p. 66]
- [74] D. Karaboga and B. Basturk. On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1):687 – 697, January 2008. ISSN 1568-4946. doi: <http://dx.doi.org/10.1016/j.asoc.2007.05.007>. [cited at p. 66, 122]

- [75] G. Kendall and M. Mohamad. Solving the fixed channel assignment problem in cellular communications using an adaptive local search. In *5th International Conference on the Practice and Theory of Automated Timetabling (PATAT2004), Pittsburgh, USA*, pages 219–231, 18-20 August 2004. URL <http://www.graham-kendall.com/papers/km2004c.pdf>. [cited at p. 38]
- [76] S. Khanna and K. Kumaran. On wireless spectrum estimation and generalized graph coloring. In *INFOCOM '98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1273–1283, Mar-2 Apr 1998. doi: 10.1109/INFCOM.1998.662942. [cited at p. 27]
- [77] J.-S. Kim, S. Park, P. Dowd, and N. Nasrabadi. Channel assignment in cellular radio using genetic algorithms. *Wireless Personal Communications*, 3(3):273–286, Aug. 1996. doi: 10.1007/BF00354875. [cited at p. 24, 27]
- [78] S. Kim and S.-L. Kim. A two-phase algorithm for frequency assignment in cellular mobile systems. *IEEE Transactions on Vehicular Technology*, 43(3):542–548, August 1994. ISSN 0018-9545. doi: 10.1109/25.312792. [cited at p. 27, 28]
- [79] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, May, 13 1983. doi: 10.1126/science.220.4598.671. [cited at p. 34]
- [80] T.-M. Ko. A frequency selective insertion strategy for fixed channel assignment. In *Personal, Indoor and Mobile Radio Communications, 1994. Wireless Networks - Catching the Mobile Future., 5th IEEE International Symposium on*, volume 1, pages 311 –314, 18-23 1994. doi: 10.1109/WNCMF.1994.530813. [cited at p. 24, 25, 26]
- [81] A. Koster, C. van Hoesel, and A. Kolen. Lower bounds for minimum interference frequency assignment problems. Open access publications from maastricht university, Maastricht University, 1999. URL citeseer.ist.psu.edu/koster99lower.html. [cited at p. 25]
- [82] A. M. C. A. Koster. *Frequency Assignment - Models and Algorithms*. PhD thesis, Maastricht University, 1999. URL <http://www.zib.de/koster/thesis.html>. [cited at p. 27, 28]
- [83] D. Kunz. Channel assignment for cellular radio using neural networks. *IEEE Transactions on Vehicular Technology*, 40(1):188–193, February 1991. [cited at p. 25, 27, 28]
- [84] A. Kuurne. On GSM mobile measurement based interference matrix generation. In *IEEE 55th Vehicular Technology Conference, 2002. VTC Spring 2002.*, volume 4, pages 1965–1969, Birmingham, AL, USA, 6-9 May 2002. doi: 10.1109/VTC.2002.1002966. [cited at p. 20, 79]
- [85] W. K. Lai and G. G. Coghill. Channel assignment through evolutionary optimization. *IEEE Transactions on Vehicular Technology*, 45(1):91–96, February 1996. ISSN 0018-9545. doi: 10.1109/25.481825. [cited at p. 25, 27, 28]

- [86] R. Leese and S. Hurley. *Methods and Algorithms for Radio Channel Assignment*. Oxford Lecture Series in Mathematics and its Applications. Oxford University Press, Oxford, United Kingdom, 2002. URL <http://www.oup.co.uk/isbn/0-19-850314-8>. [cited at p. 8]
- [87] R. A. Leese. A unified approach to the assignment of radio channels on a rectangular hexagonal grid. *IEEE Transactions on Vehicular Technology*, 46(4):968–980, Nov 1997. [cited at p. 27]
- [88] Y.-C. Liang, ngela Hsiang-Ling Chen, and C.-Y. Tien. Variable neighborhood search for multi-objective parallel machine scheduling problems. In *Proceeding of the 8th International Conference on Information and Management Sciences (IMS 2009)*, volume 16, pages 511–535, 2009. [cited at p. 29]
- [89] H. R. Lourenço, O. Martin, and T. Stützle. A beginner’s introduction to iterated local search. In *Proceedings 4 th Metaheuristics International Conference, Porto, Portugal*, pages 1–11, 2001. [cited at p. 34]
- [90] H. R. Lourenço, O. Martin, and T. Stützle. Iterated local search. In F. S. Hillier, F. Glover, and G. Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 320–353. Springer New York, 2003. ISBN 978-0-306-48056-0. doi: 10.1007/0-306-48056-5_11. URL http://dx.doi.org/10.1007/0-306-48056-5_11. [cited at p. 34]
- [91] F. Luna, E. Alba, A. J. Nebro, and S. Pedraza. Evolutionary algorithms for real-world instances of the automatic frequency planning problem in gsm networks. In C. Cotta and J. van Hemert, editors, *Evolutionary Computation in Combinatorial Optimization*, volume 4446 of *Lecture Notes in Computer Science*, pages 108–120. Springer Berlin / Heidelberg, April 2007. doi: 10.1007/978-3-540-71615-0_10. URL http://dx.doi.org/10.1007/978-3-540-71615-0_10. [cited at p. 16, 25, 29]
- [92] F. Luna, C. Blum, E. Alba, and A. J. Nebro. ACO vs EAs for solving a real-world frequency assignment problem in GSM networks. In *Proceedings of the 9th annual conference on genetic and evolutionary computation*, GECCO ’07, pages 94–101, New York, NY, USA, 7- 11 July 2007. ACM Press. ISBN 978-1-59593-697-4. doi: <http://doi.acm.org/10.1145/1276958.1276972>. URL <http://doi.acm.org/10.1145/1276958.1276972>. [cited at p. 2, 14, 16, 18, 20, 21, 25, 29, 30, 34, 79, 105, 108]
- [93] F. Luna, C. Estébanez, C. León, J. M. Chaves-González, E. Alba, R. Aler, C. Segura, M. A. Vega-Rodríguez, A. J. Nebro, J. M. Valls, G. Miranda, and J. A. Gómez-Pulido. Metaheuristics for solving a real-world frequency assignment problem in gsm networks. In *Proceedings of the 10th annual conference on genetic and evolutionary computation*, pages 1579–1586, New York, NY, USA, 12-16 July 2008. ACM. ISBN 978-1-60558-130-9. doi: <http://doi.acm.org/10.1145/1389095.1389396>. URL <http://doi.acm.org/10.1145/1389095.1389396>. [cited at p. 18, 20, 25, 29, 37, 38, 44, 47, 79, 95, 105, 108]

- [94] F. Luna, C. Estébanez, C. León, J. M. Chaves-González, A. J. Nebro, R. Aler, C. Segura, M. A. Vega-Rodríguez, E. Alba, J. M. Valls, G. Miranda, and J. A. Gómez-Pulido. Optimization algorithms for large-scale real-world instances of the frequency assignment problem. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, pages 1–16, 2010. ISSN 1432-7643. doi: 10.1007/s00500-010-0653-4. URL <http://dx.doi.org/10.1007/s00500-010-0653-4>. [cited at p. 29]
- [95] R. Malladi and D. P. Agrawal. Applications of mobile and wireless networks: Current and future. *Communications of the ACM*, 45(10):144–146, Oct. 2002. [cited at p. 25, 26]
- [96] C. Mannino, G. Oriolo, and F. Ricci. Solving stability problems on a superclass of interval graphs. Technical Report 26-02, Universitá di Roma ‘La Sapienza’, 2002. URL <http://www.dis.uniroma1.it/~mannino/papers/thin.pdf>. [cited at p. 25]
- [97] O. C. Martin and S. W. Otto. Combining simulated annealing with local search heuristics. *Annals of Operations Research*, 63(1):57–75, 1996. ISSN 0254-5330. doi: 10.1007/BF02601639. URL <http://dx.doi.org/10.1007/BF02601639>. [cited at p. 34]
- [98] R. Mathar and J. Mattfeldt. Channel assignment in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 42(4):647–656, Nov. 1993. [cited at p. 27, 28]
- [99] R. Mathar and M. Schmeink. Optimal base station positioning and channel assignment for 3G mobile networks by integer programming. *Annals of Operations Research*, 107(1–4):225–236, 2002. doi: 10.1023/A:1014959317542. URL <http://ipsapp007.lwwonline.com/content/getfile/4495/39/13/abstract.htm>. [cited at p. 27]
- [100] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. El algoritmo evolución diferencial aplicado al problema de la asignación de frecuencias. In *I Jornadas sobre Algoritmos Evolutivos y Metaheurísticas (JAEM’07)*, pages 81–88. Thomson, September 2007. ISBN 978-84-9732-614-8. [cited at p. 28]
- [101] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. O algoritmo de evolução diferencial como abordagem de resolução do FAP. In *Conferência Ibero-Americana IASK InterTIC 2007*, 978-972-99397-6-1, pages 173–178, December 2007. ISBN 978-972-99397-6-1. [cited at p. 28]
- [102] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. Solving the frequency assignment problem with differential evolution. In *15th International Conference on Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007.*, SoftCOM 2007, pages 1–5, September 2007. ISBN 953-6114-95-X. doi: 10.1109/SOFTCOM.2007.4446075. [cited at p. 28]
- [103] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. A hybrid differential evolution algorithm to solve a real-world frequency assignment problem. In *International Multiconference on Computer Science and*

- Information Technology, 2008. IMCSIT 2008.*, volume 3, pages 201 – 205, Wisla, Poland, 20-22 October 2008. IEEE Catalog Number (USA) CFP0864E-CDR. ISBN 978-83-60810-14-9. doi: 10.1109/IMCSIT.2008.4747240. [cited at p. 25]
- [104] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. Analysis of parameter settings for differential evolution algorithm to solve a real-world frequency assignment problem in GSM networks. In *The Second International Conference on Advanced Engineering Computing and Applications in Sciences, 2008. ADVCOMP '08.*, pages 77–82, Los Alamitos, CA, USA, 29 Sep - 4 Oct, Valencia, Spain 2008. IEEE Computer Society. ISBN 978-0-7695-3369-8. doi: 10.1109/ADVCOMP.2008.18. [cited at p. 25, 47]
 - [105] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. Multiobjective frequency assignment problem using the MO-VNS and MO-SVNS algorithms. In *World Congress on Nature Biologically Inspired Computing, 2009. NaBIC 2009.*, pages 221–226. IEEE, 9-11 Dec. 2009. doi: 10.1109/NABIC.2009.5393695. ISBN: 978-1-4244-5612-3. [cited at p. 25]
 - [106] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. Parameter analysis for differential evolution with pareto tournaments in a multiobjective frequency assignment problem. In E. Corchado and H. Yin, editors, *Intelligent Data Engineering and Automated Learning - IDEAL 2009*, volume 5788 of *Lecture Notes in Computer Science*, pages 799–806. Springer Berlin / Heidelberg, 2009. ISBN 3-642-04393-3, 978-3-642-04393-2. doi: 10.1007/978-3-642-04394-9. URL http://dx.doi.org/10.1007/978-3-642-04394-9_98. [cited at p. 25]
 - [107] M. Maximiano, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, and J. M. Sánchez-Pérez. Comparative analysis of a hybrid DE algorithm with the VNS algorithm and its variation SVNS to solve a real-world frequency assignment problem. *Applied Artificial Intelligence Journal*, pages 1–24, 2011. ISSN 0883-9514. [cited at p. 25]
 - [108] Microsoft. MSDN. [online]. <http://msdn.microsoft.com/en-gb/library/kx37x362.aspx>, Accessed on Jan 2011. [cited at p. 85]
 - [109] N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, November 1997. ISSN 0305-0548. doi: 10.1016/S0305-0548(97)00031-2. URL <http://www.sciencedirect.com/science/article/B6VC5-3SX0K5F-1B/2/d2d61cc88e3aeb65c97cd8e8149d44f2>. [cited at p. 34, 47, 48]
 - [110] M. Mouly and M.-B. Pautet. *The GSM System for Mobile Communications*. Telecom Publishing, 1992. ISBN 0945592159. [cited at p. 14]
 - [111] R. A. Murphrey, P. M. Pardalos, Mauricio, and M. G. Resende. Frequency assignment problems. In *Handbook of Combinatorial Optimization*, pages 295–377. Kluwer Academic Publishers, 1999. [cited at p. 7]
 - [112] L. Narayanan and S. Shende. Static frequency assignment in cellular networks. In *Proceedings of the Fourth Colloquium on Structural Information and Communications Complexity, Ascona, Switzerland*, pages 215–227, July 1997. [cited at p. 27]

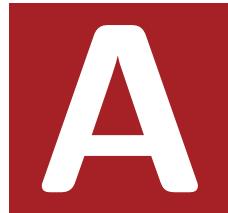
- [113] L. Narayanan and S. M. Shende. Static frequency assignment in cellular networks. *Algorithmica*, 29:396–409, 2001. ISSN 0178-4617. URL <http://dx.doi.org/10.1007/s004530010067>. doi: 10.1007/s004530010067. [cited at p. 27]
- [114] P. Ngatchou, A. Zarei, and M. A. El-Sharkawi. Pareto multi objective optimization. In *Proceedings of the 13th International Conference on Intelligent Systems Application to Power Systems*, pages 84–91, November 2005. doi: 10.1109/ISAP.2005.1599245. [cited at p. 29, 53]
- [115] I. H. Osman and G. Laporte. Metaheuristics: A bibliography. *Annals of Operations Research*, 63(5):511–623, 1996. ISSN 0254-5330. doi: 10.1007/BF02125421. URL <http://www.springerlink.com/content/V0872P6K2086186W>. [cited at p. 34]
- [116] K. Price and R. Storn. DE website [online]. <http://www.ICSI.Berkeley.edu/~storn/code.html>, Accessed on Jan 2011. [cited at p. 38, 41, 57]
- [117] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Natural Computing Series. Springer; 1 edition, December 2005. ISBN: 3540209506. [cited at p. 38, 57]
- [118] A. Qing. *Differential Evolution: Fundamentals and Applications in Electrical Engineering*. Wiley-IEEE Press, July 2009. ISBN: 978-0-470-82392-7. [cited at p. 29]
- [119] C. R. Raquel and J. Prospero C. Naval. An effective use of crowding distance in multiobjective particle swarm optimization. In *Proceedings of the 2005 conference on Genetic and evolutionary computation*, GECCO ’05, pages 257–264, New York, NY, USA, 2005. ACM. ISBN 1-59593-010-8. doi: 10.1145/1068009.1068047. URL <http://doi.acm.org/10.1145/1068009.1068047>. [cited at p. 70]
- [120] P. Raymond. Performance analysis of cellular networks. *IEEE Transactions on Communications*, 39(12):1787–1793, December 1991. ISSN 0090-6778. doi: 10.1109/26.120164. [cited at p. 10]
- [121] H. G. Sandalidis, P. P. Stavroulakis, and J. Rodriguez-Tellez. An efficient evolutionary algorithm for channel resource management in cellular mobile systems. *IEEE Transactions on Evolutionary Computation*, 2(4):125–137, November 1998. ISSN 1089-778X. doi: 10.1109/4235.738972. [cited at p. 27]
- [122] M. Schulz and T. U. Berlin. *Solving Frequency Assignment Problems with Constraint Programming*. PhD thesis, Institut fur Mathematik der Technischen Universitat Berlin, Berlin, February 2003. [cited at p. 27]
- [123] A. Sen, T. Roxborough, and S. Medidi. Upper and lower bounds of a class of channel assignment problems in cellular networks. In *INFOCOM ’98. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1284–1291, March-2 April 1998. doi: 10.1109/INFCOM.1998.662943. [cited at p. 27]
- [124] A. Sen, T. RoxBorough, and B. P. Sinha. On an optimal algorithm for channel assignment in cellular networks. In *IEEE International Conference on Communications, 1999. ICC ’99. 1999*, volume 2, pages 1147–1151, Vancouver, Canada, 6-10 June 1999. doi: 10.1109/ICC.1999.765476. [cited at p. 26]

- [125] M. Sengoku, H. Tamura, S. Shinoda, T. Abe, and Y. Kajitani. Graph theoretical considerations of channel offset systems in a cellular mobile system. *IEEE Transaction on Vehicular Technology*, 40(2):405–411, May 1991. ISSN 0018-9545. doi: 10.1109/25.289421. [cited at p. 26, 27]
- [126] Y.-J. Shen and M.-S. Wang. New resource management strategy for wireless cellular networks. *Computers and Electrical Engineering*, 32(1-3):135–146, 2006. ISSN 0045-7906. doi: 10.1016/j.compeleceng.2006.01.004. URL <http://www.sciencedirect.com/science/article/B6V25-4JKJSXW-2/2/508a2fc976e69f2e41f9ab1bda570e01>. [cited at p. 24, 25]
- [127] W.-Y. Shin, S. Y. Chang, J. Lee, and C.-H. Jun. Frequency insertion strategy for channel assignment problem. *Wireless Networks*, 12(1):45–52, February 2006. ISSN 1022-0038. doi: <http://dx.doi.org/10.1007/s11276-006-6149-6>. URL <http://dx.doi.org/10.1007/s11276-006-6149-6>. [cited at p. 24]
- [128] S. A. G. Shirazi and H. Amindavar. Fixed channel assignment using new dynamic programming approach in cellular radio networks. *Computers and Electrical Engineering*, 31(4-5):303–305, June 2005. ISSN 0045-7906. doi: 10.1016/j.compeleceng.2005.04.002. URL <http://www.sciencedirect.com/science/article/B6V25-4GYH7J7-1/2/ba0f4ff04418dd0c78a7a6a2300659b3>. [cited at p. 10, 11, 24, 26, 28]
- [129] K. Sivarajan, R. McEliece, and J. Ketchum. Channel assignment in cellular radio. In *Vehicular Technology Conference, 1989, IEEE 39th*, volume 2, pages 846–850, May 1-3 1989. doi: 10.1109/VETEC.1989.40173. [cited at p. 9, 24, 26, 27, 28]
- [130] D. H. Smith, S. Hurley, and S. U. Thiel. Improving heuristics for the frequency assignment problem. *European Journal of Operational Research*, 107(1):76–86, 1998. ISSN 0377-2217. doi: 10.1016/S0377-2217(98)80006-4. URL <http://www.sciencedirect.com/science/article/B6VCT-3TDH3FX-6/2/4a95d42d52930054fac9e30691cdb50b>. [cited at p. 26]
- [131] D. H. Smith, S. Hurley, and S. M. Allen. A new lower bound for the channel assignment problem. *IEEE Transactions on Vehicular Technology*, 49(4):1265–1272, July 2000. ISSN 0018-9545. doi: 10.1109/25.875238. [cited at p. 26]
- [132] K. Smith and M. Palaniswami. Static and dynamic channel assignment using neural networks. *IEEE Journal on Selected Areas in Communication*, 15(2):238–249, February 1997. ISSN 0733-8716. doi: 10.1109/49.552073. [cited at p. 27]
- [133] K. A. Smith. A genetic algorithm for the channel assignment problem. In *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, volume 4, pages 2013–2018, 1998. doi: 10.1109/GLOCOM.1998.775893. [cited at p. 27]
- [134] J. soo Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi. Cellular radio channel assignment using a modified hopfield network. *IEEE Transactions on Vehicular Technology*, 46(4):957–967, November 1997. ISSN 0018-9545. doi: 10.1109/25.653070. [cited at p. 24, 27]

- [135] R. Storn and K. Price. Differential evolution - a simple evolution strategy for fast optimization. *Dr. Dobb's Journal*, (22(4)):18–24 and 78, April 1997. [cited at p. 28, 38, 41, 57]
- [136] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997. [cited at p. 28, 38, 57]
- [137] T. Stützle. *Local Search Algorithms for Combinatorial Problems - Analysis, Algorithms and New Applications*. PhD thesis, Technische Universität Darmstadt, 1999. [cited at p. 34]
- [138] T. Stützle. Iterated local search for the quadratic assignment problem. *European Journal of Operational Research*, 174(3):1519 – 1539, 2006. ISSN 0377-2217. doi: 10.1016/j.ejor.2005.01.066. URL <http://www.sciencedirect.com/science/article/B6VCT-4G5BJF-5/2/036f19a8d78e2ce79e0bcdac40c51c47>. [cited at p. 34]
- [139] C. W. Sung and W. S. Wong. Sequential packing algorithm for channel assignment under cochannel and adjacent-channel interference constraint. *IEEE Transactions on Vehicular Technology*, 46(3):676–685, August 1997. ISSN 0018-9545. doi: 10.1109/25.618193. [cited at p. 24, 26]
- [140] E.-G. Talbi. *Metaheuristics: From Design to Implementation*. Parallel & Distributed Computing. John Wiley & Sons, Inc., July 2009. [cited at p. 29, 54, 66]
- [141] H. Tamura, M. Sengoku, and S. Shinoda. Channel assignment problem in a cellular mobile system and a new coloring problem of networks. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E74-A(10):2983–2989, October 1991. [cited at p. 27]
- [142] D.-W. Tcha, J.-H. Kwon, T.-J. Choi, and S.-H. Oh. Perturbation-minimizing frequency assignment in a changing tdma/fdma cellular environment. *IEEE Transactions on Vehicular Technology*, 49(2):390–396, March 2000. ISSN 0018-9545. doi: 10.1109/25.832969. [cited at p. 27]
- [143] S. R. Tiourine, C. A. J. Hurkens, and J. K. Lenstra. An overview of algorithmic approaches to frequency assignment problems. In *CALMA Symposium on Combinatorial Algorithms for Military Applications*, pages 53–62, 1995. [cited at p. 25]
- [144] C. Valenzuela, S. Hurley, and D. H. Smith. A permutation based genetic algorithm for minimum span frequency assignment. In A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN V*, volume 1498, pages 907–916. Springer Berlin / Heidelberg, 1998. doi: 10.1007/BFb0056932. URL <http://link.springer.de/link/service/series/0558/bibs/1498/14980907.htm>. [cited at p. 27]
- [145] C. Voudouris and E. Tsang. Solving the radio link frequency assignment using guided local search, 1998. [cited at p. 38]

- [146] W. Wang and C. Rushforth. An adaptive local-search algorithm for the channel-assignment problem (CAP). *IEEE Transactions on Vehicular Technology*, 45(3):459–466, August 1996. ISSN 0018-9545. doi: 10.1109/25.533761. [cited at p. 24, 27, 28, 38]
- [147] N. Weicker, G. Szabo, K. Weicker, and P. Widmayer. Evolutionary multiobjective optimization for base station transmitter placement with frequency assignment. *IEEE Transactions on Evolutionary Computation*, 7(2):189–203, April 2003. ISSN 1089-778X. doi: 10.1109/TEVC.2003.810760. [cited at p. 28, 57, 58]
- [148] M. Zhang and T.-S. Yum. The nonuniform compact pattern allocation algorithm for cellular mobile systems. *IEEE Transactions on Vehicular Technology*, 40(2):387–391, May 1991. ISSN 0018-9545. doi: 10.1109/25.289419. [cited at p. 28]
- [149] X. Zhou and T. Nishizeki. Efficient algorithms for weighted colorings of series-parallel graphs. In P. Eades and T. Takaoka, editors, *Algorithms and Computation*, volume 2223 of *Lecture Notes in Computer Science*, pages 514–524. Springer Berlin / Heidelberg, 2001. doi: 10.1007/3-540-45678-3_44. URL http://dx.doi.org/10.1007/3-540-45678-3_44. [cited at p. 26]
- [150] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In A. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, editors, *Proceedings of the 5th International Conference on Parallel Problem Solving from Nature*, volume 1498 of *PPSN V*, pages 292–304. Springer Berlin / Heidelberg, London, UK, 27-30 September 1998. ISBN 3-540-65078-4. [cited at p. 29, 54, 91, 93]
- [151] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999. ISSN 1089-778X. doi: 10.1109/4235.797969. [cited at p. 73]
- [152] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm. TIK-Report 103 103, Computer Engineering and Networks Laboratory (TIK), Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001. [cited at p. 73, 74]

Appendices



Software Development

This appendix presents some of the main characteristics followed by the several applications that were developed to implement the algorithms analyzed in this work.

A.1 Metaheuristics Applications

In the implementation of the several algorithms an approach was followed that tried to design a modular software, able to incorporate the specific characteristics of the problem as well as the intrinsic logic of the several metaheuristics. Therefore, since the beginning of the design phase of the project, an object-oriented programming approach was used.

A specific application was developed for each metaheuristic, although they have shared the same base classes. Inside each project we have grouped in specific modules the classes containing similar features. As an example, all the classes inside the *Problem* module define the characteristics of the problem, and they are exactly the same for all the types of metaheuristics. In the *Algorithm* module, all the classes required to define the metaheuristic are available. In some special cases, the name *Algorithm* module was replaced by the name of the algorithm, as for example the case of DEPT algorithm, where all the classes required to implement the metaheuristic are inside the *DE* module. These two modules were incorporated in the implementation of all the metaheuristics, with some adjustments in the *Algorithm* module.

In Figure A.1 two examples, which express the classes organization developed

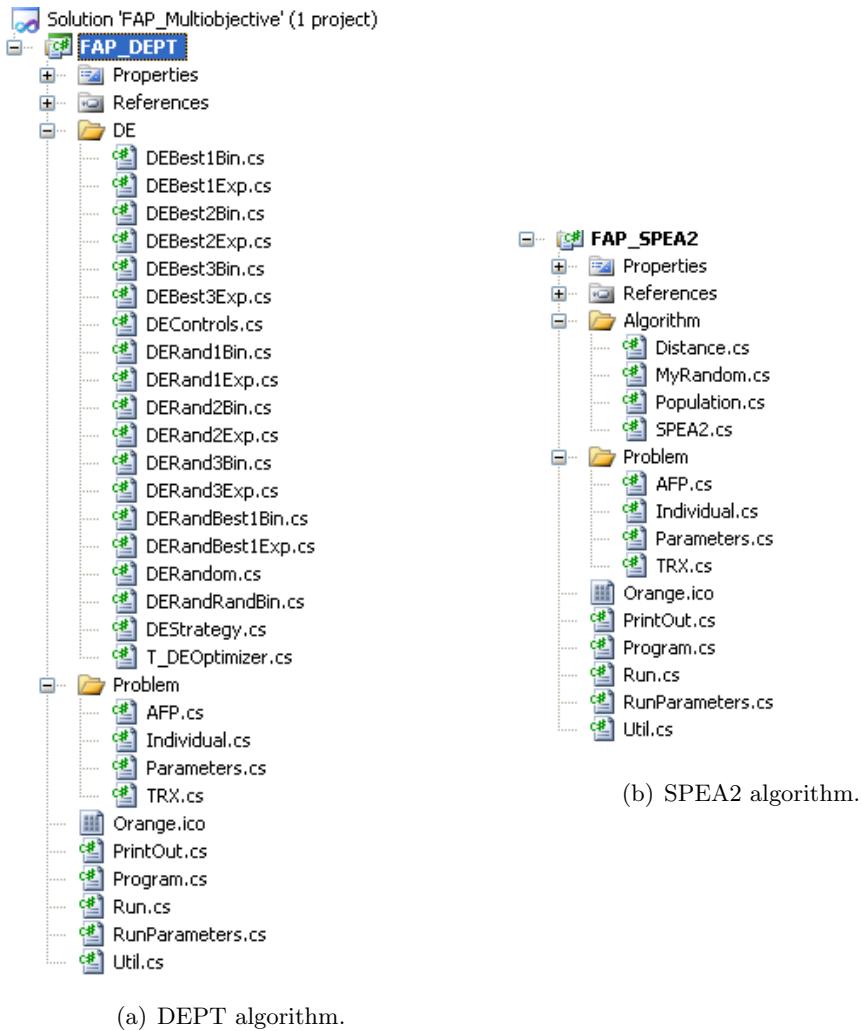


Figure A.1: solutions.

for the several applications are visible. They represent two different metaheuristics implementation. The first represents the DEPT algorithm implementation and the second, the SPEA2 implementation. It is visible that the common classes are organized in the *Problem* module. The *Algorithms* module includes the classes required to implement the algorithms logic. The remaining applications follows a similar structure.

The remaining classes are available in the root project and they define the main class and the other auxiliary classes. They are required to process the input parameters and export the statistical information. The input parameters are obtained using a XML file, which is structured based on the parameters required by the implemented algorithm. For each algorithm the required parameters can change. Therefore, the XML input file is structured based on the parameters

```

C:\DEPT>FAP_DEPT.exe -instance=denver
N. de Parametros: 1
----- Params<XML File>-----
nCR:      0,01
nF:       1
popSize:   75
strategy: Rand1Bin
localSearch: False
seed:      5
numberOfRuns: 10
type:

1 run(s)
Instance Name: denver
Options sucessfully readed
2612 TRMs sucessfully readed
708 IM sucessfully readed
711 Sectors sucessfully readed
289 Sites sucessfully readed
***.
.....End
----
```

Figure A.2: Command line interface on Windows to launch DEPT algorithm.

required by the algorithm. The output information is reported in text files. These files summarize the obtained frequency plans and all statistical data of the run.

The software, which was developed for every metaheuristic runs on the command line and the user needs to set on what instance will be performed in the execution. The application requires the *parameters.xml* file with the configuration parameters containing the necessary settings for the algorithm. In the command line, besides the parameter settings, it is also possible to configure some run environment settings, as for example the execution time, the FAP instance name, etc.

Next, in the XML listing we present some examples about the content of some XML files used when loading the configurations for the metaheuristics. Additionally, we have also set a default configuration for some of the command line parameters. In the example, the first XML example file shows the configuration parameters required for an execution of the DE algorithm. The second example is more complex and it represents multiple runs, which were all configured in a unique XML file. The XML file will process the executions one at a time. The second example represents the approach set to perform an automatic tuning set of experiments.

Listing A.1: XML configuration file with a single configuration.

```
<?xml version="1.0" encoding="UTF-8"?>
<runs>
  <parameters>
    <F>0.9</F>
    <CR>0.01</CR>
    <popSize>75</popSize>
    <seed>5</seed>
    <strategy>RandBest1Exp</strategy>
    <numberOfRuns>30</numberOfRuns>
    <localSearch>true</localSearch>
  </parameters>
</runs>
```

Listing A.2: XML configuration file containing multiple configurations.

```
<?xml version="1.0" encoding="UTF-8"?>
<runs>
  <parameters>
    <F>0.8</F>
    <CR>0.9</CR>
    <popSize>10</popSize>
    <seed>5</seed>
    <strategy>Rand2Bin</strategy>
    <numberOfRuns>10</numberOfRuns>
    <localSearch>false</localSearch>
  </parameters>
  <parameters>
    <F>0.8</F>
    <CR>0.9</CR>
    <popSize>25</popSize>
    <seed>5</seed>
    <strategy>Rand2Bin</strategy>
    <numberOfRuns>10</numberOfRuns>
    <localSearch>false</localSearch>
  </parameters>
...
</runs>
```

In Figure A.2 an example is given for the output console execution of an algorithm. In the *console application* feedback is given to the user about the loaded parameter file, the FAP instance data and also the current status run. Although it is not visible in the figure, for each run it is created an output directory where the application creates all the output files for every run.

To launch the command line interface, the user needs to set as an argument the instance to be used (using the `-instance=[instance name]` command). By default, the application is carried out for 30 minutes, but if the users desire it can also configure the time of execution, using the argument `-time=[control time]`.

A.2 Graphic Interface to Multiobjective Optimization Indicators

In multiobjective optimization, two metrics were used to quantify the performance: Hypervolume and Coverage Relation. Both metrics use the Pareto Front solutions. Since we compared a large amount of information generated by all

the sequences of test performed during research, we have decided to develop an additional application to perform the final evaluation of the accomplished results. The application aims to evaluate the Pareto Fronts, considering both metrics.

Some of the most important features of the application are:

- The input information required by the application is represented in text files.
- The user can configure which instance is associated to the data being analyzed.
- For the hypervolume indicators the user can configure the range of values for the necessary reference points.
- It represents graphically the solutions read from a text file, and identifies the Pareto front solutions.
- For every metric it permits the export of the results information.

To better understand some of the features available in the software, we present the most important features.

The software was developed considering that users may need to simultaneously analyze several Pareto Fronts, therefore it was developed based on a MDI application. By adopting this approach, several windows can be used simultaneously by the user inside the Statistic Indicators application. Figure A.3 shows the hypervolume reference points configuration window. For the user, the focus in that configuration windows permits to identify which instance of the Pareto Front it is being analyzed. In the menu area, on the right top of the application, it is also possible to identify which FAP instance it represents. Beside the configuration window, the Figure A.3 also shows the data about an execution text file, which represents graphically all the solutions obtained with the algorithm.

As already mentioned, the metrics application reads all the data from the algorithm output file. Figure A.4 summarizes the Pareto Front solutions. The option is available in the main menu of the application. The selection of this feature implies that the user only desires to see the Pareto Front solutions. The information is shown graphically and in the text format (in the lower area of the window). The interface permits the user to export the information to other file formats.

The comparison of several output data can be performed simultaneously, as shown in Figure A.5. It shows two distinct Pareto Fronts graphically and in the third window it shows in text mode the solutions belonging to the Pareto front for both algorithms. Figure A.5 and Figure A.6 show simultaneously the interface

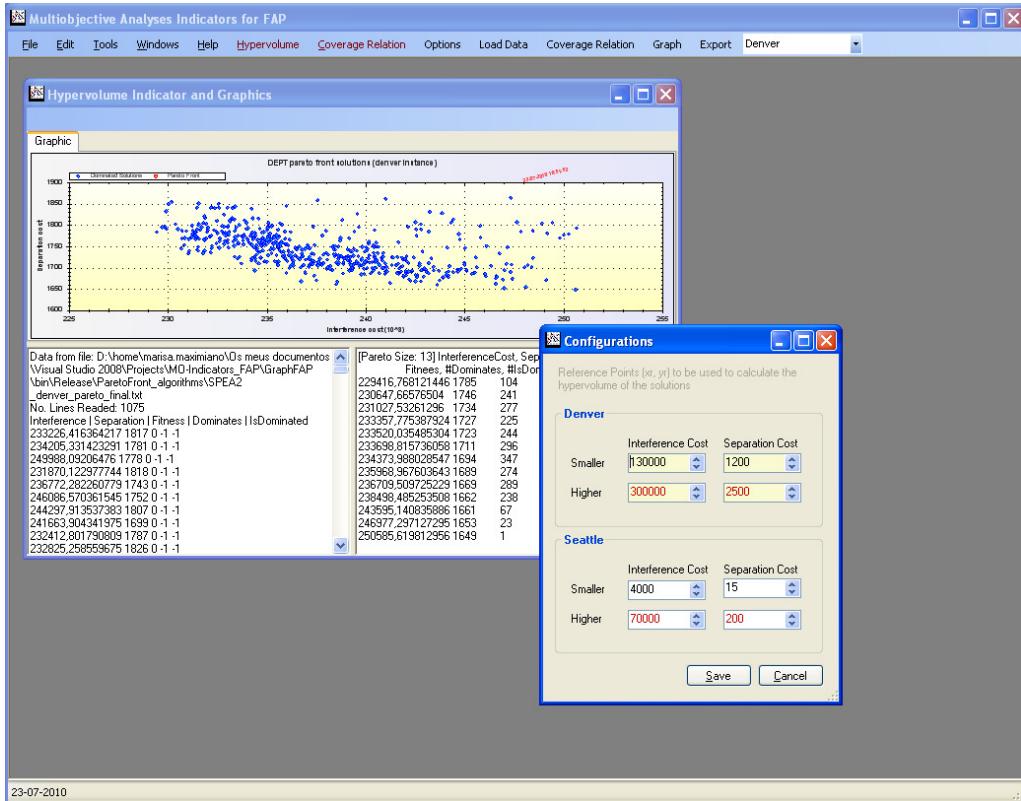


Figure A.3: Configuration interface for the Hypervolume reference points.

for the graphical representation of the Pareto Front information as well as the correspondent coverage relation data. In both of these figures, each Pareto Front is shown in its own graphic and the coverage relation values are presented in an independent form. At each time the user can choose to see only one of the Pareto Fronts.

The user is able to see in the same graph the Pareto Fronts of the two Pareto Fronts for each the coverage relation data is being processed. This scenario is shown in Figure A.7.

Software Requirements Notes

The metrics interface application was developed as a Windows Application, using C# language and Microsoft .NET Framework version 3.5. In the metaheuristics applications it was used the same programming language and development framework.

Indeed, the main restriction regarding software requisites is the .NET framework, which is essential to execute the software developed in the research. The minimum hardware and software requirements to install and run Microsoft .NET

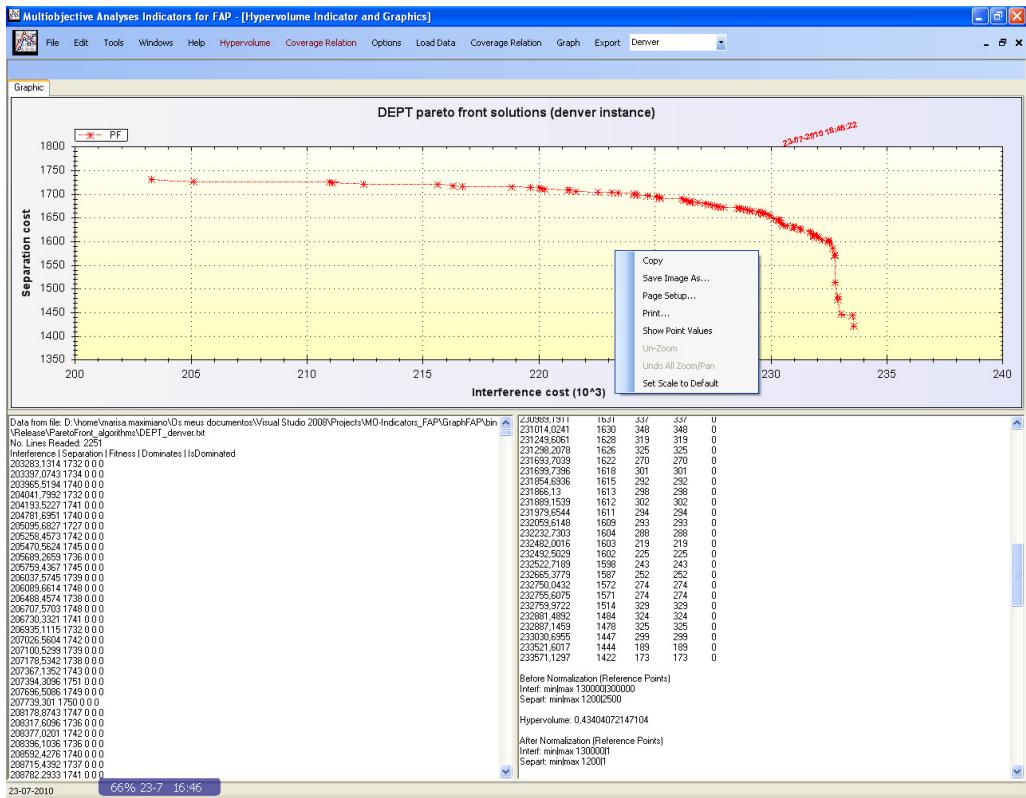


Figure A.4: Graphic interface showing in the graphic the Pareto Front solutions and in the text are the related data.

Framework version 3.5 and Visual Studio .NET 2008 are some of the characteristics that need to be considered.

The framework is supported by the following architectures:

- x86
- x64
- ia64 (Windows Server 2008)

Supported Operating Systems:

- Microsoft Windows XP
- Microsoft Windows Server 2003
- Windows Vista
- Windows 7



Figure A.5: Graphic interface showing in separate graphics the Pareto Front solutions and a third form showing their coverage relation information.

- Windows Server 2008

Hardware Requirements:

- Minimum: 400 MHz CPU, 96 MB RAM, 800x600 256-color display
- Recommended: 1.0 GHz or higher CPU, 256 MB or more RAM, 1024x768 high-color 32-bit display
- Up to 500 MB of hard disk space may be required

The solutions used to develop the several applications can easily be compiled to future releases of the .NET framework.

More information about how to use the framework is available in the official Microsoft web page at <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=333325fd-ae52-4e35-b531-508d977d32a6>.

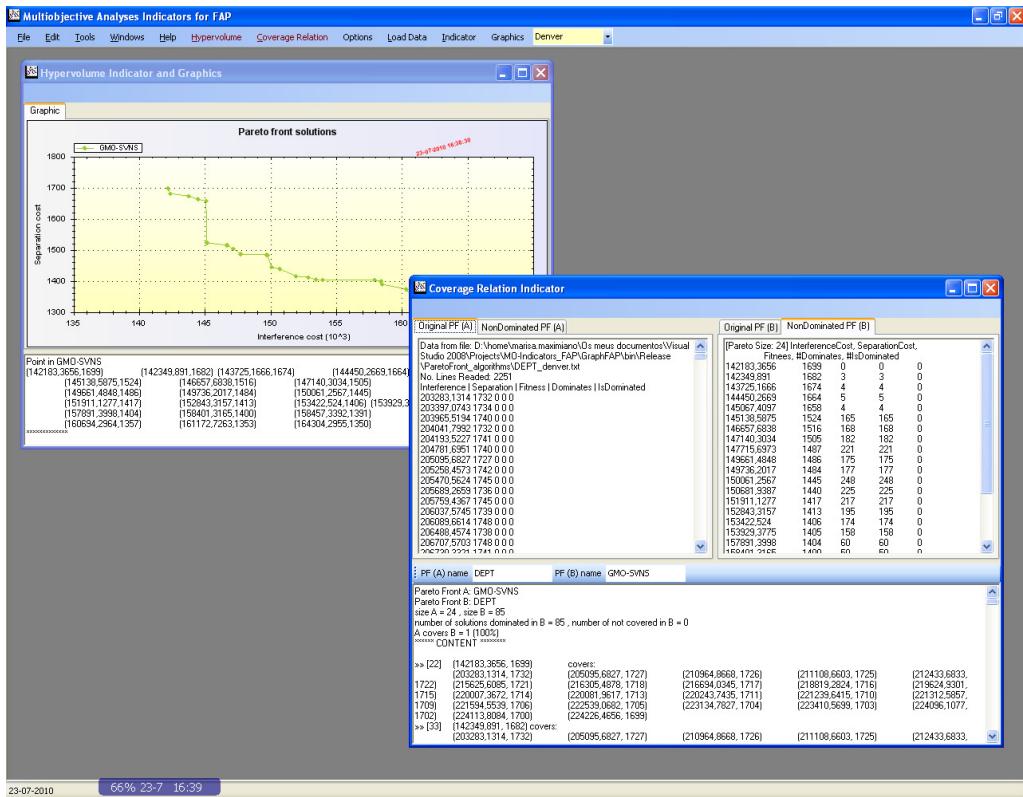


Figure A.6: Graphic interface showing the coverage relation information. Simultaneously, it is also visible the graphic representation of one of Pareto Fronts of the algorithms being analyzed.

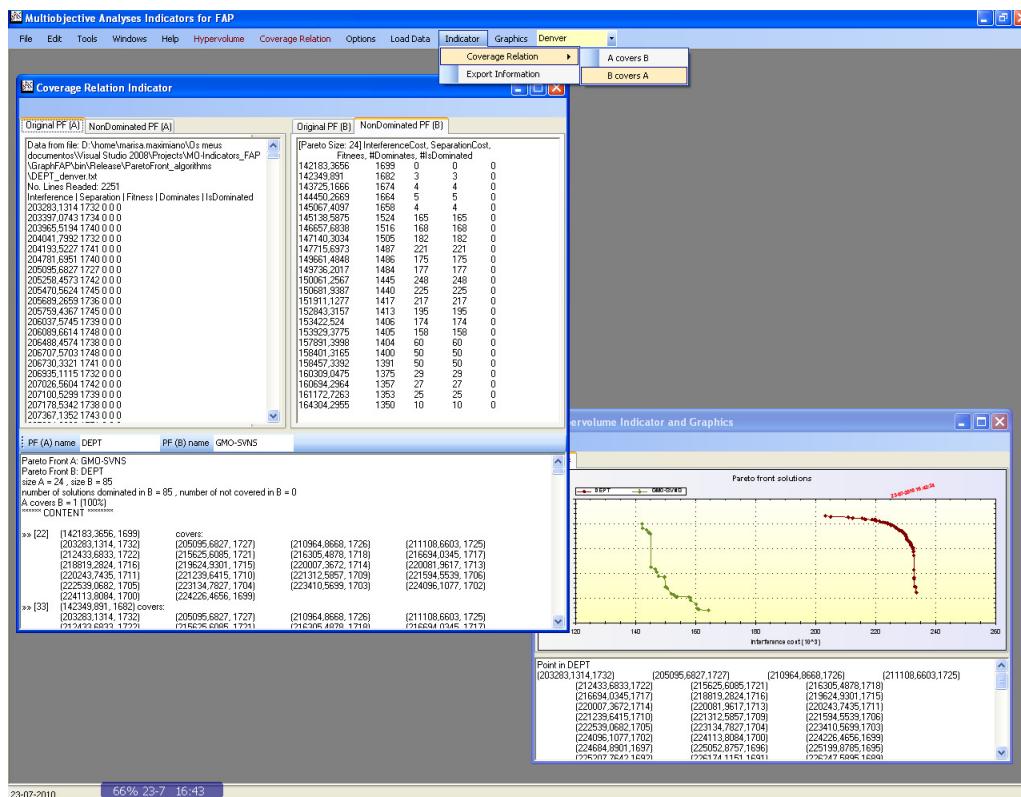


Figure A.7: Graphic interface showing the coverage relation information. Simultaneously, it is also visible a graphic representation containing both Pareto Fronts for the algorithms being analyzed.

Index

- ABC, 66
- ACO, 105, 108
- adjacent-channel, 17
 - constraint, 9, 11
 - interference, 18, 20, 24, 36, 45
- AFP, 7
- Ant Colony Optimization, 34
- antennas, 1, 8, 10, 13
- arithmetic operators, 38
- available frequencies, 19, 37, 45, 65
- available sector, 19
- available spectrum, 10, 18, 19, 23
- base station, 13, 23, 81
- Base station positioning problem, 28
- BCCH, 16, 80
- bees
 - employed, 66, 69
 - onlooker, 66, 68, 69
 - scout, 69
- benchmark, 79
 - instance, 24, 25, 30
 - problem, 13, 14, 22, 23, 30
- binary tournament, 72, 74
- boundary values, 92
- BSC, 15
- BSS, 15
- BTS, 15–17, 21, 78
- candidate solution, 38, 56, 61
- CAP, 7
- CDMA, 14
- cell, 20
- cellular network, 26, 30
- cellular radio network, 11
- channel, 12, 17
- channel assignment, 12, 26
- channel interference, 27
- channel separation matrix, 81
- chromosome, 84, 87
- classic characteristics, 35
- classic method, 27
- clusters, 158
- co-channel, 17
 - constraint, 9, 11, 13, 26
 - interference, 18, 20, 24, 36, 45
- co-site
 - constraint, 9
 - interference, 24
- colony size, 66, 122
- combinatorial problem, 8, 34
- compatibility matrix, 12
- computational characteristics, 89
- configuration file, 21, 78, 80, 83, 85, 88
- conflicting objectives, 53
- constraint, 87
- Constraint Programming, 27
- coverage relation, 53, 90, 91, 93, 145, 158
- crossover
 - one-point, 133
 - eight-points, 133
 - four-points, 133
 - operator, 135, 140
 - probability, 94, 96, 111, 127, 131, 135
 - two-points, 133
 - uniform, 40, 113, 133
- crossover operator, 131
- crossover scheme, 41
 - binomial, 41, 99, 113
 - exponential, 41, 99, 113
- crowding
 - distance, 69, 70, 72
 - operator, 70
 - region, 57
- DCA, 10
- DE parameters, 95
- DE schema, 40, 98
- DE strategy, 39, 41, 94, 98
- DE tuning, 94

- demand vector, 12
- density value, 73
- Denver topology, 79
- DEPT, 57, 88, 144, 146, 148, 157
 - tuning, 111
- DEPT scheme, 58
- DEPT strategy, 111, 113
- Differential Evolution, 28, 33, 34, 38, 46, 57, 107, 156
 - operators, 57
 - parameters, 39
- differential mutation, 39
- differential variation, 58
- distance function, 48, 49, 63
- distance value, 27
- domain knowledge, 43
- domain restrictions, 34
- dominance, 54, 55, 58, 75
- dominance concept, 56, 57, 60, 65, 68
- dynamic assignment, 10
- electromagnetic constraints, 8
- EMC, 2, 9, 11–13, 23
- engineering problems, 8, 31
- evaluation criterion, 36
- Evolutionary Algorithm, 29, 38, 105
- Evolutionary Computing, 34
- evolutionary methods, 34
- evolutionary operations, 85
- execution time, 90
- experiments scenario, 89
- FAP, 1–15, 22, 23
 - classic formulation, 28
 - classification, 9, 23, 24
 - formulation, 2, 4, 5, 7, 13, 15, 18–20, 22, 23, 25, 37, 66
 - instances, 105
 - multiobjective, 5, 22
 - solution, 37
 - variants, 10
- FCA, 10
- fitness function, 19, 53, 70
- fitness value, 39, 49, 52, 56–59, 65, 66, 68, 69, 73
- fixed channel assignment, 10
- food source, 66, 67, 69
- framework, 85, 89
- frequency, 8, 12, 18, 19, 78
 - assignment, 14, 17, 30
 - constraint, 9
 - domain, 11
- plan, 14, 19, 36, 56, 84, 155
- planning, 7, 8, 16–19, 22
- range, 36, 84
- restriction, 36
- spectrum, 2, 7, 12, 14
- value, 20, 27, 36–38
- gene, 84, 87
- Genetic Algorithm, 27, 34, 75
- global optimum, 63
- Global System for Mobile, 1
- GMO-SVNS, 64, 66, 144, 146, 148, 151, 157
 - tuning, 119
- GMO-VNS, 61, 64, 66, 144, 148, 152, 157
- Graph Coloring algorithms, 27
- graph methods, 26
- graph modeling, 26
- graph-coloring problem, 9, 26, 28
- grid computing, 158
- GSM, 2, 8, 14, 15, 17, 18, 80
 - components, 7
 - network, 4, 5, 7, 10, 18, 20, 22, 23, 25, 28, 30, 155
 - optimization, 31
 - service, 14
 - topology, 14, 18
- heuristic methods, 27
- heuristic multi-boot, 48
- Hybrid DE, 43, 94, 106
- hybrid metaheuristics, 28, 35, 36, 38, 87
- hypervolume, 90, 91, 158
 - indicator, 53, 90, 92, 145, 146
- individual representation, 84
- instance
 - CALMA, 24
 - CELAR, 79
 - COST, 25
 - COST259, 24, 79
 - Denver, 25, 78
 - Finland, 25
 - GRAPH, 24, 79
 - Helsinki, 25
 - Philadelphia, 24, 26, 28, 79
 - Seattle, 25, 78
- Integer Programming, 27
- interference constraint, 9
- interference cost, 20, 36, 64
- interference level, 79
- interference matrix, 20, 79
- interferences, 2, 13, 14, 18–20, 23, 36

- highest-cost, 36
- Iterated Local Search, 34
- iterative local search, 48
- large-scale, 36, 37
- limit parameter, 124
- local optimum, 47, 48
- Local Search, 27, 33–35, 37, 38, 43, 48, 94, 95
- LSHR, 105
- mathematical formulation, 53, 158
- mathematical model, 7, 8, 14, 22
- MB-FAP, 10, 11
- metaheuristics, 26, 34, 35, 37
 - classes, 35
 - classification, 34, 74
 - population-based, 33–35, 38, 66, 74, 77, 146
 - single-objective, 33, 75, 94
 - trajectory-based, 33–35, 74, 77, 146
- MI-FAP, 10, 11
- minimum distance, 27
- minimum span, 11
- MMR, 79
- MO-ABC, 66, 68, 144, 146, 148, 157
 - tuning, 122
- MO-FAP, 10, 11
- MO-SVNS, 62, 66, 144, 148, 151, 157
 - tuning, 119
- MO-VNS, 60, 61, 66, 144, 146, 151, 157
- mobile network, 8, 36, 37, 78, 80
- mobile service, 8
- mobile user, 24
- MOEA, 29
- MS-FAP, 10, 11, 24
- multiobjective, 55
 - concepts, 51
 - formulation, 51, 63, 66, 157
 - metaheuristic, 5, 37, 51–53, 71, 89, 111, 155
 - optimization, 7, 22, 52–57
 - optimization problem, 29, 62, 156
- mutation probability, 94, 97, 111, 113, 124, 129, 135, 137
- mutation vector, 40
- neighbor sector, 83
- neighborhood, 46
 - environment, 60
 - solution, 38, 60, 65
 - structures, 46
- network
 - configuration file, 36
 - constraints, 87
- Neural Networks, 27
- nondominated front, 72
- nondominated individual, 57
- nondominated solution, 52, 54, 55, 60, 72, 73, 149
- nondominated sort, 72
- NP-hard, 8, 9, 28, 155
- NSGA-II, 5, 51, 57, 70, 71, 144, 148, 151, 157
 - tuning, 126
- NSS, 15
- number of interferences, 17
- number of iterations, 36
- objective function, 52
- operating system, 89
- operator
 - crossover, 38–40, 52, 58, 72
 - crowding, 70
 - greedy mutation, 62, 64, 65, 68
 - mutation, 38, 39, 52, 60–62, 72, 94, 97
 - non-greedy mutation, 62
 - selection, 38, 40, 57, 69, 72
- OPLINK, 25, 29
- optimal solution, 52
- optimization, 26
 - problem, 34, 51, 57
 - strategies, 30
- OSS, 15
- output file, 89–91
- packages, 86
- parallel genetic algorithm, 27
- parameter, 105, 144
 - probability, 68
 - tuning, 87, 90, 94, 103, 111, 119, 122, 126, 133, 157
- Pareto dominance, 56, 58, 63
- Pareto Front, 52, 53, 55, 65–67, 69, 70, 89, 91, 92, 116, 158
- Pareto optimal, 52, 53, 55, 66
- Pareto solution, 61, 66
- Pareto Tournament, 57–59, 66–70
- penalty, 87
- penalty cost, 36
- penalty values, 82
- performance, 89
 - analyses, 88
 - indicator, 91

population, 38, 56, 57
 ranking, 57
 size, 39, 57, 72, 94, 95, 111, 127, 135
 problem constraints, 14
 problem domain knowledge, 36, 37
 QoS, 1, 8, 14, 17, 19, 24, 37
 radio frequency, 12
 radio interference, 9, 24
 radio spectrum, 24, 26
 random frequency, 36
 ranking value, 69, 71, 72
 real scenario, 25
 real-world instance, 3, 4, 7, 18, 19, 36, 77, 78, 90
 real-world networks, 14
 reference point, 91, 92
 results comparison, 92
 reuse distance, 23
 scalar function, 53
 scale factor, 40, 98
 scaled difference vector, 39
 Scatter Search, 105, 108
 search
 process, 34
 space, 34, 35, 55
 Seattle topology, 79
 sector, 16, 17, 19, 20, 37, 81
 sector channel separation, 21
 selection
 best-so-far, 99, 113
 random, 99, 119
 uniform, 57
 selection tournament, 58
 separation constraint, 18
 separation cost, 64
 signaling threshold, 79
 Simulated Annealing, 27, 34
 single criterion, 52
 single-objective, 19, 27, 52, 89, 155
 optimization problem, 30
 site, 16, 21
 site channel separation, 21
 smaller cost, 38
 software, 85, 88, 91
 solution
 candidate, 60
 population of solutions, 34
 single solution, 34
 SPEA2, 5, 51, 57, 73, 144, 148, 151, 157
 tuning, 133
 SPP, 8
 statistical confidence, 90
 stop criterion, 38, 41
 strong dominance, 54
 SVNS, 33, 35, 48, 62, 106, 108, 156
 tuning, 103
 Tabu Search, 27, 34
 target individual, 58
 TCH, 16, 80
 transceivers, 16
 TDMA, 14
 telecommunication
 operator, 9
 telecommunication network, 13
 types, 24
 terrain morphology, 8
 tests sequence, 87
 traffic constraint, 9, 13
 traffic demand, 10, 12, 14
 transceiver, 19, 20
 trial individual, 58
 trial vectors, 39
 TRX, 15–21, 36, 38, 65, 78, 80
 tuning process, 77
 unused frequencies, 65, 68, 87
 valid frequency, 20, 36, 38, 84
 valid solutions, 36
 variants of FAP, 10
 VNS, 33–35, 46, 60, 62, 106, 108, 156
 weak dominance, 54
 wireless network, 25, 30
 XML, 87
 element, 88
 file, 88