Universidad de Málaga

Escuela Técnica Superior de Ingeniería de Telecomunicación

TESIS DOCTORAL

Self-Tuning Algorithms for the Assignment of Packet Control Units and Handover Parameters in GERAN

Autor:

Matías Toril Genovés

Directores:

Volker Wille
Iñigo Molina Fernández

Acknowledgments

Many people and institutions have contributed to this thesis, and, therefore, something of the following pages belongs to them.

My first and foremost debt of gratitude is to my supervisors, Volker and Iñigo, for their valuable time and contribution. In all these years, Volker has provided me with lots of interesting problems, which have been the main driver of this thesis and most of my research activity. Without his guidance, this thesis would not have been possible. More importantly, with his optimism and sense of humor, he has made this work worthwhile. In parallel, Iñigo has helped me to give the academic touch required in a scientific work like this. Words are not enough to thank them for the role they have played in this work.

I would like to thank Chris Walshaw for fruitful discussions on graph partitioning issues during my stay at the University of Greenwich. His valuable comments have helped to improve the computational aspects of this work and his hospitality made my stay at Greenwich an experience that I would be more than happy to repeat. I also wish to thank Sean McGrath and Jeroen Wigard for taking the time to read this document and, of course, for their recommendation letters.

I wish to thank my office mates, Francis and Miguel, for providing a stimulating and fun environment in which to spend part of my life. I also wish to express all my gratitude to my colleagues in the mobile research group. To Mariano, Fernando and Maribel, for their help in developing the simulation tool used in this thesis. To Raquel, for sharing with me Volker's time and topics. I am also grateful to my team mates in Nokia's venture, Salvador, Ricardo and Juanjo, with whom I shared one of the most hectic, yet most rewarding, experiences of my life. This would not have been possible without the invitation of our project leader, Carlos Camacho.

The financial support given by the TIC2003-07827 grant from the Spanish Ministry of Science and Technology, the research group TIC-102 Ingeniería de Comunicaciones and, more generally, the University of Málaga, is kindly acknowledged. I am also highly indebted to Nokia for giving me the opportunity to join one of the most prestigious research groups in the world and providing the data from live networks used in this thesis.

Finally, I wish to thank my entire family. To my wife and son, for their love and tolerance during long periods of neglect. To my parents, who taught me and loved me. To my grandmother and brother, for their happiness. They make my life worth living, and, to them, I dedicate this thesis.

Contents

Abstract

In the last years, cellular networks have undergone profound changes to cope with the increasing demand of mobile communication services. As a result, the size and complexity of these networks have increased considerably, which makes network management a very challenging task. In the past, operators have tackled this problem by incrementing their workforce, but, even so, it is difficult to configure the network to achieve optimal performance due to the sheer size of the network and its heterogeneity. Hence, operators demand automatic procedures and tools that help them to optimise network configuration.

This thesis deals with the problem of automatic optimisation of network parameters in GSM-EDGE Radio Access Network (GERAN). As the set of network parameters is extremely large, this work focuses on the main processes involved in mobility management: the cell (re)selection process for mobiles handling packet-data traffic and the handover process for mobiles conveying circuit-switched voice traffic.

To improve the performance of the cell (re)selection process, this work proposes methods to optimise the assignment of cells to packet control units (PCUs) in a base station controller. The main goal of these methods is to minimise the number of users of packed-data services that change their PCU, since this reduces the cell (re)selection delay. The PCU-assignment problem is formulated as a graph partitioning problem, which is solved by both exact and heuristic methods. To solve the problem exactly, this work proposes the use of a branch-and-cut algorithm over an integer linear programming model of the problem. As an approximate method, this work proposes the extension of the classical multi-level refinement method with adaptive multi-start techniques and connectedness checks.

In parallel, this work proposes methods to slowly modify HO parameters for traffic management purposes. The main goal is to relieve permanent local congestion problems by sharing traffic with adjacent cells. The classical diffusive approach, based on the adjustment of handover margins, is extended here with the optimisation of handover signal-level constraints. The combination of these two heuristic strategies in a single method, implemented with fuzzy logic, circumvents the problems of the classical approach associated to the loss of network quality.

All the methods proposed in this thesis are based on statistical information. Consequently, these methods are conceived as network re-planning procedures, which could be applied regularly during network operation. Likewise, the proposed methods deal with parameters and performance indicators that are currently available in the network management system, thus requiring no change in network infrastructure.

Performance assessment is based on a combination of field tests and computer simulations. Preliminary field trials aim to show the need for the optimisation process by showing how a simple method can greatly improve current network performance. More sophisticated methods

are then tested, based on analytical models constructed from data of a live network or simulation models that intend to reflect a realistic scenario.

Resumen

En los últimos años las redes de telefonía móvil han experimentado cambios sustanciales para hacer frente a la demanda creciente de servicios de comunicaciones móviles. Como resultado, el tamaño y la complejidad de este tipo de redes se ha incrementado notablemente, dificultando sobremanera las tareas de gestión. En el pasado, los operadores han solventado este problema aumentando su plantilla, pero, aun así, es difícil asegurar una configuración óptima de la red debido a su tamaño y heterogeneidad. Por ello, los operadores demandan cada día más herramientas automáticas de optimización de red.

Esta tesis aborda el problema de la optimización automática de parámetros en redes de acceso radio basadas en *GSM-EDGE Radio Access Network* (GERAN). Dada la extensión del conjunto de parámetros que se puede optimizar, este trabajo se centra en dos de los procesos encargados de la gestión de la movilidad: el proceso de (re)selección de celda para servicios por conmutación de paquetes y el proceso de traspaso para servicios de voz por conmutación de circuitos.

Para mejorar el rendimiento del proceso de (re)selección de celda, este trabajo propone métodos que optimicen la asignación de celdas a unidades de control de paquetes (*packet control units*, PCUs) en un controlador de estación base. El principal objetivo de estos métodos es minimizar el número de usuarios de servicios de transmisión de paquetes que experimentan un cambio de PCU, ya que con ello se minimiza el retardo del proceso de (re)selección. Para conseguir dicho objetivo, el problema de asignación de PCUs se formula como un problema de partición de grafos, para el que se estudian métodos de resolución tanto exactos como aproximados. Para su resolución exacta, se propone el uso del método de ramificación y corte sobre un modelo de programación lineal entera del problema. Como método aproximado, se propone la extensión del método clásico de refinamiento multi-nivel con técnicas multi-arranque adaptativas y chequeos de conectividad.

De manera paralela, este trabajo propone la optimización de los parámetros de traspaso como estrategia de gestión del tráfico. Para ello, se estudian métodos de control del área de servicio de las celdas de una red celular mediante la modificación de los márgenes del traspaso. El principal objetivo es la resolución de problemas de congestión local compartiendo la demanda de tráfico entre celdas vecinas. La estrategia clásica de balance de carga por difusión, basada en la modificación de los márgenes de traspaso, se extiende aquí con la optimización de las restricciones de nivel de señal en el traspaso. La combinación de estas dos estrategias heurísticas en un único método, implementado mediante lógica difusa, solventa las limitaciones debidas a la pérdida de calidad en la red.

Todos los métodos propuestos en esta tesis se basan en información estadística. Consecuentemente, estos métodos están concebidos para ser utilizados en los procedimientos de replanificación de la red, ejecutados de manera periódica durante la fase de operación. Asimismo, los métodos propuestos manejan parámetros e indicadores de rendimiento disponibles en los sistemas de gestión de red actuales, no exigiendo cambios en la infraestructura de red.

La validación de los métodos se basa tanto en pruebas de campo como en simulaciones por ordenador. Las pruebas de campo realizadas inicialmente sobre una red real tratan de justificar la necesidad del proceso de optimización, comprobando cómo un método simple mejora significativamente el rendimiento actual de la red. Posteriormente, se analizan métodos más sofisticados sobre modelos analíticos construidos a partir de medidas de una red real o modelos de simulación que reflejen situaciones realistas.

List of Abbreviations

AMS	Adaptive Multi-Start
BB	Branch-and-Bound
BC	Branch-and-Cut
BCCH	BroadCast CHannel
BER	Bit Error Rate
BFS	Breadth-First Search
BH	Busy Hour
BR	Blocking Rate
BSC	Base Station Controller
BSS	Base Station System
BTS	Base Transceiver Station
CAC	Call Access Control
CAMS	Clustered Adaptive Multi-Start
CDF	Cumulative Distribution Function
CPAP	Cell-to-PCU Assignment Problem
CR	Congestion Rate
CRS	Cell Re-Selection
DFS	Depth-First Search
DTX	Discontinuous Transmission

ECDF	Empirical Cumulative Distribution Function
EDGE	Enhanced Data rates for Global Evolution
EIRP	Equivalent Isotropic Radiated Power
EL, EH	Extremely Low, Extremely High
ES	Edgecut-based Sharing
FEP	Frame Error Probability
FER	Frame Error Rate
FIS	Fuzzy Inference System
FM	Fiduccia-Mattheyses
FOM	Figure Of Merit
FOSLC	Fuzzy Optimisation of Signal-Level Constraints
FSHMC	Fuzzy Slow HO Margin Control
FTP	File Transfer Protocol
FW-GGGP	Floyd-Warshall Greedy Graph Growing Partitioning
GERAN	GSM/EDGE Radio Access Network
GGGP	Greedy Graph Growing Partitioning
GK	Graph-Walking
GOS	Grade Of Service
GPP	Graph Partitioning Problem
GPRS	General Packet Radio Service
GR	Greedy Refinement
GSM	Global System for Mobile communications
H	High
HEM	Heavy-Edge Matching

HO	Hand-Over
HOC	Hand-Over Control
ILP	Integer Linear Programming
IO	Initial Operator
KKT	Karush-Kuhn-Tucker
KL	Kernighan-Lin
KPI	Key Performance Indicator
L	Low
LA	Location Area
M	Medium
ML	Multi-Level
MR	Measurement Report
MS	Multi-Start
NH	Non-Hopping
NP	Non-Polynomial
OAM	Operational, Administration and Maintenance
OMNI	OMNI-directional
OR	Outage Rate
RO	Refined Operator
OSLC	Optimisation of Signal-Level Constraints
PCU	Packet Control Unit
PBGT	Power BudGeT
PDF	Probability Density Function
POC	POwer Control

QOS	Quality Of Service
RA	Routing Area
R-GGGP	Random Greedy Graph Growing Partitioning
RH	Random hopping
RLB	Recursive Level Bisection
RM	Random Matching
RRM	Radio Resource Management
RXLEV	Received signal LEVel
RXQUAL	Received signal QUALity
SACCH	Slow Associated Control CHannel
SGSN	Serving GPRS Support Node
SHEM	Sorted Heavy Edge Matching
SHMC	Slow HO Margin Control
SM	Site Matching
SR	Scratch and Remap
SS	Size-based Sharing
TCH	Traffic CHannel
TRI	TRI-sectorised
TSL	Time SLot
VL, VH	Very Low, Very High

List of Symbols

$\mathbf{A}$	Offered traffic vector
a, $\widehat{a}$	Location parameter of Gumbel distribution and its estimation
a_f	Activity factor
A_i, A_{bi}, A_{ci}	Offered, blocked and carried traffic in cell i
a_{ij}	Element of the adjacency matrix
$\mathbf{AM}$	Adjacency matrix
A_{lbi}, A_{ubi}	Lower and upper bound for offered traffic in cell i
A_T, A_{bT}, A_{cT}	Total offered, blocked and carried traffic
B_{aw}, B_{rw}	Maximum absolute weight, maximum relative weight of a subdomain
BR, BR_i	Blocking rate, blocking rate in cell i
$\overline{BR}$, $\overline{BR}_t$	Overall blocking rate, overall blocking rate target
b, $\widehat{b}$	Scale parameter of Gumbel distribution and its estimation
$\mathbf{c}$	Cell capacity vector
c	Minimum average number of vertices per subdomain in the coarsest level
c_i	Number of channels in cell i
$\frac{C}{I}$	Carrier-to-interference ratio
D	Number of network dimensions
$D_{crs}(x, y)$	Dominant cell in position (x,y) defined by CRS

$D_{cvg}(x,y)$	Dominant cell in position (x,y) based on coverage
$D_{ho}(x,y)$	Dominant cell in position (x,y) defined by HO
$d(i,j)$	Distance between vertices i and j in a graph
$d(x,y)$	Distance between points x and y in a map
$d(\Pi,\Psi)$	Distance between partitions Π and Ψ
$diag(AM)$	Diagonal of adjacency matrix
E	Set of graph edges
e	Exponent in penalties to penalise non-fulfilment of objectives
$E(A_i,c_i)$	Blocking probability given by the Erlang-B formula
E_j	Set of edges in problem instance j
$E^{(i)}$	Set of edges in the coarsened version of the graph in level i
$e(i,j)$	Edge from vertex i to vertex j
$E(i)$, $\lvert E(i)\rvert$	Edges incident to vertex i, degree of vertex i
$F_n(z)$	Empirical cumulative density function of random variable z
$FER_{\max}$	Maximum FER for acceptable connection quality
G, G'	Graph, subgraph
g	Number of solutions per generation in CAMS algorithm
$G^{(i)}$	Coarsened version of graph G in level i
h	Time horizon
$\overline{HR}$	Overall HO rate
k, k_j	Number of subdomains, number of subdomains in problem instance j
L	Average traffic load
L_b	Basic transmission loss
M	Matching

m_s	Sample mean
MCD	Mean call duration
MHT	Mean holding time
N	Number of cells
$\overline{N_a}$	Average number of attempts until success
N_{bi}	Number of blocked call attempts in cell i
N_{const}, N_{var}	Number of constraints and variables
$N_{f\,i,j}$	Number of frequencies shared by cells i and j
$N_{fc}(A_i, c_i)$	Average number of free channels with offered traffic A_i and c_i channels
$N_{hoi \to j}$	Number of HOs from cell i to cell j
N_i	Number of call attempts in cell i
N_p	Number of problem instances
N_r	Number of rules
N_s	Number of samples
N_T , N_{cT}, N_{hT}, N_{hoT}	Total number of call attempts, carried calls, connections and HOs
N_{tsi}, N_{trxi}	Number of time slots and transceivers in cell i
N_{mri}	Number of MRs in cell i
$N_{mri}\|_{FER \geq FER_{\max}}$	Number of MRs with unacceptable FER in cell i
$\mathcal{NP}$	Family of non-deterministic polynomial problems
$O(f(n))$	Set of functions that grow no faster than $f(n)$
O_{fis}	Output of fuzzy inference system
o_l	Output of rule l
OR, OR_i	Outage rate, outage rate in cell i
$\overline{OR}$, $\overline{OR}_t$	Overall outage rate and overall outage rate target

P, P'	Total penalty, overall penalty
$\mathcal{P}$	Family of polynomial problems
p, p_k	Penalty, penalty in epoch k
$p_{ci \to j}$	Probability of interfering collision of cell i to cell j
$P(e)$	Probability of event e
p_j	Problem instance j
$P_{rx}(x, y, i)$	Signal-level received from cell i in position (x,y)
p_s	Probability of success
q	Convergence order
Q_j	Edge-cut of a heuristic solution to problem instance j
Q_ρ	Normalised edge-cut from algorithm with intensity ρ
$\overline{Q_{\rho,i}}$	Average edge-cut from algorithm with intensity ρ over instance i
Q_{ρ_T}	Total edge-cut from algorithm with intensity ρ
$Q_{\rho,i,r}$	Edge-cut from run r of algorithm with intensity ρ over instance i
R^2	Square of sample (or Pearson product-moment) correlation coefficient
r	Run of an algorithm
$\|\|r\|\|$	Norm of the residual of an equation system
r_c	Cell radius
S, s_{qr}	Similarity matrix, element of the similarity matrix
$s_l(\mathbf{X})$	Strength of rule l with input vector $\mathbf{X}$
T	Runtime
T_h, T_{hT}	Channel holding time and total channel holding time
$T_{\rho,i,r}$	Runtime of attempt r of an algorithm with intensity ρ over instance i
T_{ov}	Total runtime dedicated to the optimisation of a set of instances

T_{ρ}	Normalised runtime from algorithm with intensity ρ
$\overline{T_{\rho,i}}$	Average runtime from algorithm with intensity ρ over instance i
T_{ρ_T}	Total runtime from algorithm with intensity ρ
$T_{\rho,i,r}$	Runtime of run r of algorithm with intensity ρ over instance i
T_{ssi}, T_{esi}	Time dedicated to the optimisation of instance i in SS and ES strategies
$T(z)$, $T^{-1}(z)$	Cumulative density function and percent point function of random variable z
V	Set of graph vertices
V_i	Subdomain i in a graph
$V^{(i)}$	Set of vertices in the coarsened version of the graph in level i
$V(i)$	Adjacent vertices (neighbour cells) to vertex (cell) i
$V(V_i)$	Adjacent vertices to subdomain V_i
υ, υ_i	Vertex, vertex i
υ_{ms}	User speed
$\mathbf{X}$, $\mathbf{x}^*$, x_i	Solution vector, optimal solution vector, component i of solution vector
$x^{(i)}$	Element i in an ordered series of values of variable x
X_{in}	Binary variable that reflects the assignment of vertex i to subdomain V_n
$\mathbf{X}_{LP}^{j}$	Optimal solution to the LP relaxation of the ILP problem j
$\overline{z}_{LP}$	Upper bound for z_{LP}
z^j	Optimal value of the solution to problem j
z_{LP}^{j}	Optimal value of the LP relaxation of the ILP problem j
α, $\alpha_{i \to j}$	Significance level, outage probability in adjacency (i,j)
β	Diffusive constant
γ	Discount factor
γ_{ij}	Weight of edge (i,j)

Δ	Density of a graph
$\Delta HoMarginPBGT_{i\rightarrow j}^{(k)}$	Deviation of $HoMarginPBGT_{i\rightarrow j}$ from default values in iteration k
$\delta(V_1,...,V_k)$	Set of edges in the cut defined by subdomains V_1,...,V_k
δ	Maximum step of PBGT HO margins
$\delta HoMarginPBGT_{i\rightarrow j}^{(k)}$	Step of PBGT HO margins in adjacency (i,j) and iteration k
η	Convergence rate
$\Theta(f(n))$	Set of functions that grow at the same rate as $f(n)$
λ_{ci}, λ_{f_i}, λ_{hoi}	Connection, new call and HO arrival rate in cell i
λ_T	Total call arrival rate
μ, μ_i	Call service rate and call service rate in cell i
μ_c, μ_{ci}	Connection service rate and connection service rate in cell i
$\mu_{ij}(x_i)$	Membership degree of crisp input i to fuzzy variable j
ξ	Limit value of a series
$\mathbf{\Pi}$, π_i	Partition, subdomain to which vertex i is assigned
ρ	Intensity of an optimisation algorithm
σ_s	Sample standard deviation
σ_{sf}, σ_{traf}	Standard deviations of slow fading and spatial traffic demand variables
Φ	Lagrangean
ω_{br}, ω_{or}	Relative weights of blocking rate and outage rate criteria
$\Omega(f(n))$	Set of functions that grow at least as fast as $f(n)$
ω_i	Weight of vertex i
$\|A\|$	Number of elements in A
$\|\|A\|\|$	Sum of the weight of elements in A
$\#(A)$	Number of samples where condition A is true

Introduction

This opening chapter aims to show the relevance of the topics covered in this thesis, present the research scope and methodology, and describe the document structure.

Motivation

In recent years, mobile communication networks have undergone profound changes. To cope with the increasing demand of mobile services, new elements have been continuously added to the network. At the same time, new technologies and services have been introduced to satisfy user expectations and outperform the competition. As a result, the size and complexity of mobile networks have increased dramatically. To complicate matters further, the mobile environment is constantly changing, which often requires re-configuring the network. All these issues have made it very difficult for operators to manage their networks at a high performance level.

In the past, mobile network operators have dealt with this problem by increasing the workforce and overdimensioning network resources. However, with the increased level competition, such an approach is not feasible anymore due to work effort and expenses involved. This is especially true for mature technologies, such as GSM/EDGE Radio Access Network (GERAN), for which operators aim to reduce operational and capital expenditures as much as possible. Hence, in these radio access technologies, an efficient network management is crucial for operators to provide high-quality services at a low operational cost.

To increase operational efficiency, network management has been progressively automated. The aim of automation is two-folded: on the one hand, to relief personnel from tasks that are performed manually and must be repeated either geographically (several times in different parts of the network) or periodically (often in the same part of the network); on the other hand, to introduce new procedures that, based on the analytical capabilities of computers, increase network performance by modifying network parameters[1]. This fact has stimulated intense research activity in the field of *self-tuning* (or *auto-tuning*) networks [1][2][3]. In this context, the self-tuning property refers to the capability of a network to adjust its parameters to achieve optimal performance without human intervention.

Different approaches can be followed to tackle the problem of modifying parameters in a cellular network. On the one hand, equipment manufacturers focus their efforts on developing advanced self-tuning features that could be used to upgrade the capabilities of existing equipment. These features are able to modify parameters of network elements in real time, based on

[1] In this work, the term *automation* is used to refer to procedures that mimic the action of an operator, while the term *optimisation* refers to procedures that cannot be performed manually due to complexity.

instantaneous measurements. On the other hand, operators aim to develop self-tuning methods based on the capabilities of existing equipment. In this approach, statistical performance measurements are gathered periodically in the network management system. From the analysis of these data, a proposal of parameters change is designed, which is implemented in the form of parameter files that are downloaded into the network. Due to effort and expenses, it is clear that the former approach becomes less appealing as a radio access technologies becomes mature. As operators avoid any investment that might be difficult to amortise given the short time horizon, manufacturers focus their efforts on emerging radio access technologies. Hence, the planning of parameters from the network management system plays an increasingly important role for operators in mature technologies.

The set of network parameters that can be optimised in GERAN is extremely large and the scope of this thesis therefore had to be limited to important issues. This work focuses on the automatic optimisation of parameters in two processes: (i) the cell (re)selection process, and (ii) the handover process. Several reasons justify the selection of these processes. Firstly, both processes have a strong impact on cellular network performance, as they are related to user mobility. The cell (re)selection process has a strong influence on the quality of packet-data services in existing networks, which lack the support of real-time packet-data services [4]. Thus, the cell (re)selection delay is strongly affected by the assignment of cells to packet control units (PCUs). Likewise, the handover (HO) process is a major contributor to network quality, as it ensures that every user is constantly served by the best cell [5]. Secondly, self-tuning methods suggested so far are either infeasible or ineffective. In the case of the assignment of packet control units, although manufacturers provide automatic configuration procedures, operators hardly ever use them due to poor results. In the case of HO parameter tuning, several automatic methods have been proposed. However, these either rely on tools that are not currently available for operators [6][7] or they do not guarantee good performance [8][9]. As a result, operators are often forced to optimise parameters in the previous processes manually. Unfortunately, the complexity and effort of the analysis task prevents operators from doing this regularly. For this reason, optimising network parameters is hardly ever considered and safe parameter settings are normally adopted, even if this results in sub-optimal performance [1]. It can thus be concluded that any method that optimises parameters in the previous processes can potentially improve the performance of GERAN networks. If these methods can be implemented in software, then it is also possible to offer these solutions cost-efficiently and operators will be able to use them.

Research Objectives

The main goal of this thesis is to develop automatic optimisation procedures for parameters in the above-mentioned processes that can be implemented with the existing network infrastructure. More specifically, this thesis aims to:

a) Develop methods for optimising the cell-to-PCU assignment based on statistical network measurements to reduce the number of users that change their PCU in GERAN, and

b) Develop methods for optimising HO margins and signal-level constraints based on statistical network measurements to solve localised congestion problems in GERAN.

The main contributions of this thesis can be summarised as follows:

- Regarding the cell-to-PCU assignment problem, the problem is formulated for the first time as a graph partitioning problem. Having identified the underlying problem, several classical graph partitioning techniques are adapted to deal with the peculiarities of the mobile environment. Two novel methods are proposed: an exact method, based on an enumerative approach, and an approximate method, based on the combination of several heuristic strategies. While the former is used for benchmarking purposes, the latter can be used to obtain good solutions efficiently.

- Regarding the optimisation of HO parameters, the classical diffusive algorithm that balances the load between adjacent cells by changing HO margins temporarily is adapted here for its use on the network management system. A novel heuristic method is then proposed to jointly optimise HO margins and signal-level constraints to circumvent the limitations of the previous approach. By considering quality issues explicitly, HO margins can change safely in a wider range, thus extending the capabilities of the method.

A distinctive feature of this work is the consideration of practical aspects that are often neglected in research work. The proposed methods take into account the constraints of existing vendor equipment and can thus be implemented without much effort by operators. During algorithm design, operator's constraints are also taken into account, paying special attention to the ease of managing the new solutions. With this mind-set, the evaluation process is based on real network data. Thus, the need for the optimisation process is first justified by testing a very simple algorithm on a live network. Then, a comprehensive analysis of more sophisticated methods is performed over models built from data taken from a live network or realistic simulation scenarios. This approach has been possible thanks to Nokia, who interacted with network operators and has provided the real network data used in this thesis.

Research Methodology

Most scientific work begins with the problem formulation based on a qualitative description. After understanding the problem, the state of research and practice is analysed. Having identified the shortcomings of current approaches, new methods are conceived. A first evaluation of these methods is performed over simple test cases, often in simulation environments. Once these methods are successfully tested by the research community, manufacturers evaluate the performance benefit and development effort. In the subsequent development stage, simplifications are normally adopted for the sake of efficiency. Finally, these methods are deployed in the network, where performance data is available on a regular basis. Thus, the real benefit of a method can finally be assessed based on statistical performance data from real networks. Such an approach has also been adopted in this thesis. However, this work has several peculiarities that are worth mentioning:

a) This work aims to solve problems faced by operators during network re-planning procedures. Having this in mind, the initial formulation of the problem in qualitative terms was performed by the operator. This included the main parameters to be optimised and the relevant assessment criteria.

b) From this general description, the problem was formulated analytically to understand the relationship between network parameters and performance data. Once the type of problem was identified, the search focused on methods from other application areas that could be applied to the problem.

c) To avoid unnecessary effort, the initial focus was on methods that could be quickly tested on a limited area of a live network. The aim of these field trials was twofold: on the one hand, to evaluate the sensitivity of network performance to the modification of selected parameters; on the other hand, to have a lower bound on performance benefit from the optimisation process. Only when it was proved that a simple method had a significant impact on network performance, more sophisticated methods were considered. This fact justifies that the validation of the methods proposed in this work started with a field trial, in contrast to the classical approach, which often starts with simulations over a test case.

d) Once the potential of the technique was verified, research focused on more sophisticated methods. These were tested on analytical models built from network data or realistic simulation scenarios. Ideally, the final assessment should have been performed in a live network. Unfortunately, operators are often reluctant to test complex methods that modify parameters that have a great impact on network performance. Nonetheless, it is expected that the good results shown by the test cases will also be seen in a live network, because (i) the test cases are representative of a real situation, (ii) the basic techniques have been validated in a live network, and (iii) the proposed algorithms are rather intuitive.

Document Structure

From the research goals, it is clear that this thesis deals with two different problems, which could have been treated independently. For this reason, both topics are covered in separate chapters. Nonetheless, an effort has been made to give both problems a unified treatment. With this idea in mind, the rest of the document is organised in four chapters: a preliminary chapter that presents the conceptual framework of this thesis, two chapters with a similar structure devoted to each of the problems and one concluding chapter.

Chapter 1 introduces the problem of automatic parameter optimisation in cellular networks, defining some basic terminology that will be used throughout the document.

Chapters 2 and 3 are devoted to the cell-to-PCU assignment problem and the optimisation of HO parameters in GERAN, respectively. Both chapters begin with a brief description of the problem from the operator's perspective, followed by an analytical formulation of the problem and the state of research and technology. The proposed solution techniques are then described, from the basic to the most advanced. Preliminary field trial results are presented, which are later extended with the analysis of more sophisticated methods over analytical or simulation models. Finally, the main performance results are highlighted.

Based on the conclusions of the previous chapters, Chapter 4 summarises the major findings of this research, emphasising the original contributions and presenting several possibilities to extend this work in the future.

Three appendices are included at the end of this thesis. Appendix A presents the runtime analysis of the heuristic method proposed in Section 2.3.3. Appendix B derives the optimality conditions for the traffic sharing problem presented in Section 3.3. Finally, Appendix C is a brief summary of the thesis in Spanish.

Chapter 1

Automatic Parameter Optimisation in GERAN

This chapter introduces the problem of automatic optimisation of cellular networks. For clarity, the chapter begins by describing the cellular network architecture that is assumed in this work. The self-tuning concept is then presented, discussing several approaches to deploy and evaluate self-tuning algorithms in live cellular networks. Finally, the following chapters are put into context.

1.1 GERAN Architecture

In this work, the term *GSM-EDGE Radio Access Network* (GERAN) is used to refer to a radio access network that is based on *Global System for Mobile communications* (GSM), *General Packet Data Services* (GPRS) and *Enhanced Data rates for Global Evolution* (EDGE). Historically, GERAN is based on GSM/EDGE Release 99, covering all new features that has been launched for GSM ever since, with full backward compatibility to previous releases [10].

Figure 1.1 displays the reference architecture of GERAN A/Gb mode assumed in this work, which is the one deployed in existing networks. In the figure, it is observed that a cellular network mainly consists of three components: the Base Station System (BSS), the Network and Switching System (NSS) and the Operations Support System (OSS).

The BSS is in charge of providing the path between the Mobile Stations (MSs) and the fixed network infrastructure, containing the elements that are specific to radio cellular networks. The BSS consists of the Base Transceiver Station (BTS), Base station Control Function (BCF) and the Base Station Controller (BSC). The BTS in current network comprises radio transmission and reception devices. The BCF is a logical entity that groups BTSs that are on the same site (and, consequently, share some equipment). Finally, the BSC is responsible for the control of a group of BTSs and BCFs. This responsability covers algorithms that must be executed on a per-cell or per-connection basis related to Radio Resource Management (RRM).

The NSS manages the communications between the mobile users and other users, whether mobile or not. The NSS also includes databases to store information about the subscribers and to manage their mobility. Among other components are the Mobile services Switching Centre (MSC) and the Serving GPRS Support Node (SGSN). While the former is in charge of

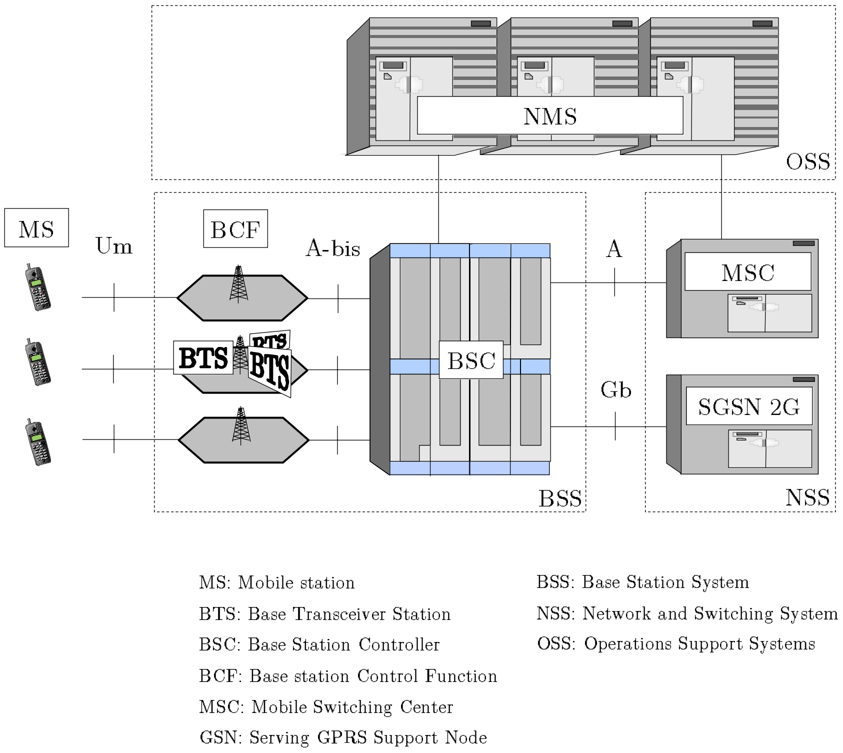

NMS
OSS
MS
Um
BCF
A-bis
BTS
BTS
BTS
BSC
A
MSC
Gb
SGSN 2G
BSS
NSS
MS: Mobile station
BTS: Base Transceiver Station
BSC: Base Station Controller
BCF: Base station Control Function
MSC: Mobile Switching Center
GSN: Serving GPRS Support Node
BSS: Base Station System
NSS: Network and Switching System
OSS: Operations Support Systems

Figure 1.1: The reference architecture of GERAN A/Gb mode.

switching functions related to Circuit-Switched (CS) connections, the latter does the same for Packet-Switched (PS) connections.

Finally, the OSS takes charge of Operation, Administration and Maintenance (OAM) tasks. The core of the OSS is the Network Management System (NMS), which is a set of computers that has been setup to control the devices in the network. For this purpose, it is connected to network elements in the NSS and the BSS to monitor performance and modify configuration parameters. Both past performance measurements and current configuration data are stored in separate databases, which are accessed frequently by maintenance personnel.

Regarding the interfaces, the A interface of the GSM standard is used for the exchange of user and signalling data related to CS services between the BSC and the MSC. The Gb interface is added in GERAN A/Gb mode for data exchange between the GSM radio network and the GPRS part. The latter interface is the carrier of PS traffic between the BSC and the SGSN.

1.2 The Self-Tuning Concept

With the increased level of competition, cellular network operators are facing the challenge of providing high-quality services at a minimum cost. In this context, it is crucial to maximise network performance with the existing infrastructure, which can be achieved by optimising network parameters. Although several definitions exist for what a good performance is, from the operator perspective, a better performance means an improvement of network capacity, while maintaining or improving the quality-of-service (QoS) offered.

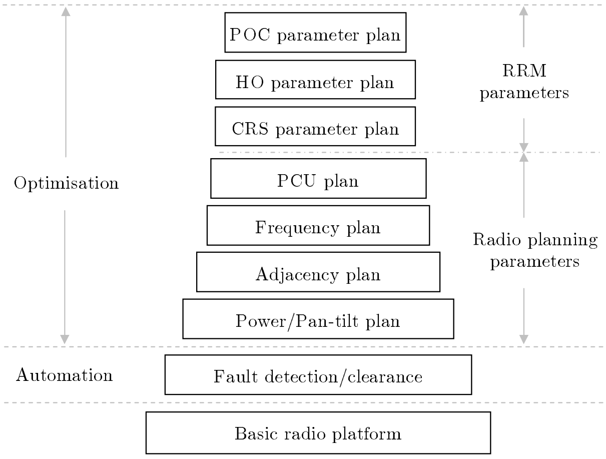


Figure 1.2: Automatic optimisation of cellular networks.

Figure 1.2 displays several factors that have a major impact on the performance of a cellular network, and can therefore be optimised [1]. The pyramidal structure aims to reflect the sequence of actions that is followed by the operator during the optimisation of a live network. Departing from a well-designed radio platform, the optimisation process starts by ensuring that the network is fault-free. The fault detection process may range from simple analysis of alarms to complex consistency checks to identify a bad configuration of network elements. Once faults have been detected and cleared, the optimisation process may proceed to the adjustment of physical BTS parameters, such as the antenna down-tilt or the maximum transmitted power. The adjacency[1] and frequency plans in the network can then be improved. After updating adjacency definitions, the assignment of cells to PCUs in a BSC can be optimised in the PCU plan. Finally, the parameters of RRM algorithms in the BSC, such as cell (re)selection (CRS), handover (HO) and power control (POC), can be tuned to obtain optimal performance.

Most of the previous actions aim to optimise the network by changing parameter settings. However, not all parameters are equally easy to change. While changing some radio planning parameters might require site visit and climbing the antenna, changes in RRM parameters can be performed remotely from the NMS or BSC site. Even in the latter category, some parameters can be changed on the fly, while others require BTS locking, which is only possible in low traffic periods (i.e., at night). For obvious reasons, operators normally prefer to modify RRM parameters that do not require time scheduling.

Unfortunately, the RRM parameter set is extremely large, as there are lots of algorithms running and some parameters are defined on a per-cell, per-adjacency or even per-transceiver basis. For instance, a typical GSM BTS comprises more than 200 parameters related to CRS, HO and POC, of which one tenth are HO parameters that are duplicated for each neighbour cell. Even if optimisation is only performed on a few relevant parameters, the relationship between parameter settings and network performance is not obvious, which makes analysis difficult. Hence, the optimisation process cannot be performed manually, but must be automated. Thus, the network would be able to regulate its parameters to achieve optimal performance without human intervention. Such a property is referred to as *self-tuning* (or *self-regulation*) capability.

[1] In this work, the terms *neighbour cell* and *adjacency* are interchangeably used. While the former refers to the cell that receives HOs from a source cell, the latter refers to the entity used in the NMS to reflect that HOs are allowed between a pair of cells.

1.3 Self-Tuning Approaches

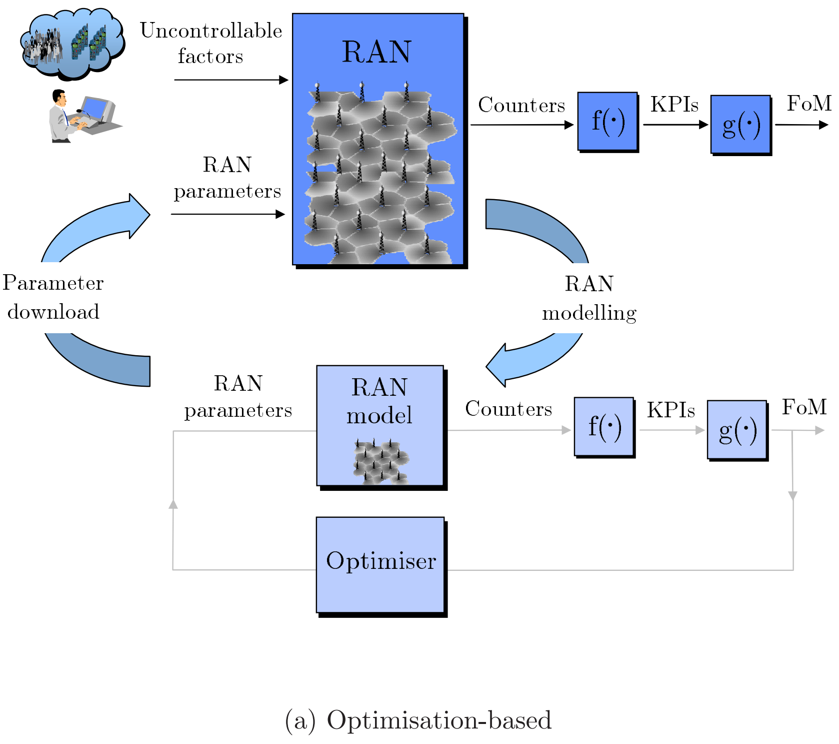
The following paragraphs present different approaches to automate parameter tuning in a cellular network. The approaches are classified according to several criteria in order to establish some basic terminology that will be used throughout the document.

1.3.1 Optimisation vs Control

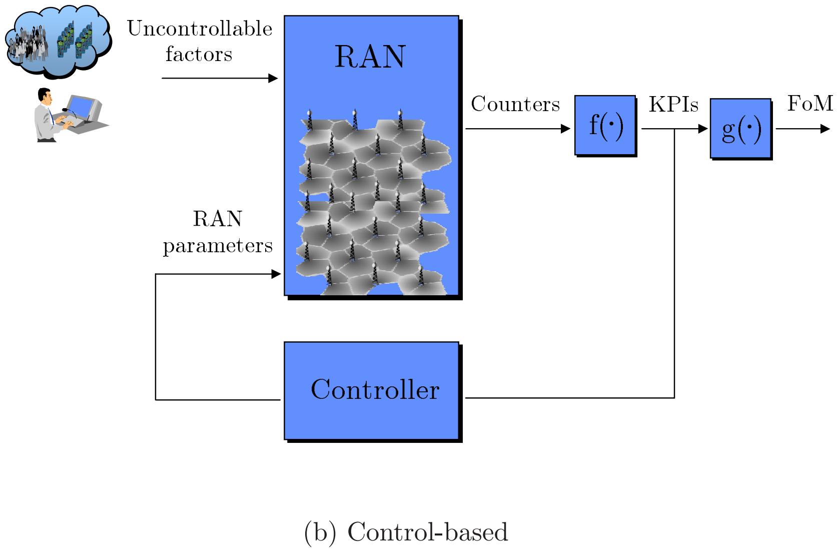
In most practical optimisation work, the search of an optimal configuration is performed over a simplified model of the system to be optimised. Such a model is used to find the best parameter settings, which are later extrapolated to the real system. Figure 1.3 (a) illustrates the main principles of this approach. The process begins with the construction of a model that describes the relationship between input and output variables. For this purpose, the system inputs and outputs are defined first. Two different types of RAN inputs are distinguished: *parameters* (i.e., decision variables) and *external disturbances* (i.e., uncontrolled variables). The RAN output consists of a set of counters associated to particular events, which are used to build the *key performance indicators* (KPIs). Several of these KPIs are combined to calculate the final *figure-of-merit* (FoM), based on operator's preferences among different criteria. Only then, the relationship between parameters and FoM is established by means of an analytical or simulation-based model. The next step is to find the parameter settings that give the best FoM, given certain constraints on KPIs. For this purpose, a naive trial-and-error strategy can only be used when the search space is small (e.g., single parameter with a limited set of feasible values). As this is not often the case, a classical optimisation method is normally used (denoted in the figure as the *optimiser*). Finally, the best parameter settings found in the model are downloaded into the network. Such a self-tuning approach will be referred to as *optimisation-based*, as it relies on the application of an optimisation method on a network model without any restriction on use.

In the previous approach, the key process is the construction of a network model that is both accurate and manageable. For complexity reasons, it is often impossible to build an analytical model that describes the relationship between performance criteria and parameters precisely. In the absence of such a mathematical model, a simulation model might capture the dynamic behaviour and randomness in the system. Unfortunately, the accuracy of these models is not always sufficient. In mobile networks, these inaccuracies come from the difficulty of predicting propagation mechanisms and traffic distribution in a cellular environment. As a result, the best parameter settings found in the simulation might lead to sub-optimal performance when implemented in the real network. In these conditions, a simulation model should only be considered as a network instance, where the performance of optimisation methods can be assessed, but never a precise model from which the optimal settings of a particular network can be derived.

During the operational stage, network measurements can be used to refine the system model over which optimisation is carried out, whether analytical or simulation-based. Thus, it is more likely that optimal parameter settings are found. However, even if this is the case, the network is subject to changes caused by uncontrollable factors. On the one hand, the spatial traffic distribution varies both in the short and the long term. On the other hand, network re-configuration carried out by the operator can cause that the old parameter settings become ineffective. To cope with these changes, the network model must be updated periodically (or, at least, after any significant change in the network). Consequently, the optimisation method



(a) Optimisation-based



(b) Control-based

Figure 1.3: Self-tuning approaches for cellular radio access networks.

must also be run periodically. In the literature, these methods appear under the name of *measurement-based re-planning*, examples of which can be found in antenna down-tilt planning [11], adjacency planning [12], frequency planning [13] and HO parameter planning [6][7].

Unfortunately, it is rather unusual that precise measurement data is available. To circumvent the need of a model, the optimisation algorithm can interact directly with the network. In these conditions, a closed-loop structure can help to deal with the optimisation problem. Figure 1.3 (b) shows the basic structure of a closed-loop control system. In these systems, a controller regulates network parameters by comparing the values of KPIs with some reference values (referred to as *setpoints*). This approach circumvents the need of a precise model. Likewise, the sensitivity to external inputs is reduced with the use of feedback. As a result, good performance is maintained, regardless of model errors and changes in the traffic demand or propagation environment. For obvious reasons, such a self-tuning approach will be referred to as *control-based*.

In the latter methods, the tuning process is performed by experimenting directly with the real network. This fact imposes several limitations on the tuning process. As no service degradation is accepted during operation, operators only allow subtle variations to the default parameter settings, which severely limits the search for an optimal solution. At the same time, this approach does not guarantee optimal performance if the reference values are not selected properly. Thus, it might happen that the control process might lead to a network configuration that is worse than the initial state, as the FoM is not directly taken into account during the tuning process. To stress this fact, these methods will be referred to in this work as *regulation* (or *tuning*) methods, even if their goal is to optimise network performance, and can thus be broadly classified as optimisation methods. Nonetheless, a sensible choice of setpoints usually leads to a solution that is better than the initial one, which is normally enough to solve network problems and satisfy the operator. In spite of these limitations, these methods are widely used by operators due to their simplicity. Examples of these are the HO parameter tuning methods proposed in [8] and [9], which are the starting point of this work.

1.3.2 On-line vs Off-line

The frequency with which a tuning method can be applied to the network depends mainly on three factors: the frequency with which measurements are gathered, the delay to update network parameters and the computational load of the calculation process. All these issues are given by the network equipment that performs these tasks.

The first equipment that can manage the changes of RRM parameters is the BSC. In GERAN, the BSC is in charge of algorithms that run on a per-connection or per-cell basis. As this equipment deals with instantaneous performance indicators, it is also given the capability to change parameters quickly (i.e., typically, in the order of seconds). Thus, advanced network features in the BSC can change RRM parameters quickly, to cope with fast changes of traffic and interference conditions. Obviously, the complexity of these methods is constrained by the short time response. Consequently, these features often implement reactive algorithms inspired by control theory. Such methods will be referred to as *on-line* tuning methods, which can be considered as *advanced RRM*.

In contrast, parameter changes can also be implemented as part of network re-planning procedures carried out from the NMS. The NMS receives statistical performance data from the entire network, which is uploaded, at most, every half an hour. At the same time, new parameter settings can be downloaded remotely in the form of configuration files. This process, albeit automatic, can take several minutes. Hence, it is clear that NMS-based procedures are not able to cope with fast network changes. However, the use of long-term statistical data leads to more robust methods. Likewise, the availability of data from the whole network gives the methods a more global perspective. In addition, the absence of tight time constraints leaves the door opened to the use of complex optimisation methods, provided that a network model is available. More importantly, while the deployment of new features requires upgrading network equipment, re-configuring network parameters in the NMS can be performed with the existing equipment, which is crucial for a mature technology as GERAN. For these reasons, these methods are often

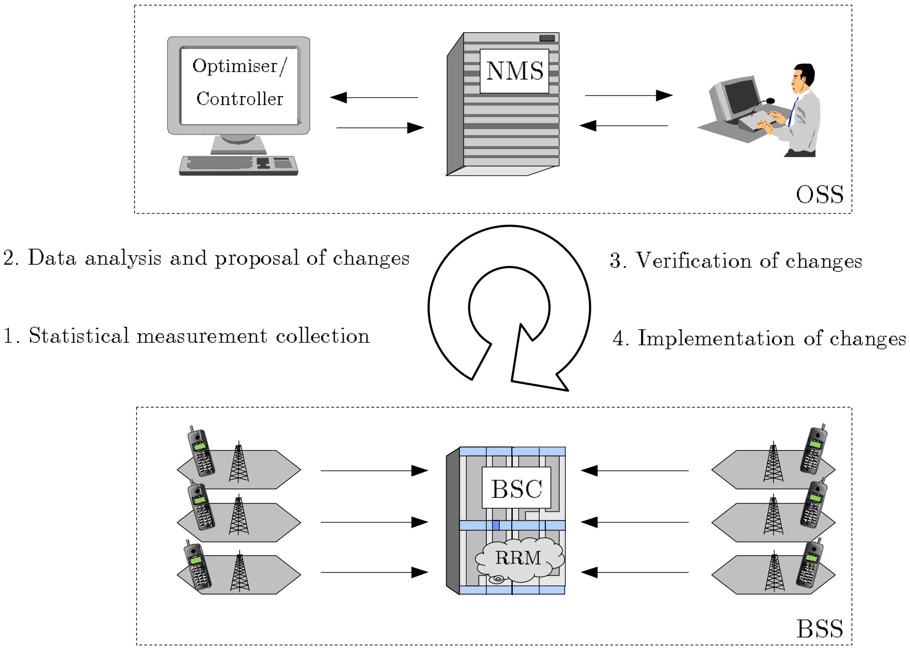

Optimiser/
Controller
NMS
OSS
2. Data analysis and proposal of changes
3. Verification of changes
1. Statistical measurement collection
4. Implementation of changes
BSC
RRM
BSS

Figure 1.4: The architectural structure of NMS-based optimisation.

preferred by network operators, and are thus considered in this thesis. The main drawback of the NMS-based parameter tuning is the inability to cope with fast network changes. Hence, these methods are conceived to cope with slow changes in user trends and scenario by tuning parameters slowly, and hence the name *off-line* (or *statistical*) tuning methods.

Figure 1.4 shows the conceptual structure of the NMS-based optimisation process. The process starts with the collection of network measurements that reflect the network state. For this purpose, the operator must first configure the BSS measurements to be gathered and the collection period. Once measurement data is available in the NMS, the optimisation (or control) algorithm analyses the data and suggests new parameter settings. These algorithms run as stand-alone programs, whose only interaction to the rest of the NMS is performed by accessing tables in the databases and generating parameter files that can be downloaded into the network. This fact minimises the coding effort and allows incremental modification of the algorithm based on field tests. The suggested changes are then checked by the operator and, finally, changes are implemented in the network.

1.4 Evaluation of Self-Tuning Algorithms

The analysis of algorithms aims to select the best among the ones available. There are many criteria that can be used to assess the value of an algorithm [14], amongst which are the simplicity, the solution quality, the workload, the storage requirements and the convergence speed. From the coding perspective, the *simplicity* of an algorithm is important, as it makes writing, debugging and modifying easier. From the performance point of view, the *solution quality* is the most determining factor, whenever the algorithm is not able to find the best solution. The memory and processing requirements are also important when dealing with large problems.

Thus, the *space complexity* of an algorithm is the amount of memory it needs to run, while the *time complexity* of an algorithm is the amount of computer time it needs to run to completion. Finally, when the algorithm is based on an iterative approach, the *convergence speed* defines how fast it approaches the final solution.

The aim of this work is the evaluation of self-tuning algorithms conceived for the NMS. In this work, the development effort is neglected, and, consequently, the analysis is focused on performance issues. As the current limitation in the NMS is the workload, and not the storage capacity, the performance evaluation is based on solution quality, time complexity and convergence speed. The rest of this section is devoted to describing assessment techniques for these criteria.

1.4.1 Solution Quality

In live networks, network size prevents operators from applying computationally-expensive algorithms during the tuning process. Otherwise, operators would be forced to either restrict the geographical area optimised or the periodicity with which the method is applied. Hence, it is common that *exact algorithms* cannot be applied to solve the problem. In this situation, *approximate* (or *heuristic*[2]) *algorithms* can be used to find good solutions efficiently.

Approximate algorithms entail the issue of estimating the quality of the solutions they find. Although some of these algorithms guarantee that the performance difference with the optimal solution is within certain limits, the quality of approximate solutions usually depends largely on the specific problem instance. To describe how good a particular solution is, it is enough to describe how close is to the optimal one. However, to assess the goodness of a method, the whole set of instances must be studied. A first approach to solve this problem is to consider three different scenarios: *best-case*, *worst-case* and *average-case*. For this purpose, it would be necessary to identify the particular problem instances that lead to these results. In this work, no thorough investigation is performed about the theoretical scenarios. Instead, the quality of all methods is tested over real test cases, as network operators are more concerned about actual rather than theoretical performance. This approach should perform reasonably well in all cases, provided that the selected test cases are significant. Sometimes, several test cases are available corresponding to different geographical areas of the same network. Under this assumption, it seems reasonable to compare methods by aggregating the value of the objective function in all test cases (which could be referred to as *aggregated-case*).

Although the previous approach may be sufficient for deterministic algorithms, it is not for random methods. In principle, operators are more interested in the average (rather than the best or worst) performance. Nonetheless, it is still interesting to know how these algorithms deviate from their average performance. For this purpose, confidence intervals are derived for the expected values of the most relevant indicators following a Monte-Carlo approach.

[2] Although some authors refer to approximate methods as those that guarantee solutions within a given performance bound, the terms *approximate* and *heuristic* will be interchangeably used in this work.

1.4.2 Time Complexity

The following paragraphs introduce basic concepts of the computational complexity theory, which is the branch of the theory of computation that studies the processing and memory requirement of algorithms. The main focus is on methods to evaluate the time complexity of algorithms. For clarity, the methods presented here are divided into analysis and measurement techniques.

Theoretical Time Complexity

The most intuitive measure of the time complexity of an algorithm is its execution time (or runtime). Unfortunately, runtime proves dependent on the particular computer, compiler, programming language and programming style. To circumvent this problem, the time complexity of an algorithm can be estimated by counting the number of operations. As it is often possible to isolate a particular operation that takes most of the computation time, time complexity can be approximated by the total number of these operations.

From the previous definition, it is clear that the complexity of an algorithm increases with the size of the problem instance. Thus, a measure of problem size must be defined first. In this work, most of the problems can be modelled by means of graphs. Consequently, the size of a problem is associated to the number of vertices or edges in the graph. Under this assumption, the time complexity analysis reduces to finding the expression that relates the number of operations to the number of vertices or edges in the graph.

Even with the previous approach, it is extremely difficult to make accurate estimations of the complexity of algorithms. Hence, several simplifications are often made in the analysis, which are described in the following paragraphs.

Asymptotic Time Complexity

Although the definition of a problem size measure simplifies the complexity analysis, the comparison between algorithms is still difficult, as the whole set of sizes must be evaluated to identify the best algorithm from the runtime perspective. Fortunately, to separate algorithms, it is normally enough to evaluate the growth rate (or, simply, the order) of complexity for large input sizes. Thus, complexity studies are only concerned with the *asymptotic* behaviour, investigating how the runtime increases with the input size in the limit.

For efficiency, the asymptotic behaviour of a function is normally expressed as bounds by means of the asymptotic notation [15]. Three symbols are commonly used: Ω for a lower bound, O for an upper bound, and Θ for a tight bound. The set $\Omega(f)$ are the functions that grow at least as fast as f, the set $\mathrm{O}(f)$ are the functions that grow no faster than f, and the set $\Theta(f)$ are the functions that grow at the same rate as f. More formally,

1) $f(n) = \mathrm{O}(g(n))$ means that there exist constants c and n_0, such that $f(n) \leq c \cdot g(n)$ for all $n \geq n_0$,

2) $f(n) = \Omega(g(n))$ means that there exist constants c and n_0, such that $c \cdot g(n) \leq f(n)$ for all $n \geq n_0$,

3) $f(n) = \Theta(g(n))$ means that there exist constants c_1, c_2 and n_0, such that $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$ for all $n \geq n_0$.

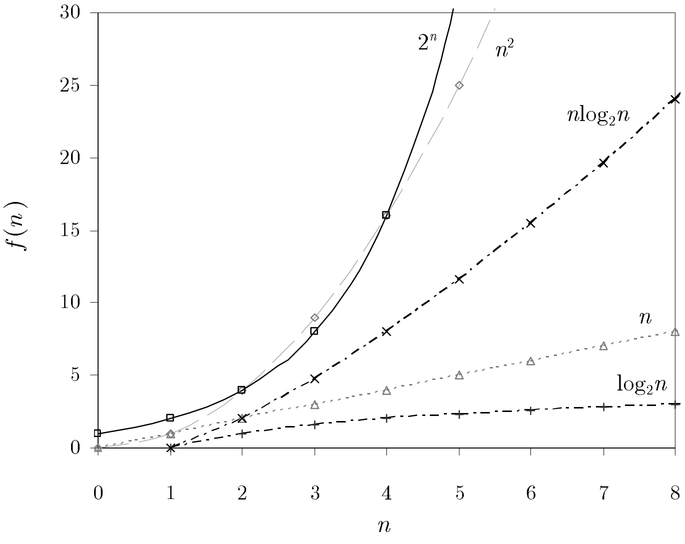


Figure 1.5: Plot of function values [14].

As n grows large, the influence of the low order terms in a function becomes negligible. Likewise, the coefficients become irrelevant when comparing functions of different order. For these reasons, the asymptotic notation only reflects the highest-order term of the function, neglecting coefficients and lower-order terms. For instance, $f(n) = 3n + 2 \in \mathrm{O}(n)$, $g(n) = n/4 + 100 \in \mathrm{O}(n)$ and $h(n) = n + logn \in \mathrm{O}(n)$. Figure 1.5 shows the values of the most common functions used to reflect time complexity. From the figure, it is clear that the utility of algorithms whose complexity is an exponential function or a polynomial of high degree of the size of the problem, n, is restricted to small values of n.

At this point, it should be pointed out that, even though the asymptotic efficiency is a valuable indicator to compare algorithms, it must be handled with care. An algorithm that is asymptotically the most efficient is not necessarily the best choice for all input sizes. Although the runtime is dominated by higher-order terms for large inputs, this is not necessarily the case for small inputs, where coefficients and lower-order terms might have an influence. This fact is worth remarking since some algorithms in this work deal with inputs that are not extremely large, but of medium size. Under these conditions, the asymptotic behaviour proves a useful performance indicator, but the final comparison among algorithms must still be performed over the actual runtimes.

Best-case, Worst-case and Average-case

The runtime of an algorithm may depend on the specific problem instance, and not only on its size. To circumvent this problem, three situations are clearly isolated: best-case, worst-case and average-case. The *best-case* runtime is the minimum runtime over all the possible inputs of a given input size. Similarly, the *worst-case* and *average-case* runtime are the ones obtained with the worst and average inputs of a given size. The definition of a worst-case scenario is easier in theory, as it is often possible to find the longest path through the algorithm without determining the exact input that could generate this. In practice, it is easier to estimate the average-case performance by the average runtime across multiple runs of the algorithm. For these reasons, the former is the most widely used in algorithm analysis, while the latter is often used in algorithm testing.

Problem Complexity

In the early design stages, it is important to identify that a problem cannot be solved exactly in a reasonable time. Thus, exact methods can be discarded and the development work can concentrate on heuristic methods. Complexity theory helps designers to identify these intractable problems.

The computational complexity theory classifies problems, according to their difficulty, into two classes: $\mathcal{P}$ and $\mathcal{NP}$ [16]. The complexity class $\mathcal{P}$ (*Polynomial*) consists of those problems that are solvable by deterministic algorithms in polynomial time, i.e., their time complexity is $\mathrm{O}(n^p)$ for some constant p, where n is the size of the input to the problem. The class $\mathcal{NP}$ (*Non-deterministic Polynomial*) consists of those problems that can only be solved in polynomial time by a non-deterministic algorithm. A *non-deterministic algorithm* is a theoretical tool to define an algorithm that has operations whose outcomes are not uniquely defined. These operations can be interpreted as choice points where different continuations are possible, without any specification of which one will be taken. By definition, the time required by a non-deterministic algorithm on a given input is the minimum number of steps needed to reach a successful completion. A non-deterministic algorithm can be viewed as multiple deterministic algorithms running in parallel, one for each of the possible choices, where the first one to finish successfully terminates all others. For obvious reasons, the class $\mathcal{P}$ is a subset of $\mathcal{NP}$, since any problem in $\mathcal{P}$ can be solved in polynomial time without the need of a non-deterministic algorithm.

Among all problems in $\mathcal{NP}$, the class $\mathcal{NP}$-*complete* represents the toughest problems. Any $\mathcal{NP}$-complete problem can be transformed, in polynomial time, into an instance of any other problem in the same class. Therefore, if there were a polynomial algorithm for an $\mathcal{NP}$-complete problem, it could be used to solve all other $\mathcal{NP}$-complete problems in polynomial time. Since such an algorithm has not been found yet, these problems are the ones most likely not to be in $\mathcal{P}$. Examples of $\mathcal{NP}$-complete problems are the traveling salesman problem, the knapsack problem, the vertex cover problem, the graph covering problem and the subset sum problem.

The previous definition is used to find evidence that there is no efficient algorithm to solve exactly a certain problem. By reducing a problem to an $\mathcal{NP}$-complete problem, it is proved that finding a good solution for it is as difficult as doing it for problems that have been very heavily studied by many experts. In this work, several problems will be identified as $\mathcal{NP}$-complete.

Time Complexity in Practice

The measurement of time complexity aims to evaluate the execution time of a particular program. As stated previously, results are dependent on many different factors, which make comparison between programs difficult. To eliminate the dependency on the computer, compiler and programming language, programs in this thesis have been developed from scratch whenever possible. Unless stated otherwise, the source code is written in the Matlab© 6.5 environment and later compiled with the supplied Matlab-to-C/C++ compiler [17]. Finally, the compiled programs are executed in a Windows-based 2.4GHz Pentium III computer with 1GB of RAM.

To measure runtime, a function is needed to check the system clock just before and after the program is executed. In Matlab, this is accomplished by inserting the *tic* and *toc* commands [18] into the code to start and stop the clock, respectively. The accuracy of these measurements largely depends on the function that gets the value of the system clock. On Windows platforms, this function is the *GetLocalTime* function, which returns a value in milliseconds. Therefore, the accuracy of the previous Matlab functions on Windows are limited to milliseconds. With this resolution, to time a short event, it may be necessary to repeat it several times and divide the total time by the number of repetitions.

To generate suitable test data, it is necessary to select whether worst-case or average-case performance is evaluated in practice. To be coherent with the approach used to assess solution quality, runtime measurements performed in this work are concerned with average performance. Thus, the runtime of an algorithm is calculated by averaging all attempts performed over the same problem instance and aggregating the entire set of instances.

1.4.3 Convergence Speed

Some optimisation algorithms and, basically, all discrete-time control algorithms are based on an iterative approach. An *iterative algorithm* aims to solve a problem by successive approximations to the final solution starting from an initial solution. In optimisation, iterative algorithms are used in problems involving a large number of variables, where direct methods would be prohibitively computationally expensive. Likewise, discrete-time control algorithms change system parameters on each iteration based on the value of system outputs on the previous iteration, and are thus iterative in nature.

In iterative algorithms, the sequence of solutions $x^{(1)}, x^{(2)}, ..., x^{(\infty)}$, that are visited is referred to as *search trajectory*. For obvious reasons, it is important to ensure that this trajectory converges to a fixed equilibrium solution, regardless of the initial situation and the presence of disturbances. An algorithm is said to be *globally convergent* if it converges for any arbitrary initial solution, while it is *locally convergent* if it only converges when the initial solution is sufficiently close to the final solution. Likewise, an algorithm is said to be *stable* if small changes in the initial solution and numerical errors in the computation do not have a significant effect. The same concept is applied to control systems, where an equilibrium state is said to be *asymptotically stable* if all nearby states converge to it.

Once convergence and stability are ensured, the main concern is how fast the trajectory approaches to the equilibrium solution. A faster convergence reduces the number of steps needed to reach the final solution, which is translated into several benefits. When the algorithm is applied to a network model, a faster convergence reduces the computational load. Likewise, when the algorithm is applied to the real network, a faster convergence brings forward the performance benefit of the optimisation process.

The speed of convergence can be quantitatively described by two parameters: the *convergence order* and the *convergence rate*. Formally, a sequence converges to ξ with order q if

$$\lim_{n\to\infty} \frac{\left|x^{(n+1)} - \xi\right|}{\left|x^{(n)} - \xi\right|^q} = \eta \qquad \text{with} \quad \eta, q > 0 \, , \tag{1.1}$$

where q is the convergence order and η is the convergence rate. If $q = 1$, convergence is called *linear*, while convergence with $q = 2$ is called *quadratic*. As the previous definition deals with the asymptotical behaviour, it does not give any information about the first states of the sequence. For clarity, the term *convergence speed* will be used to refer broadly to the speed in the entire sequence, whereas the term *convergence rate* will only be applied to refer to the speed in the limit.

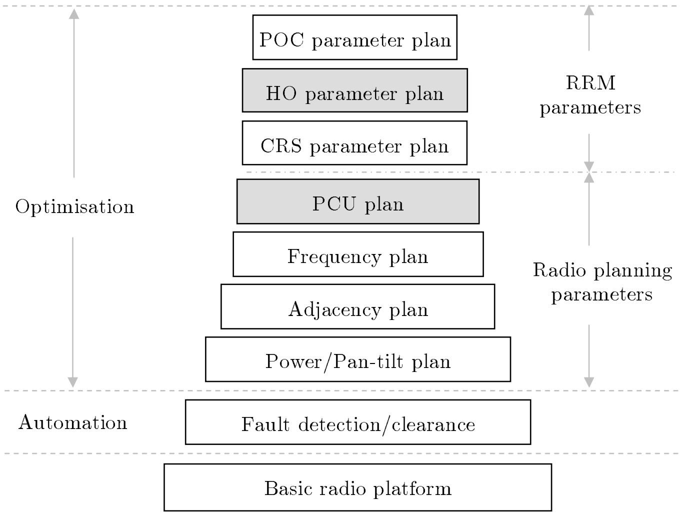


Figure 1.6: Topics on this thesis.

Most iterative algorithms used for system optimisation or control are based on an incremental approach. In such an approach, the current solution is replaced by a neighbouring one, which only differs slightly from the previous one. In continuous systems, such an operation can be expressed by the recurrence formula

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \beta^{(n)} \cdot \mathbf{g}^{(n)} , \tag{1.2}$$

where $\mathbf{x}^{(n)}$ is a vector with the values of the decision variables (i.e., parameters) in iteration n, $\mathbf{g}^{(n)}$ is a vector indicating the search direction and $\beta^{(n)}$ is an iterative parameter that indicates the step-length in the direction of $\mathbf{g}^{(n)}$ (provided that $\left\|\mathbf{g}^{(n)}\right\| = 1$). From the previous definition, it is clear that β has a strong influence on the stability and convergence properties of the algorithm. A larger β normally leads to a faster convergence at the expense of a reduced stability. Simple algorithms use a fixed value of β, while more advanced algorithms modify β across iterations. In optimisation algorithms, $\beta^{(n)}$ can be obtained by estimating the value that minimises the objective function in the direction of $\mathbf{g}^{(n)}$. In feedback control algorithms, $\beta^{(n)}$ is the gain of the feedback loop, which can be used to change the sensitivity of the controller (referred to as *gain-scheduling* [19]).

1.5 Conclusions

This thesis deals with the automatic optimisation of several parameters in GERAN. Figure 1.6 shows the parameters under optimisation highlighted in grey. Firstly, this thesis considers the optimisation of the assignment of cells to PCUs in the BSCs, which is defined in the PCU plan. Secondly, this work considers the tuning of HO parameters. While the former process deals with radio planning parameters, which should be changed occasionally, the latter deals with RRM parameters, which can be modified frequently.

The methods proposed in this thesis are thought to be run in the NMS as part of network re-planning procedures. Therefore, they are conceived to cope with long-term changes in the network by applying a mixture of optimisation and control methods based on statistical information. For this reason, these methods are considered as static policies, which can be applied on a regular basis.

Chapter 2 is devoted to the optimisation of the PCU plan to increase the efficiency of the CRS process. To solve the problem, an optimisation method is used, since an accurate model can be derived from network measurements. As this problem can be reduced to a problem that is known to be $\mathcal{NP}$-complete, exact methods are computationally expensive. Thus, the bulk of the study concentrates on heuristic methods. Nonetheless, exact methods are still considered, since the problem instances are not extremely large.

Chapter 3 is devoted to the tuning of HO parameters in GERAN to solve localised congestion problems. As accurate propagation data is rarely available, no accurate model exists. Thus, the tuning algorithm must interact directly with the real network. To reduce the risk of causing new problems, parameters are tuned by an intuitive control algorithm that aims to balance the traffic between adjacent cells. The complexity of these algorithms proves to be linear to the size of the problem, which justifies that complexity is not an issue on these algorithms. The price to be paid is the lack of an optimality proof and a greater sensitivity to convergence and stability issues, which will be dealt with by a simple gain-scheduling algorithm.

Chapter 2

Optimisation of the Assignment of Packet Control Units in GERAN

This chapter deals with the optimisation of the assignment of cells to PCUs in GERAN. After a brief description of the problem, the problem is formulated analytically by means of graph theory. Based on this formulation, both an exact and a heuristic method are proposed to solve the problem. Field trial results are then presented to show the performance enhancement a simple algorithm can produce in a live network. Finally, a comprehensive analysis of more refined methods is performed over an extensive collection of graphs constructed from data of a live GERAN.

2.1 Introduction

The steady traffic growth in cellular networks calls for continuous addition of network elements. To allow system scalability, these network elements are hierarchically structured. Thus, it is not unusual that the addition of a new element forces a re-configuration of network hierarchy. In this situation, efficient methods are needed to find the best clustering of elements to be assigned to elements of a higher level in the network hierarchy.

In parallel, GERAN operators have been actively launching a number of new packet-data based services over the past years. A few such examples are *Web Browsing*, *Multimedia Messaging*, *Streaming*, *Push-to-Talk over Cellular* and *On-line Gaming* [10]. As a consequence, network optimisation related to these services has come into focus. The aim of these optimisation procedures is to maximise network capacity with existing resources, since the user experience of these services is strongly linked to the data rates offered. In GERAN, these services are based on GPRS or EDGE. Although theoretical peak data rates per Time Slot (TSL) of 20kbps for GPRS and 48kbps for EDGE might be considered acceptable, the actual data rate is often below these values for several reasons [10]. First, the tight frequency reuse limits the carrier-to-interference ratio and thus leads to the selection of lower coding schemes than would otherwise be possible. Inaccurate dimensioning of cell capacity also leads to reductions in the data rate as more users have to share the same TSL than originally planned. Finally, every change of serving cell made by the user (known as *cell re-selection*) causes interruption of the associated data flow, which leads to a service outage period, the duration of which depends on mobility-management procedures and network features in use.

In this framework, the assignment of cells to PCUs within a BSC is key to maximising user data throughput in GERAN. Each BSC contains a certain number of PCUs, which are responsible for the control of packet data traffic. Cells connected to the BSC where packet-data services must be offered have to be associated with one of these PCUs. Field trial results have shown that inter-PCU cell re-selection causes far longer service breaks than intra-PCU cell re-selection [20]. Hence, the PCU plan should minimise the number of inter-PCU re-allocations experienced by users when a change of serving cell takes place. At the same time, the PCU plan should also ensure that the load of all PCUs remains within the limits stated by the vendor.

The *Cell-to-PCU assignment problem* (CPAP) can be formulated as a *graph partitioning problem* (GPP) [21]. Since most formulations of the latter problem are known to be $\mathcal{NP}$-complete [16], several heuristics have been proposed to find near-optimal solutions to the problem efficiently. These heuristic methods produce a fixed or arbitrary number of clusters of bounded size that minimises inter-cluster handovers. However, most of the previous approaches have been conceived for stationary networks, where the problem is focused on the initial design of the network. Therefore, little attention has been paid to the re-planning and maintenance procedures, which are carried during the operational stage. Consequently, performance aspects such as the time complexity, the number of network changes, the time validity and the ease of maintenance of the new solutions have traditionally been neglected. Only recently have some studies been published related to adaptive re-partitioning, where a graph is partitioned based on an existing solution, and with the secondary objective that the number of changes is minimised [22][23][24]. Nonetheless, most network management issues still remain unexplored.

Although the previous methods usually provide an acceptable solution to the partitioning problem, the quality of these solutions remains unknown, since it is normally not possible to find the optimum solution for graphs of practical size in reasonable time. Thus, the analysis is traditionally limited to the comparison between heuristics. However, unlike other applications, the size of graphs handled in the CPAP is relatively small. Therefore, exact methods become a viable option, provided that runtime constraints are sufficiently relaxed.

In this work, the CPAP is analysed from the perspective of re-planning procedures in GERAN. The problem is formulated for the first time by means of graph theory based on network statistics. The analytical formulation presented later makes use of the classical integer-linear programming model of the graph partitioning problem. This formulation can be solved by exact methods included in most commercial optimization packages. The solution thus obtained can be used as a benchmark for approximate methods. As an alternative, a novel heuristic algorithm is proposed. The proposed algorithm combines two classical graph partitioning techniques that have been considered separately in the past: the multi-level refinement [25][26][27] and the clustered adaptive multi-start [28]. Likewise, the new strategy includes algorithms to ensure the connectivity of cells assigned to the same PCU, thus improving the geographical consistency of solutions.

The assessment of the proposed methods is based on a two-folded approach. Firstly, a field trial shows the performance enhancement that a simple optimisation algorithm produces on a limited network area. Subsequently, a comprehensive analysis of more refined methods is performed over a large set of test cases constructed from data of a live network.

The rest of the chapter is organised as follows. Section 2.2 provides a brief description of the CPAP in GERAN. Section 2.3 describes two alternative methods to solve the problem, either exactly or approximately. Section 2.4 presents the results of a simple heuristic algorithm in a field trial. Section 2.5 presents a comprehensive analysis of more refined methods over models constructed from real data. Finally, the main conclusions are discussed in Section 2.6.

2.2 Problem Formulation

This initial section begins with the analysis of the impact of the CRS process on the performance of data transmission in GERAN. The assignment of cells to PCUs is modelled as a graph partitioning problem, describing the network data handled in the PCU planning process and the main differences with other classical formulations of the graph partitioning problem. Three different analytical formulations of the problem are presented, as a result of different actions to improve the efficiency of exact methods. Finally, the current state of research and technology is discussed. All the issues presented more than justify the need for the method proposed in the next section.

2.2.1 The Cell Re-Selection Process in GERAN

In any mobile communication system, it is crucial that an MS is always served by the cell that offers the best coverage. During connections, the HO mechanism implemented on the network side takes charge of this responsability. However, when an MS is in idle mode, a different mechanism must control in which cell the MS will initiate future connections. This mechanism is commonly referred to as the *cell (re)selection* (CRS) process.

In GSM, CRS decisions are taken by the MS based on field-strength measurements and parameters broadcasted by the BTS. Two alternative criteria can be used to rank the different candidate cells [29]. The C1 criterion (also called path loss criterion) compares the received signal level from the different BTSs against some threshold that is broadcasted by each BTS. Among the cells with positive values of C1, the MS chooses the one with the highest C1 value. Alternatively, the C2 criterion provides an extension of C1 by including an offset to bias the CRS decision in favour of some cells.

In GPRS, an idle MS follows the same algorithm as in GSM to select the best cell. However, the use of a differentiated parameter set leads to two new criteria: C31 and C32 [29]. Once per second, idle MSs update the values of C1, C2, C31 and C32 criteria for the serving and non-serving cells. A CRS is then performed if the C1 value for the serving cell falls below zero or a non-serving cell is better than the serving cell based on C32. The target cell is the one with the highest C32 among those with C31 ≥ 0.

The previous algorithm is applied by MSs in idle mode. However, when an MS is transferring packet data, it is neither strictly in idle mode, nor has it a permanent connection to the serving cell where the HO mechanism would work. In this situation, the CRS is responsible to select to/from which cell user data is sent. Depending on network configuration, there exist different CRS mechanisms for an MS in packet-transfer mode. In [29], three different network control CRS modes are described: NC0, NC1 and NC2. In NC0, which is the default mode, the MS performs the CRS autonomously like in idle mode. In NC1, the MS also performs the CRS autonomously, but, in addition, it sends measurement reports to the network periodically. In NC2, the MS sends periodical measurement reports and the network has the control of the CRS process, except when the CRS is triggered by a DL signalling failure or a random access failure.

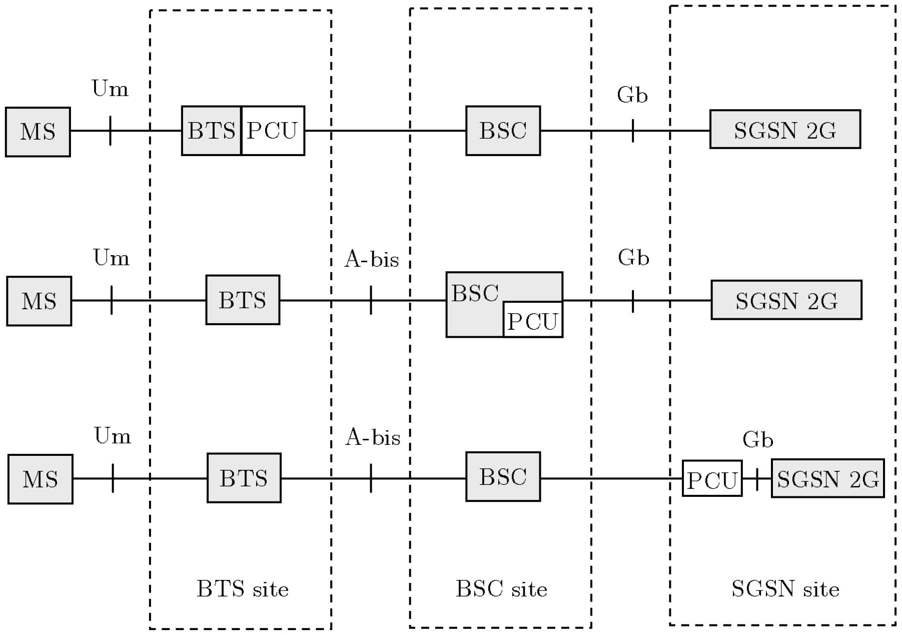


Figure 2.1: The packet control unit in GERAN A/Gb mode.

The CRS process has a strong influence on the performance of packet-data services in GERAN. Existing GPRS networks, which are based on GERAN A/Gb mode [30], lack the support of real-time packet-data services. The main challenge in providing real-time services over these networks is the delay and packet loss incurred during a cell change. Trial results over live networks have shown that interruption associated to cell change ranges from 2 to 12 seconds [4][31]. Although such an interruption is acceptable for background and streaming services, it is not for conversational services, for which a maximum delay of 0.15 seconds is allowed [32] [33]. For future GERAN releases, several features have been proposed to reduce this delay, the foremost of which are Packet Broadcast Channel [10], Network-Assisted Cell Change [10] and Packet-data Handover [34]. Unfortunately, none of them are available in current networks and legacy mobiles make these features difficult to implement for operators. Hence, current GPRS networks must be optimised for the existing BSS capabilities (i.e., CRS).

In GPRS, the delay associated to a cell change can be minimised by proper planning of the network hierarchy. In particular, the planning of PCUs has a significant impact on the performance of packet-data services. The PCU is a physical unit that is responsible for the control of packet data. Its main functions are packet-data radio resource management, connection establishment and management, data transfer and uplink power control.

Figure 2.1 presents three different options to integrate the PCU into the logical architecture of GERAN A/Gb mode. In the figure, it is observed that, although the PCU is logically associated with the BSC, the PCU could lie in any physical element between the BTS and the SGSN. Nonetheless, most manufacturers implement PCUs as a physical unit of the BSC (i.e., intermediate option).

Figure 2.2 illustrates how the PCU is currently used within the BSC. In the figure, it is observed that each BSC contains a certain number of PCUs to which the cells of the BSC have to be associated. The number of PCUs in a BSC is defined by the operator during the

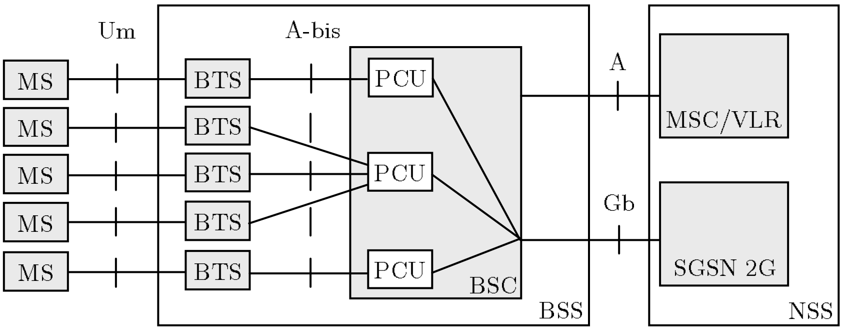


Figure 2.2: The packet control unit in the Base Station Controller.

network dimensioning stage based on demand of packet-data services. For that purpose, the constraints of the PCU element are taken into account. The most common PCU constraints are the maximum packet-data traffic handled, the maximum number of users with a temporary flow identity, as well as the maximum number of cells, transceivers and TSL devoted to data traffic in cells served by the same unit. Once the network is dimensioned, the operator must configure the association between cells and PCUs, which is known as the *PCU plan.* The problem of assigning cells to PCUs during the construction of a PCU plan will be referred to as the *Cell-to-PCU Assignment Problem* (CPAP). The aim of the PCU plan is two-fold: connect cells that are close in propagation terms to the same PCU, while keeping the PCU load within given limits.

The former objective aims to reduce the number of PCU re-selections suffered by users when a change of serving cell takes place, since CRS between cells of the same PCU (i.e., intra-PCU CRS) causes far shorter service breaks than CRS between cells on different PCUs (i.e., inter-PCU CRS). During intra-PCU CRS, the untransferred data is not deleted immediately from the buffer of the old cell, but forwarded to the buffer of the new cell. This fact improves the end-user experience, since data transfer may continue immediately after the MS has completed the cell update procedure. By contrast, in inter-PCU CRS, the untransferred data is not moved to the new cell, but is deleted. In this case, higher protocol layers ensure that the deleted data is re-transmitted by the SGSN. Therefore, in order to maximise data throughput for mobile users of packet-based services, it is crucial to minimise the number of inter-PCU CRSs. In other words, cells between which many CRSs occur should be on the same PCU, whilst cells weakly related can be placed on different PCUs.

Ideally, the optimisation process should be based on CRS statistics of users in packet-switched (PS) mode. Unfortunately, such information is not currently available in the NMS, as the CRS process is controlled by the MS. In the absence of such PS-based statistics, it is deemed most suitable to utilise HO statistics related to circuit-switched (CS) services. Assuming that the mobility of mobiles in PS mode is similar to that of mobiles in CS mode, the errors in such an analysis should be relatively small. Hence, the main objective is re-formulated as the minimisation of the number of HOs between cells of different PCUs.

As a secondary objective, the optimisation process should reject those solutions that lead to a large traffic imbalance between PCUs. The robustness of solutions against an increase of traffic demand (whether temporary or permanent) is thus improved and the validity of the solutions is prolonged. Nonetheless, putting too much emphasis on perfect balance is not recommended, since this strategy would possibly lead to solutions with worse performance. As a consequence, the balance should not be included in the objective function to be minimised, but only considered as a constraint. This imbalance constraint is dealt with by limiting the maximum load ratio among PCUs of the same BSC. Thus, a trade-off between robustness and performance is achieved in the solution finally implemented in the network.

Several other performance criteria must be taken into account to ease the adoption of these methods by operators as part of their daily routine. In a live environment, the time to compute a new solution for the problem is a key performance aspect. Ideally, every time a new cell or PCU is added to the network, the CPAP must be solved in the BSC to which it is associated. Since this event is rather frequent, the execution time is one of the most relevant criteria to assess the value of an algorithm. The application of the method should also minimise the number of changes from the initial solution. Since the re-configuration of the existing PCU plan might require disabling the packet-data transmission on affected cells, the download time must be kept to a minimum. As new cell-to-PCU assignments can only be downloaded sequentially, the smaller the number of cells that suffer a PCU re-allocation, the higher the cell availability. Finally, operators prefer solutions in which cells in the same PCU are geographically close to each other. Although this property is not strictly necessary for network operation, this sort of solution is easier to check visually by maintenance personnel.

2.2.2 The PCU Assignment as a Graph Partitioning Problem

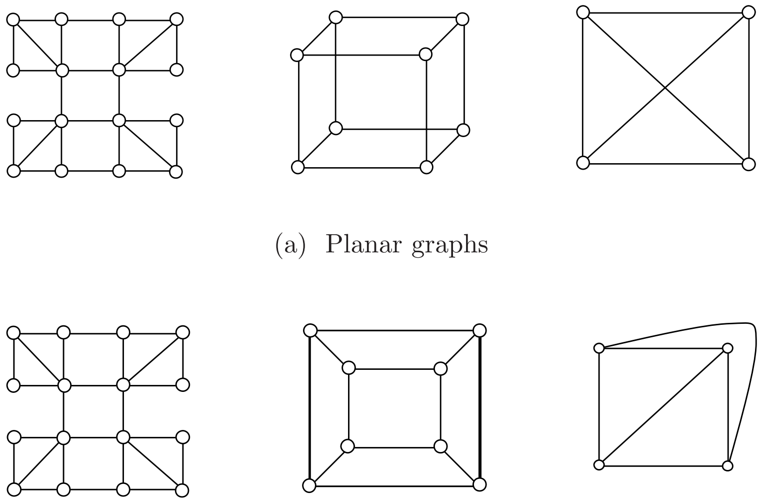
An analytical formulation of the graph partitioning problem is now presented. For clarity, the following paragraphs introduce basic notation and definitions from graph theory, which are used throughout the document. The subsequent formulation will form the basis of exact solution methods discussed in the next section.

Notation and Definitions from Graph Theory

A *graph* G=(V,E) is a pair of sets V and E, where V is a set of elements referred to as *vertices* and E is a set of pairs of vertices referred to as *edges* [35]. The two vertices i, j of an edge (i, j) are its *endvertices*, which are defined as *adjacent* (or *neighbours*) to each other. An edge is *incident* to a vertex if the vertex is at one end of the edge, while a vertex is incident to an edge if that edge is incident to the vertex. The set of neighbour vertices of a vertex i will be denoted as $V(i)$, while the set of edges incident to a vertex will be denoted as $E(i)$. The *degree* of a vertex i, $|E(i)|$, is the number of edges for which i is an endvertex[1]. A vertex i is called *isolated* if it does not have any adjacent vertex (i.e., $|E(i)| = 0$).

Graphs can be classified in terms of their attributes. A *simple graph* is one where every edge is defined between different vertices (i.e., both endvertices of an edge cannot be the same vertex). In a *directed graph*, an origin and a source vertex are defined for every edge, while in a *non-directed graph*, no distinction between endvertices is made. In the latter graphs, at most one edge is defined between two vertices. A graph is called *complete* if there exists an edge for every pair or vertices. In a *weighted* graph, vertices and edges are associated a certain weight. Hereafter, ω_i denotes the weight of vertex i and γ_{ij} denotes the weight of edge (i, j).

[1] In this work, $|E|$ denotes the number of elements in E (i.e. the cardinality of E), while $\|E\|$ denotes the sum of the weights of elements in E.



(a) Planar graphs

(b) Plane graphs

Figure 2.3: Several representations of graphs.

A graph can be characterised by several parameters. The *order* of a graph is its number of vertices, $|V|$, and the *size* of a graph is its number of edges, $|E|$. From these parameters, the *density* of a graph, Δ, is defined as the ratio between its number of edges and the number of edges of a complete graph with the same number of vertices. Concretely, the density of a directed graph is $|E|/|V|^2$, whose value ranges from 0 to 1. In this work, a distinction is made between sparse and low-density graphs [36]. A graph is said to be *sparse* if $|E|$ is $O(|V|)$, so the density decreases with increasing $|V|$. Meanwhile, a graph is said to be *low density* if $|E|$ is $O(|V|^2)$, so the density remains constant with increasing $|V|$.

Several ways exist to represent a graph. The usual way to depict a graph on a piece of paper is by drawing a dot for each vertex and joining any pair of dots by a line if the corresponding two vertices form an edge. This simple representation is clear as long as no intersection exists between the lines. This condition is only fulfilled in planar graphs. A *planar graph* is a graph that can be embedded in the plane so that no edges intersect. A planar graph already drawn in the plane is called a *plane graph.* To clarify the difference between the two definitions, Figure 2.3 (a) illustrates three planar graphs and Figure 2.3 (b) shows their corresponding plane graphs. From left to right, Figure 2.3 (a) represents a mesh of a planar structure, the cube graph and the complete graph with four vertices. Although graphs in Figure 2.3 (a) display intersections between edges, they are still planar graphs, since it is possible to re-draw them to avoid intersections, as shown in Figure 2.3 (b).

To handle graphs in computer programs, the adjacency matrix is the most common representation. The adjacency matrix $\mathbf{AM} = (a_{ij})_{|V|\text{x}|V|}$ of G is defined by

$$a_{ij} = \begin{cases} \gamma_{ij} & \text{if } (i,j) \in E, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

In a simple graph, $diag(\mathbf{AM}) = 0$. If the graph is also non-directed, AM is symmetrical and $a_{ij} = a_{ji}$. In this case, the adjacency matrix can be represented by its upper triangular matrix, reducing its size from $|V|^2$ to $\frac{|V|(|V|-1)}{2}$. The density of the graph is thus re-defined as $|E|/\frac{|V|(|V|-1)}{2}$. For sparse and low-density graphs, a more compact representation can be obtained by using *adjacency lists*. This structure is a collection of lists, one per vertex, which gives the vertices to which each vertex is adjacent. The size of this structure is therefore $\mathrm{O}(|E|)$.

Inside a graph, a *path* is an ordered series of adjacent vertices $V = \left\{v^{(0)}, v^{(1)}, ..., v^{(n)}\right\}$, linked by a series of adjacent edges $E = \left\{(v^{(0)}, v^{(1)}), (v^{(1)}, v^{(2)}), ..., (v^{(n-1)}, v^{(n)})\right\}$. The number of edges of a path is its *length*. The *distance* between two vertices i and j, $d(i,j)$, is the minimum length of the paths between them. If no path exists between two vertices, it is assumed that $d(i,j) = \infty$. A graph is said to be *connected* if there exists a path between any pair of vertices.

This work deals with the problem of partitioning a graph. A *partition* of G, $\Pi(G)$, is a subdivision of V into disjoint subsets of vertices $V_1, V_2, ..., V_k$, referred to as *subdomains*. Each subdomain V_i defines a *subgraph* $G' = (V_i, E(V_i))$ of G, such that $V_i \subseteq V$ and $E(V_i) \subseteq E$. A *k-way partition* is a partition that consists of k (non-empty) subdomains. In this work, a partition will be denoted by an n-dimensional vector $\mathbf{\Pi} = \{\pi_1, \pi_2, ..., \pi_n\}$ $(n = 1 : |V|)$, where π_i is the index of the subdomain to which vertex i is assigned (i.e., $\pi_i \in \{1, 2, \cdots, k\}$). The sum of weights of the vertices in subdomain V_i is defined as the *subdomain weight*, $\|V_i\|$. If the graph is unweighted, the *subdomain weight* is the number of elements in the subdomain, $|V_i|$. The ratio of the most heavily loaded and the most lightly loaded subdomains in a partition is referred to as the *weight imbalance ratio*. Every partition defines a set of edges that join vertices in different subdomains, $\delta(V_1, ..., V_k)$, referred to as a *multi-cut*. The sum of weights of the edges in the cut, $\|\delta(V_1, ..., V_k)\|$, is the *edge-cut*. For an unweighted graph, the *edge-cut* is the number of edges in the multi-cut, $|\delta(V_1, ..., V_k)|$. To refine a partition, partitioning algorithms need to identify vertices in the limit of a subdomain. The *neighbourhood* of a subdomain V_i, $V(V_i)$, comprises all vertices in the graph adjacent to, but excluding, vertices in V_i. The concept of distance can also be defined for partitions as a measure of the difference between partitions. Thus, the *distance* between two partitions of the same graph is defined as the number of vertices that are not assigned to the same subdomain in both solutions.

Model of the Problem by means of Graph Theory

The procedure of assigning cells to PCUs aims to minimise the number of HOs between cells in different PCUs, while keeping the load of the different PCUs within certain limits. This assignment problem can be formulated as a combinatorial optimisation problem known as a *graph partitioning problem* (GPP) [37]. In this approach, the network area under optimisation (i.e., a BSC) is modelled by a simple non-directed weighted graph, as shown in Figure 2.4. The vertices of the graph represent the cells within the BSC, while the undirected edges between them represent the adjacencies defined by the operator for HO purposes. Since edges are non-directed, it is assumed that adjacencies are bi-directional in nature (i.e., the adjacency between two cells is unique, regardless of the direction of the user movement). The weight associated to each vertex might represent its contribution to the load of the PCU in terms of cells, transceivers, GPRS TSLs or packet-data traffic. However, preliminary analysis of field data proved that the number of GPRS TSLs per PCU is currently the most restrictive constraint [31]. Thus, the weight of each vertex, ω_i, denotes the number of TSLs devoted to GPRS traffic (i.e., GPRS territory). The weight of each edge, γ_{ij}, denotes the number of HOs between the cells on its ends, which can be derived from HOs in both directions of the adjacency. The partitioning of the graph, performed by grouping vertices into a fixed number of subdomains, k, reflects the

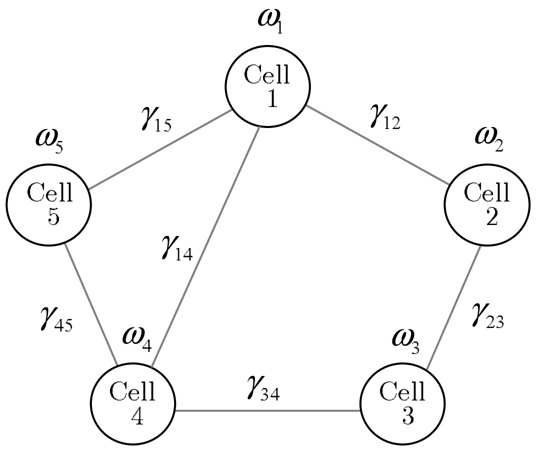


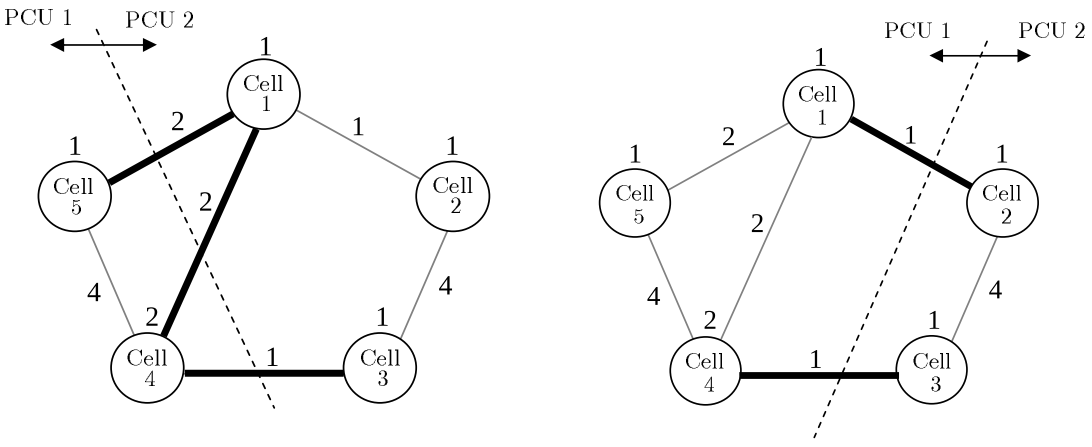
Figure 2.4: Model of the cell-to-PCU assignment problem by means of a graph.

assignment of cells to PCUs. Thus, the sum of weights of the edges that join vertices in different subdomains represents the number of users that change PCU after a cell change.

Over this graph, the CPAP aims to find a partition of the graph such that the edge-cut is minimised and the weight of each subdomain is within certain limits. For clarity, Figure 2.5 presents an example of CPAP over an instance of the graph depicted in Figure 2.4. In the example, it is intended to distribute 5 cells over 2 PCUs. The underlying problem aims to find a partition that subdivides the graph into two parts (i.e., k=2). Such a partition is called a *bisection*. Figure 2.5 (a)-(b) depict two bisections of the same graph. To ease the interpretation of results, those edges that contribute to the edge-cut of each bisection are highlighted in bold. In the bisection of Figure 2.5 (a), the aim is to minimise the edge-cut, but still enforcing the balance between subdomain weights. For that purpose, the two clusters of vertices (i.e., $\{1, 2, 3\}$ and $\{4, 5\}$) were selected so that the frontier between subdomains did not include the edges of a largest weight (i.e., (4,5) and (2,3)). This bisection results in an edge-cut of 5 and a perfect equilibrium between subdomains, since both subdomains have a weight of 3. In contrast, the bisection of Figure 2.5 (b) aims to minimise the edge-cut at the expense of an increase of the imbalance between subdomains. The cluster of vertices were defined so as to set the frontier over the edges of lightest weight (i.e., (1,2) and (3,4)). As a result, cell 1 moves from PCU 2 to PCU 1 and the edge-cut is thus reduced from 5 to 2. In contrast, the weight of PCU 2 is now twice that of PCU 1. The bisection of Figure 2.5 (b) proves that it is possible to minimise the edge-cut, provided that a certain load imbalance between subdomains is permitted. Unlike in other applications, where perfect balance is targeted, the strategy in this work will partly sacrifice the load equilibrium for the sake of minimising edge-cut.

2.2.3 Mathematical Formulation

The GPP behind the CPAP can be formulated as a *bounded, min-k cut problem* [38], which is described as follows. Let $G = (V, E)$ be an undirected weighted graph, consisting of a set of vertices V and edges E, vertex weights ω_i and edge weights γ_{ij}. Let B_{aw}, B_{rw} be real numbers defined as absolute and relative weight bounds, such that $0 < B_{aw} \leq \sum\limits_{i \in V} \omega_i$, $1 < B_{rw} < \infty$. The problem stands for the partition of V into k subdomains, V_1, V_2,..., V_k, such that



$$\|V_1\| = \sum_{i \in V_1} \omega_i = 2+1 = 3$$

$$\|V_2\| = \sum_{i \in V_2} \omega_i = 1+1+1 = 3$$

$$\|\delta(V_1, V_2)\| = \sum_{(i,j) \in \delta(V_1,V_2)} \gamma_{ij} = 2+2+1 = 5$$

(a) Maximum equilibrium

$$\|V_1\| = \sum_{i \in V_1} \omega_i = 2+1+1 = 4$$

$$\|V_2\| = \sum_{i \in V_2} \omega_i = 1+1 = 2$$

$$\|\delta(V_1, V_2)\| = \sum_{(i,j) \in \delta(V_1,V_2)} \gamma_{ij} = 1+1 = 2$$

(b) Minimum edge-cut

Figure 2.5: An example of a graph partitioning problem.

$$\|V_n\| = \sum_{i \in V_n} \omega_i \leq B_{aw} \qquad \forall \quad n = 1 : k \tag{2.2}$$

(i.e., the weight of each subdomain is bounded),

$$\frac{\max(\|V_1\|, ..., \|V_k\|)}{\min(\|V_1\|, ..., \|V_k\|)} \leq B_{rw} \tag{2.3}$$

(i.e., the weight imbalance ratio is bounded) and

$$\|\delta(V_1, ..., V_k)\| = \sum_{(i,j) \in \delta(V_1,...,V_k)} \gamma_{ij} \tag{2.4}$$

(i.e., the edge-cut) is minimised.

The previous problem can be formulated as an *integer linear programming* (ILP) problem, for which there exist several different formulations. In contrast to *linear programming* (LP), the selection of a good model in ILP is of crucial importance for solving the problem. In the following paragraphs, three different ILP models are presented for the CPAP as a result of different actions to simplify or improve the initial formulation.

General Model

A natural starting formulation is one that defines variables to describe which vertices and edges lie within each cluster [39][40]. Let X_{in} be binary variables that reflect the decision to assign vertex i to subdomain n, such that

$$X_{in} = \begin{cases} 1 & \text{if } \pi_i = n, \\ 0 & \text{otherwise,} \end{cases} \tag{2.5}$$

where π_i is the subdomain to which vertex i belongs. Let Y_{ij} also be binary variables that reflect whether the edge (i,j) does not contribute to the overall edge-cut, such that

$$Y_{ij} = \begin{cases} 1 & \text{if } \pi_i = \pi_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

The single-homing constraint forces that each cell must be assigned to only one PCU, and therefore

$$\sum_{n=1}^{k} X_{in} = 1 \qquad \forall\, i = 1 : |V|. \tag{2.7}$$

The maximum number of GPRS TSLs in a PCU is limited by physical hardware capabilities, which leads to a constraint on the maximum subdomain weight as

$$\sum_{i=1}^{|V|} \omega_i X_{in} \leq B_{aw} \qquad \forall\, n = 1 : k. \tag{2.8}$$

Following operator demands, the load must be evenly balanced among PCUs. Thus, a maximum weight imbalance is permitted between PCUs. This constraint can be expressed as

$$\frac{\sum\limits_{i=1}^{|V|} \omega_i X_{im}}{\sum\limits_{i=1}^{|V|} \omega_i X_{in}} \leq B_{rw} \qquad \forall\, m, n = 1 : k,\ m \neq n, \tag{2.9}$$

which can be easily transformed into the linear constraint

$$\sum_{i=1}^{|V|} \omega_i X_{im} - B_{rw} \sum_{i=1}^{|V|} \omega_i X_{in} \leq 0 \qquad \forall\, m, n = 1 : k,\ m \neq n. \tag{2.10}$$

The variables X_{in} and Y_{ij} are not independent, but are logically connected by a conjunctive operator in a constraint of the form

$$Y_{ij} = \sum_{n=1}^{k} X_{in} X_{jn} \qquad \forall\, i, j = 1 : |V|. \tag{2.11}$$

Thus, Y_{ij} is 1 if X_{in} and X_{jn} are 1 for any n. Constraint (2.11) is not linear, but quadratic. The underlying problem is thus a quadratic integer programming problem, which can not be solved by standard ILP techniques. However, it is possible to replace (2.11) by a linear program by extending the variable and constraint set [40]. Let Z_{ijn} be new binary variables defined as

$$Z_{ijn} = X_{in} X_{jn} \qquad \forall\, i, j = 1 : |V|,\; n = 1 : k, \tag{2.12}$$

which equals 1 if vertices i and j are assigned to subdomain n, 0 otherwise. This non-linear constraint can be disaggregated into the group of linear constraints

$$\begin{aligned}
&Z_{ijn} \le X_{in} && \forall\, i, j = 1 : |V|,\; n = 1 : k, && (2.13)\\
&Z_{ijn} \le X_{jn} && \forall\, i, j = 1 : |V|,\; n = 1 : k, && (2.14)\\
&Z_{ijn} \ge X_{in} + X_{jn} - 1 && \forall\, i, j = 1 : |V|,\; n = 1 : k, && (2.15)\\
&Z_{ijn} \ge 0 && \forall\, i, j = 1 : |V|,\; n = 1 : k. && (2.16)
\end{aligned}$$

The objective function can be interchangeably defined as

$$\text{Minimise} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \gamma_{ij} (1 - \sum_{n=1}^{k} Z_{ijn}) \tag{2.17}$$

(i.e., minimise the edge-cut) or

$$\text{Maximise} \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \gamma_{ij} \sum_{n=1}^{k} Z_{ijn} \tag{2.18}$$

(i.e., maximise the weight of edges that join vertices in the same subdomain).

The previous equations can be simplified by taking advantage of the symmetry of the problem. Thus, $Z_{ijn} = Z_{jin}$ and γ_{ii}=0, and the former variables can be restricted to those combinations of i and j in the upper triangular matrix of the adjacency matrix. Therefore, (2.12) can be rewritten as

$$Z_{ijn} = X_{in} X_{jn} \quad \forall\, i = 1 : (|V| - 1)\,,\; j = 2 : |V|,\; i < j,\; n = 1 : k, \tag{2.19}$$

where it is observed that the number of these variables has been reduced from $|V|^2$ to $\frac{|V|(|V|-1)}{2}$. Likewise, the adjacency matrix is symmetrical (i.e., $\gamma_{ij} = \gamma_{ji}$), as CPAP graphs are undirected. As a result, the objective function can be expressed as

$$\begin{aligned}
&\text{Minimise} \textstyle\sum_{i=1}^{|V|-1} \sum_{j=i+1}^{|V|} \gamma_{ij} (1 - \sum_{n=1}^{k} Z_{ijn}) \quad \text{or}\\
&\text{Maximise} \textstyle\sum_{i=1}^{|V|-1} \sum_{j=i+1}^{|V|} \gamma_{ij} \sum_{n=1}^{k} Z_{ijn},
\end{aligned} \tag{2.20}$$

where indices have been restricted to the upper triangular matrix.

For clarity, the resulting model, denoted as *general model* (GM), is summarised as follows:

$$
\begin{aligned}
(GM)\quad \text{Min}\quad & \sum_{i=1}^{|V|-1}\sum_{j=i+1}^{|V|}\gamma_{ij}(1-\sum_{n=1}^{k}Z_{ijn}) && && (2.21)\\
\text{s.t.}\quad & \sum_{n=1}^{k}X_{in}=1, && \forall\, i\in V, && (2.22)\\
& \sum_{i=1}^{|V|}\omega_i X_{in}\le B_{aw}, && \forall\, n\in N, && (2.23)\\
& \sum_{i=1}^{|V|}\omega_i X_{im}-B_{rw}\sum_{i=1}^{|V|}\omega_i X_{in}\le 0, && \forall\, m,n\in N,\ m\neq n, && (2.24)\\
& Z_{ijn}\le X_{in}, && \forall\,(i,j)\in\mathbf{U},\ n\in N, && (2.25)\\
& Z_{ijn}\le X_{jn}, && \forall\,(i,j)\in\mathbf{U},\ n\in N, && (2.26)\\
& Z_{ijn}\ge X_{in}+X_{jn}-1, && \forall\,(i,j)\in\mathbf{U},\ n\in N, && (2.27)\\
& X_{in}\in\{0,1\}, && \forall\, i\in V,\ n\in N, && (2.28)\\
& Z_{ijn}\in\{0,1\}, && \forall\,(i,j)\in\mathbf{U},\ n\in N. && (2.29)
\end{aligned}
$$

where $\mathbf{U}$ is the upper triangular adjacency matrix and $N=\{1,2,\cdots,k\}$.

From (2.5) and (2.19), it is deduced that the number of variables X_{in} and Z_{ijn} in (GM) is

$$N_{var\,GM}=k|V|+k\frac{|V|\,(|V|-1)}{2}=k\left(|V|+\frac{|V|\,(|V|-1)}{2}\right), \tag{2.30}$$

where k is the number of subdomains and $\frac{|V|\,(|V|-1)}{2}$ is the size of the upper triangular matrix of the adjacency matrix. Likewise, it can be deduced from (2.22)-(2.29) that the number of constraints in the model is

$$N_{const\,GM}=N_{var\,GM}+|V|+k+2\sum_{n=1}^{k-1}n+3k\frac{|V|\;(|V|-1)}{2}\,, \tag{2.31}$$

where $N_{var_{GM}}$ of these constraints are the integrality constraints on the binary variables in (2.28) and (2.29).

Compact Model

The (GM) model has considered a complete graph (i.e., $\gamma_{ij}\neq 0$). Thus, the number of variables and constraints is dominated by the size of the upper triangular adjacency matrix (i.e., $\frac{|V|\,(|V|-1)}{2}$) when combined with the number of subdomains (i.e., k). For efficiency, it is beneficial to take advantage of the sparse nature of the adjacency matrix. In this application, the number of active adjacencies (i.e., those where HOs take place), $|E|$, is limited by both configuration and propagation reasons, and thus $|E| << \frac{|V|\,(|V|-1)}{2}$. Therefore, the number of adjacencies in the model is reduced from $\frac{|V|\,(|V|-1)}{2}$ to $|E|$ by considering only active adjacencies. Under this assumption, (2.21) and (2.25)-(2.27),(2.29) are rewritten as

$$\text{Min} \sum_{(i,j)\in E} \gamma_{ij}(1 - \sum_{n=1}^{k} Z_{ijn}) \tag{2.32}$$

and

$$\begin{aligned}
&Z_{ijn} \leq X_{in}, && \forall\,(i,j) \in E,\ n \in N, && (2.33)\\
&Z_{ijn} \leq X_{jn}, && \forall\,(i,j) \in E,\ n \in N, && (2.34)\\
&Z_{ijn} \geq X_{in} + X_{jn} - 1, && \forall\,(i,j) \in E,\ n \in N, && (2.35)\\
&Z_{ijn} \in \{0,1\}, && \forall\,(i,j) \in E,\ n \in N, && (2.36)
\end{aligned}$$

respectively. Thus, the number of variables and constraints in this model, denoted as *compact model* (*CM*), is reduced to

$$N_{var\,CM} = k|V| + k|E| = k(|V| + |E|)\ , \tag{2.37}$$

$$N_{const\,CM} = N_{var\,CM} + |V| + k + 2\sum_{n=1}^{k-1} n + 3k|E|\ , \tag{2.38}$$

and the size of the model becomes O($k(|V| + |E|)$).

Compact Model with Less Symmetry

The (*GM*) and (*CM*) models necessarily have multiple optimal solutions, since the interchange of subdomain indices, n, leads to indistinguishable solutions. This symmetry is known to degrade the performance of enumeration algorithms used to solve the ILP model [41]. To avoid redundant solutions, any one vertex (e.g., υ) can be assigned to any one subdomain (e.g., V_1), [41]. This is easily accomplished by attaching the constraint $X_{\upsilon 1} = 1$. Fixing the assignment of a particular vertex to a subdomain implies that the set of variables $\{Z_{\upsilon jn}\,/(\upsilon, j) \in E,\ n \in N\}$ can be deleted from the model, together with the constraints on these variables. The number of deleted variables and constraints is proportional to the number of edges incident to vertex υ (i.e., $|E(\upsilon)|$). Therefore, υ should be a vertex of maximal degree (i.e., maximum number of incident edges). It is clear that the previous step does not eliminate symmetry completely, as subdomains other than V_1 are still indistinguishable. However, the inclusion of additional constraints, such as forcing that the number of vertices decreases with subdomain indices, have been reported to be not so effective [42].

As a result of these changes, the model proposed in this work, denoted as *compact model with less symmetry* (*CMS*), can be described as

$$
\begin{aligned}
(CMS)\ \ \text{Min}\quad & \sum_{(i,j)\in E}\gamma_{ij}-\left(\sum_{j\in V(\upsilon)}\gamma_{\upsilon j}X_{j1}+\sum_{(i,j)\in E-E(\upsilon)}\gamma_{ij}\sum_{n=1}^{k}Z_{ijn}\right) && (2.39)\\
\text{s.t.}\quad & \sum_{n=1}^{k}X_{in}=1, \qquad \forall\, i\in V,\ i\neq\upsilon, && (2.40)\\
& \sum_{i\in V,\, i\neq\upsilon}\omega_i X_{i1}\leq B_{aw}-\omega(\upsilon), && (2.41)\\
& \sum_{i\in V}\omega_i X_{in}\leq B_{aw}, \qquad \forall\, n\in N,\ n\neq 1, && (2.42)\\
& \sum_{i\in V,\, i\neq\upsilon}\omega_i X_{i1}+\omega_\upsilon-B_{rw}\sum_{i\in V,\, i\neq\upsilon}\omega_i X_{in}\leq 0, \qquad \forall\, n\in N,\ n\neq 1, && (2.43)\\
& \sum_{i\in V,\, i\neq\upsilon}\omega_i X_{in}-B_{rw}\left(\sum_{i\in V,\, i\neq\upsilon}\omega_i X_{i1}+\omega_\upsilon\right)\leq 0, \qquad \forall\, n\in N,\ n\neq 1, && (2.44)\\
& \sum_{i\in V,\, i\neq\upsilon}\omega_i X_{im}-B_{rw}\sum_{i\in V,\, i\neq\upsilon}\omega_i X_{in}\leq 0,\\
& \qquad\qquad \forall\, m,n\in N,\ \ m,n\neq 1,\ m\neq n, && (2.45)\\
& Z_{ijn}\leq X_{in}, \qquad \forall\,(i,j)\in E{-}E(\upsilon),n\in N, && (2.46)\\
& Z_{ijn}\leq X_{jn}, \qquad \forall\,(i,j)\in E{-}E(\upsilon),n\in N, && (2.47)\\
& Z_{ijn}\geq X_{in}+X_{jn}-1, \qquad \forall\,(i,j)\in E{-}E(\upsilon),n\in N, && (2.48)\\
& X_{in}\in\{0,1\}, \qquad \forall\, i\in V,\ i\neq\upsilon,\ n\in N, && (2.49)\\
& Z_{ijn}\in\{0,1\}, \qquad \forall(i,j)\in E{-}E(\upsilon),n\in N, && (2.50)
\end{aligned}
$$

where $V(\upsilon)$ and $E(\upsilon)$ are the neighbour vertices and the incident edges of vertex υ, respectively. Briefly, (2.39) presents the objective function, (2.40) ensures that each cell belongs to only one PCU, (2.41) and (2.42) reflect the capacity limitations of the PCU, (2.43)-(2.45) ensure the maximum weight imbalance between PCUs, (2.46)-(2.48) ensure the correct relationship between variables in the model, and (2.49)-(2.50) are the binary constraints.

The number of variables and constraints in (CMS) is

$$
\begin{aligned}
N_{var\,CMS} &= N_{var\,CM}-k(1+|E(\upsilon)|), && (2.51)\\
N_{const\,CMS} &= N_{const\,CM}-k(1+|E(\upsilon)|)-(1+3k|E(\upsilon)|)\\
&= N_{const\,CM}-(1+k+4|E(\upsilon)|). && (2.52)
\end{aligned}
$$

It is worth noting that, although there exist more compact formulations of the GPP, such as the ones described in [43], it was shown there that they are less efficient for values of $k>3$.

2.2.4 Current State of Solution Techniques

Once the problem has been formulated, the following paragraphs outline the state of research and technology for solving the problem. While the former is focused on methods for the GPP, the latter discusses the methodology and tools currently used by operators to solve the CPAP.

State of Research

From the theoretical perspective, the CPAP can be formulated as a *bounded, min k-cut* problem for k fixed. The simplest approach to solve this problem exactly is the brute-force enumeration of all possible solutions. The evaluation of the entire solution space certainly yields to the optimal solution. Unfortunately, this approach is impractical for graphs of reasonable size, since the size of the solution space is $k^{|V|}$. To circumvent this difficulty, other exact methods aim to reduce the size of the solution space that must be explicitly enumerated. Thus, the algorithm proposed by Goldschmidt and Hochbaum [44][45] solves the problem with a running time of $O(|V|^{k^2})$. Nonetheless, even though this algorithm is polynomial in $|V|$, it is still exponential in k. Alternatively, the *bounded, min k-cut* problem can be formulated as an ILP model, which can be solved exactly. The *Branch-and-Cut* method, described in the next section, is the algorithm used to solve this model in most commercial optimisation packages [46].

In spite of all efforts to improve efficiency, most exact methods are still computationally intensive. As a consequence, several heuristic methods have been proposed to find approximate solutions to the problem efficiently (for a comprehensive survey, see [21]). Amongst all, the *Kernighan-Lin* (KL) algorithm [47] is the common benchmark against which more refined methods are compared. Other promising techniques are the *Multi-Level Refinement* [48][25][26][49][50][51] and *Adaptive Multi-Start* [52][28] algorithms. These heuristic methods have recently gained momentum driven by applications in the supercomputing, integrated circuit design and internetworking areas. In the context of cellular networks, these methods have been applied to the assignment of cells to switches and location area planning during the design stage of the network [40][53][54][55][56][57][58]. However, the PCU planning problem has not been addressed in the literature yet.

State of Technology

In the equipment provided by manufacturers, the process of assigning cells to PCUs can be carried out manually or automatically, based on the operator's choice.

The automatic assignment algorithm is triggered when the operator enables GPRS on a cell. After this event, the algorithm checks if there is any other cell in the BSC that has a strong relationship with the original cell, either because it shares the site or it is defined as an adjacency. If this is the case, the algorithm assigns the original cell to the PCU that comprises cells with which it shares the highest overall number of defined adjacencies, provided that the number of GPRS TSLs in the PCU is below a certain ratio of the maximum capacity (2/3*100=67% typ.). If there are no related cells or the load of the PCU initially targeted is excessive, the cell is assigned to the least loaded PCU.

As main drawback of the automated algorithm, the solution is clearly dependent on the order in which cells in a BSC are activated. To circumvent this problem, it is usually recommended that, during the initial deployment of the network, operator enables first cells of very different geographical areas, so that they can be assigned manually to different PCUs. Another drawback is the fact that the assignment procedure is based on static configuration data established during planning stage (i.e., defined adjacencies) and not on network statistics gathered during the operational stage (i.e., actual user mobility trends). In addition, even though preliminary versions did entail the possibility of re-allocating cells to balance the load among PCUs, current releases do not include such a mechanism. Thus, no reactive action is taken when PCUs become overloaded due to subsequent extensions of network resources. As a result, these algorithms tend to group most cells in a single PCU, which, as time goes by, reaches its maximum capacity. Cells enabled after this event are necessarily assigned to new PCUs, regardless of the neighbourhood relationship with existing cells.

Due to the shortcomings of the automatic approach, operators are forced to use the manual assignment procedure. In EDGE, the use of this approach is unavoidable, since it is the only one in existence. In this approach, the diversity of network data spread over different database tables in the NMS and the complexity of solution techniques cause that sub-optimal solutions are generally adopted. Likewise, the optimisation of an existing PCU plan is hardly ever considered and, consequently, the number of changes in the PCU plan is restricted to the addition of a new cell or PCU. New cells are normally assigned to the PCU with the largest number of common adjacencies, as the automatic method would do. As a result, the inter-PCU CRS ratio in the network is high and the load is unevenly distributed among PCUs. Analysis of real data, presented in section 2.4, shows evidence that current network configuration is far from optimal.

Once the possibility to formulate the CPAP as an ILP model has been identified, there exist several commercial applications that can be used to solve the problem exactly. Amongst other are CPLEX [59], Xpress-MP [60] and LINDO [61]. In the graph partitioning field, there exist a number of tools that develop several heuristic methods. Amongst the most spread are CHACO [62], PARTY [63], METIS [64], SCOTCH [65] and JOSTLE [66]. Some of these applications have been used in this thesis, either as part of the proposed method (i.e., CPLEX) or for benchmarking purposes (i.e., METIS and JOSTLE).

2.3 Method of Assigning Packet Control Units

This section starts with a brief discussion of how the complexity of the CPAP affects the selection of a solution technique. Two alternative methods are then presented to solve the CPAP. The first method aims to solve the ILP model of the underlying GPP. The second method combines several heuristic techniques to obtain approximate solutions efficiently. Finally, the algorithmic complexity of both approaches is evaluated.

2.3.1 Preliminary Issues

As part of network re-planning routine, operators have to solve periodically the CPAP in every BSC of the network. Unlike the initial planning stage, where the problem is solved just once, during the operational stage, this process must be performed every time a new cell or PCU is added to the network, or a significant change is detected in user mobility trends. Hence, any solution method faces the challenge to solve a set of instances of the CPAP (i.e., one per BSC in the network) in a limited time (e.g., a week's time). Otherwise, operators would be forced to either restrict the geographical area under optimisation or the periodicity with which the method is applied.

In the context of live cellular networks, it is envisaged that the size of the PCU planning problem prevents computationally-expensive methods from being applied. The number of cells, adjacencies and BSCs (i.e., vertices, edges and problem instances, respectively) rapidly increases when a whole operator network is considered. For instance, a typical GERAN comprises 200 BSCs, each one containing 150 cells and 1800 defined adjacencies on average. Therefore, the speed of the method is critical, since the higher the time complexity of computation, the less the tool will be used as it increases the workload on the NMS.

From the theoretical perspective, most formulations of the graph partitioning problem are known to be $\mathcal{NP}$-complete. The bounded, min-cut partitioning problem is $\mathcal{NP}$-complete for arbitrary k [16]. For fixed k, it is solvable in polynomial time, but existing algorithms have a high-order polynomial complexity in the number of vertices, which is still exponential in k [45]. The bounded, connected, min k-cut problem is also $\mathcal{NP}$-complete, since it can be reduced by transformation to a complete graph by adding zero-weight edges to the integer bin-packing problem, which is known to be $\mathcal{NP}$-complete [16].

It can thus be concluded that the large size of the network to be optimised, the frequency with which the method must be applied and the high complexity of exact methods justify the development of heuristic algorithms. For the same reason, computationally-expensive heuristics have been discarded in this work for ease of implementation in live networks. Nonetheless, due to the academic nature of this study, exact approaches are also considered for benchmarking purposes. In this sense, the limited size of the graphs that model the CPAP, unlike in other applications, makes this approach feasible.

2.3.2 Exact Method

The following paragraphs describe the method to solve the CPAP exactly. The description begins with the state-of-the-art algorithms proposed in the literature to solve the ILP problem. The classical method is then extended to deal with runtime constraints in a live environment.

Original Exact Method

The basic algorithm used to solve mixed-integer linear programming problems in most available commercial tools is the *Branch-and-Bound* (BB) algorithm [67]. This algorithm follows an enumerative approach that searches the complete space of solutions for the best solution of a given problem. However, as explicit enumeration is normally impossible due to the exponentially increasing number of solutions, the algorithm must search some parts of the solution space only implicitly. This implicit enumeration can be performed by using bounds for the function to be optimised, together with the value of the current best solution.

The BB algorithm is based on the 'divide and conquer' principle. The method solves the original problem by solving a series of subproblems. The term *subproblem* denotes a problem derived from the original one through addition of constraints. These constraints are used to divide the complete solution space into complementary subregions. Thus, the solution of a problem with a BB algorithm is traditionally described as a search through a search tree, in which the root corresponds to the original problem and each other node corresponds to a problem derived from the original problem. As the algorithm progresses, the number of additional constraints increases and so does the number of subproblems. Consequently, the search tree is developed dynamically. In the following description, a minimisation problem is considered: the problem is to minimise an *objective function* over a region of *feasible solutions*. $\mathbf{X} = [x_1\ x_2\ \cdots\ x_n]$ is a solution vector with the values of the decision variables, x_i, while z denotes the value of the objective function of a particular solution.

The method starts by considering the original problem. For this problem, lower and upper bounds for the optimal value are calculated (referred to as *bounding*). If both bounds coincide and are feasible, an optimal solution has been found and the procedure terminates. Otherwise, the original problem is subdivided into several problems (referred to as *branching*), which is accomplished by subdividing the search space into two or more regions that together cover the whole feasible region. The algorithm is applied recursively to the problems, generating a tree of problems. At any moment, there exists a collection of problems that must still be solved, which are referred to as *active problems*. Bounds for the problems are progressively tightened as branching progresses. Thus, the difference between the lower and upper bound is a measure of proximity to the optimal solution (when it exists). If an optimal feasible solution is found to a problem in the tree, it is a feasible solution to the original problem, but it is not necessarily globally optimal. These feasible solutions can be used to prune those branches whose lower bound is higher than the best value found so far (referred to as *incumbent*). The search proceeds until there is no unexplored parts of the solution space (i.e., all nodes have been solved or pruned) or until some specified threshold is met between the best feasible value found so far and the lower bounds on all unsolved subproblems.

From the previous discussion, it can be deduced that a BB algorithm consists of three main components:

a) a *bounding function* that provides a lower and upper bound for the best value of a given problem;

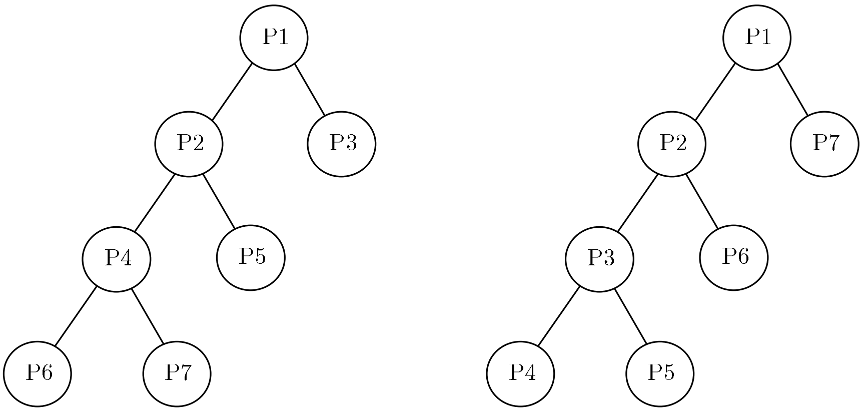
b) a *branching rule* to divide a subspace if it cannot be discarded; and

c) a *strategy for selecting the next solution subspace* to be investigated.

Amongst all, the bounding function is the key component, since it is responsible for keeping the size of the search tree as small as possible. Therefore, a weak bounding function cannot be compensated for by good branching and problem selection strategies.

The *bounding function* must be easy to solve and provide tight bounds. An upper bound of the optimal value of a problem can be easily derived from any feasible solution. Hence, the actual problem is to calculate a lower bound for the objective function under some given constraints in polynomial time. For this purpose, either the set of constraints or the objective function must be modified. The first approach eliminates some of the constraints of the original problem, enlarging the set of feasible solutions to the problem. The optimal solution to this relaxed problem is a lower bound (not necessarily feasible) for the optimal value of the original problem. The most common relaxation is the discard of the integrality constraint. Thus, the LP relaxation of an ILP problem gives a lower bound to the latter problem. Alternatively, the objective function (and not the constraint set) can be modified, as proposed in [68]. Both approaches can also be combined, thus leading to very tight but computationally-expensive bounds.

The *selection of the next problem* from the list of active problems must ensure that good feasible solutions are found early in the search. Thus, a good solution is already available if time



(a) Breadth-First Search (b) Depth-First Search

Figure 2.6: A comparison of search strategies in a tree of subproblems.

runs out, which can be used to prune useless nodes in the tree. For this purpose, the selection mechanism can follow a static or adaptive rule to traverse the nodes of the tree. Among the former methods are *Breadth-First Search* (BFS) and *Depth-First Search* (DFS) strategies [69]. Figure 2.6 illustrates the difference between both strategies by representing the order in which the subproblems are visited in a simple search tree. BFS begins at the root node and explores all the adjacent nodes. All nodes at a given level are considered before any node at the next lower level. By contrast, DFS starts at the root node and explores each problem as far as possible before backtracking (i.e., going back on the path toward the root until finding a node that has a son that has not yet been considered). The latter approach is able to provide tighter bounds faster, which can later be used to discard other active problems, and it is therefore the default option in most commercial codes. However, when the current node is pruned, the next node is not generally determined by the static backtracking strategy, but, on the contrary, the selection is based on the status of the active problems. For this purpose, a reasonable criterion is the selection of the problem that is more likely to contain an optimal solution or find quickly a feasible solution.

Finally, the *branching operation* is based on the lower bound of the problem that remains unsolved. Since this lower bound usually corresponds to the LP relaxation of the problem, it is associated to a solution that does not fulfill the integrality constraint in one or more variables (referred to as fractional variables). The latter variables take values that can be represented by the combination of an integer and a fractional part, i.e., $x_i = a.b$, where $a \in \mathbb{N}$, $0 \leq b \leq 1$ and $b \in \mathbb{R}$. Two new problems are then created by selecting one of the fractional variables and alternately adding the constraints $x_i \leq a$ and $x_i \geq a + 1$ to the original problem. In case all variables in the model are binary (i.e., $x_i \in [0, 1]$), the branching operation is reduced to fixing the selected variable to 0 and 1 (i.e., $x_i = 0$ and $x_i = 1$). The selection of the variable where the branching takes place when there exists more than one with fractional values can be critical in keeping the size of the search tree small. If the problem structure is known, a priority order on the variables can be set a priori by favouring the branching on the most important variables. When no such information is available, the simplest rule is to select a variable whose fractional part is closest to $\frac{1}{2}$ in order to solve the maximum integer infeasibility. Instead, commercial solvers select the fractional variable that causes the LP objective function to deteriorate faster, in an attempt to prune one or both of the children.

The template of the basic BB algorithm for an ILP problem with binary variables is summarised in Figure 2.7. The operation of the algorithm is illustrated by an example, where the following model is solved:

$$\begin{array}{ll}\text{Minimise} & 5x_1 + 10x_2 + 2x_3 \\ \text{subject to} & \quad\;\; x_2 - x_3 \geq 0 \\ & x_1 - x_2 \qquad\;\; \leq 0 \\ & x_1 + x_2 + x_3 \geq 1 \\ & x_1, x_2, x_3 \in \{0, 1\}.\end{array}$$

Figure 2.8 shows the evolution of the search tree in this problem instance. For clarity, some additional information has been included in the figure. Nodes are progressively numbered to reflect the order in which they are visited. For any problem solved, p_j, the optimal solution, $\mathbf{X}_{LP}^j$, and its value, z_{LP}^j, for the LP relaxation are presented. In addition, the upper bound for the original problem, $\overline{z}_{ILP}$ is continuously updated. The value of the branching variable, x_i, is shown on top of the edges between a father and its child nodes. Also, a line below the node has been superimposed to reflect that the node has been pruned in the past, together with the reason for discarding it. The algorithm first considers the original problem (label 0), initialising $\overline{z}_{ILP} = \infty$ and solving its LP relaxation by conventional methods (e.g., the simplex method [70]). Since the LP solution, $\mathbf{X}_{LP}^0$, is not integer (i.e., $x_1 = x_2 = x_3 = 0.33$), the lower bound of the original problem is updated to the LP value, z_{LP}^0, and branching is performed on the first fractional variable with the maximum integer infeasibility (i.e., x_1). Two new problems are then created by alternately fixing x_1=0 and x_1=1, and the former (label 1) is selected by the DFS strategy. The LP solution to this problem, $\mathbf{X}_{LP}^1$, is still fractional on x_2 and x_3 (i.e., $x_2 = x_3 = 0.5$), and its value $z_{LP}^1 = 6$. Again, two problems are added to the active list by fixing x_2=0 and x_2=1, and the first one (label 2) is selected by the DFS strategy. The LP relaxation of the problem is not feasible and the node is pruned. The backtracking mechanism of DFS leads to the selection of the node where x_1=0 and x_2=1 (label 3). The LP solution of this problem, $\mathbf{X}_{LP}^3$, is integer, so it is also a solution of the original problem. Since no integer solution has been found so far, this solution becomes the best candidate and the upper bound of the original problem is tightened by setting $\overline{z}_{ILP} = z_{LP}^3 = 10$. Again, the backtracking mechanism leads to the node where x_1=1 (label 4). The LP relaxation does fulfill the integrality constraint, so the node does not need any further branching and it is pruned. However, it provides a value that is worse than the best candidate found so far, so it is finally discarded. The process ends by selecting the best candidate $\mathbf{X} = \mathbf{X}_{LP}^3 = [0\ 1\ 0]$ with $z_{ILP} = \overline{z}_{ILP} = z_{LP}^3 = 10$.

From the previous example, it is clear that the computational efficiency of the BB algorithm greatly depends on the size of the search tree. It is therefore crucial to reduce the number of visited nodes by keeping the number of branching operations to a minimum. This goal is equivalent to reducing the likelihood that branching conditions hold:

a) A fractional value is present in the integer variables on the solution to the LP relaxation of the current problem (i.e., $\mathbf{X}_{LP}^j(i) \notin \mathbb{N}$, for some i).

b) The lower bound for the current problem is lower than the value for the best candidate solution (i.e., $z_{LP}^j < \overline{z}_{ILP}$).

From the latter condition, it is clear that the faster the bounds are tightened, the smaller is the size of the tree. Concretely, branching can be avoided by either increasing the lower bound of

```
Step 1) Initialisation
    1.1) Initialise the list of active problems with the original problem.
    1.2) Set lower bound to $-\infty$ and upper bound to $\infty$ for the original problem.

Step 2) Termination test
    2.1) if the list of unsolved problems is empty, then the optimisation process stops and
           i. if there exist a best candidate solution, then
          ii.     it is the optimal solution to the original problem;
         iii. else the original problem is infeasible.

Step 3) Problem selection and relaxation
    3.1) Select next problem by DFS of the list of active problems.
    3.2) Solve the LP-relaxation of the problem.

Step 4) Bounds update
    4.1) if the optimal LP solution to the current problem is integer and its value is lower than the
         upper bound of the original problem, then
           i. the upper bound of the original problem is updated to the optimal LP value,
          ii. the optimal LP solution becomes the best candidate solution found so far;
    4.2) elseif the LP solution to the current problem is not integer and its value falls below the
         upper bound of the original problem, then
           i. the lower bound of the current problem is updated to the LP value.

Step 5) Pruning
    5.1) if the LP solution to the current problem is integer, then
           i. if the upper bound of the original problem was updated in the previous stage, then
              delete active problems with a lower bound higher than the new upper bound;
          ii. go to stage 2.
    5.2) if the LP solution to the current problem is not integer and its value is higher than the
         upper bound of the original problem, then go to stage 2;
    5.3) if the current problem is infeasible, then go to stage 2.

Step 6) Branching
    6.1) Branch on the integer variable $x_i$ with a fractional value closest to 1/2 in the LP solution.
           i. Add two new problems to the list of active problems by alternately fixing $x_i = 0$ and
              $x_i = 1$ in the current problem and go to step 3.
```

Figure 2.7: Template of the basic Branch-and-Bound algorithm.

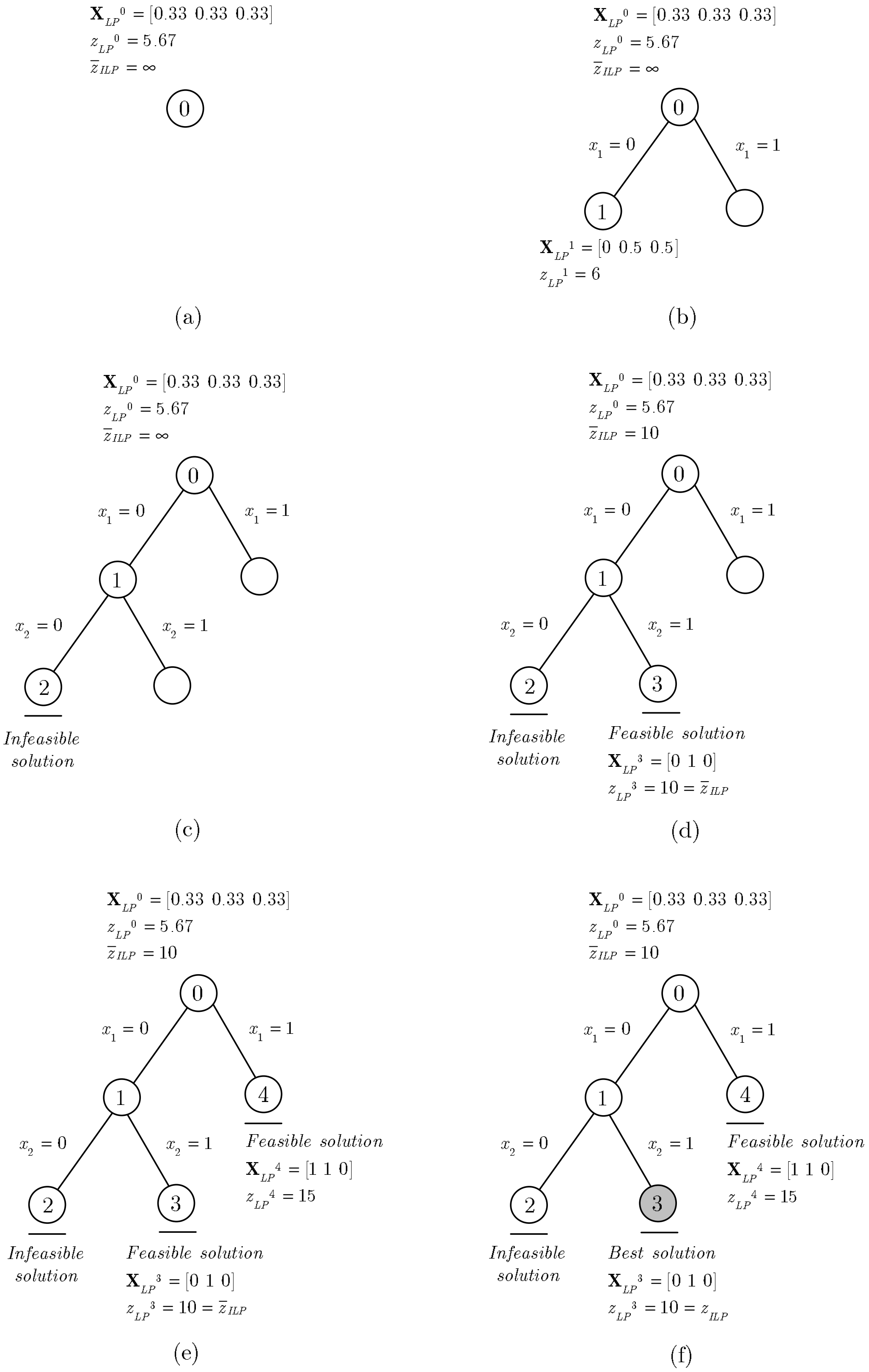

$\mathbf{X}_{LP}{}^{0} = [0.33\ 0.33\ 0.33]$
$z_{LP}{}^{0} = 5.67$
$\overline{z}_{ILP} = \infty$
0
(a)
$\mathbf{X}_{LP}{}^{0} = [0.33\ 0.33\ 0.33]$
$z_{LP}{}^{0} = 5.67$
$\overline{z}_{ILP} = \infty$
0
$x_1 = 0$
$x_1 = 1$
1
$\mathbf{X}_{LP}{}^{1} = [0\ 0.5\ 0.5]$
$z_{LP}{}^{1} = 6$
(b)
$\mathbf{X}_{LP}{}^{0} = [0.33\ 0.33\ 0.33]$
$z_{LP}{}^{0} = 5.67$
$\overline{z}_{ILP} = \infty$
0
$x_1 = 0$
$x_1 = 1$
1
$x_2 = 0$
$x_2 = 1$
2
Infeasible solution
(c)
$\mathbf{X}_{LP}{}^{0} = [0.33\ 0.33\ 0.33]$
$z_{LP}{}^{0} = 5.67$
$\overline{z}_{ILP} = 10$
0
$x_1 = 0$
$x_1 = 1$
1
$x_2 = 0$
$x_2 = 1$
2
3
Infeasible solution
Feasible solution
$\mathbf{X}_{LP}{}^{3} = [0\ 1\ 0]$
$z_{LP}{}^{3} = 10 = \overline{z}_{ILP}$
(d)
$\mathbf{X}_{LP}{}^{0} = [0.33\ 0.33\ 0.33]$
$z_{LP}{}^{0} = 5.67$
$\overline{z}_{ILP} = 10$
0
$x_1 = 0$
$x_1 = 1$
1
4
$x_2 = 0$
$x_2 = 1$
Feasible solution
$\mathbf{X}_{LP}{}^{4} = [1\ 1\ 0]$
$z_{LP}{}^{4} = 15$
2
3
Infeasible solution
Feasible solution
$\mathbf{X}_{LP}{}^{3} = [0\ 1\ 0]$
$z_{LP}{}^{3} = 10 = \overline{z}_{ILP}$
(e)
$\mathbf{X}_{LP}{}^{0} = [0.33\ 0.33\ 0.33]$
$z_{LP}{}^{0} = 5.67$
$\overline{z}_{ILP} = 10$
0
$x_1 = 0$
$x_1 = 1$
1
4
$x_2 = 0$
$x_2 = 1$
Feasible solution
$\mathbf{X}_{LP}{}^{4} = [1\ 1\ 0]$
$z_{LP}{}^{4} = 15$
2
3
Infeasible solution
Best solution
$\mathbf{X}_{LP}{}^{3} = [0\ 1\ 0]$
$z_{LP}{}^{3} = 10 = z_{ILP}$
(f)

Figure 2.8: Evolution of the search tree in the Branch-and-Bound algorithm.

active problems (i.e., z^j_{LP}) or decreasing the upper bound of the original problem (i.e., $\overline{z}_{ILP}$). While the former can be achieved by a proper choice of the branching variable (e.g., maximum degradation of LP objective function value), the latter is achieved by a proper choice of next active node (e.g., by DFS) or the provision of a good initial feasible solution (e.g., from a heuristic method).

To further tighten the lower bound of problems, the BB algorithm is usually combined with *cutting planes* methods [71]. This combination is commonly referred to as *Branch-and-Cut* (BC) algorithm. The fundamental idea behind cutting planes is to add constraints to the LP relaxation of a problem in order to tighten the lower bound without affecting the feasible region. To be effective, these additional constraints (i.e., referred to as *cuts*) must ensure that every feasible integer solution for the original problem is feasible for the cut and the current optimal solution for the relaxed problem is not feasible for the cut. The latter condition ensures that a tighter lower bound is achieved. Cutting planes are iteratively added until either an integer solution is found or it becomes impossible (or too expensive) to find another cutting plane. In the latter case, a traditional branching operation is performed and the search for cutting planes continues on the subproblems[2]. In most cases, the combination of branching and cutting planes significantly reduces the computational requirements, since it reduces the number of nodes visited in the search tree.

From the previous explanation, it is clear that the core of the previous algorithm is to find a violated cut, which is called the *separation problem.* For space reasons, the reader is referred to [71] and [72] for a comprehensive survey of methods to generate effective cuts. However, the following example tries to clarify how cuts can be used to tighten bounds given by the LP relaxation of a problem. Recall the previous model solved by the BB algorithm

$$\begin{array}{ll}\text{Minimise} & 5x_1 + 10x_2 + 2x_3 \\ \text{subject to} & \qquad\ \ x_2 \ - \ x_3 \geq 0 \\ & x_1 \ - \ x_2 \qquad\quad \leq 0 \\ & x_1 \ + \ x_2 \ + \ x_3 \geq 1 \\ & x_1, x_2, x_3 \in \{0,1\},\end{array}$$

whose LP relaxation had an optimal solution $x_1 = x_2 = x_3 = 0.33$ of value $z = 5.67$. By inspection of the constraints, it can be deduced that no feasible solution exists with $x_2 = 0$. Therefore, the constraint $x_2 \geq \frac{1}{2}$ is a valid inequality, since it does not affect the feasible region. Moreover, it is also a violated cut, since it is not satisfied by the previous optimal solution (i.e., $x_2 = 0.33 < \frac{1}{2}$). By adding this constraint, the new LP relaxation has an optimal solution $x_1 = 0, x_2 = x_3 = 0.5$ and $z = 6$. Thus, the lower bound has been tightened, but some variables still show fractional values. If the constraint $x_2 = 1$ is now added, the LP solution becomes $x_1 = 0, x_2 = 1, x_3 = 0$. Since this solution fulfills the integrality constraints, it is the optimal solution to the original problem. Thus, the optimal solution to the problem has been found without the need of branching. It is worth noting that, in this particular case, fixing $x_2 = 1$ is not to be considered as a branching, since this action does not affect the feasible region. For referral purposes, it can be added that the previous cuts belong to the family of Gomory-Chvátal inequalities, which is the best known class of cuts.

[2] *Pure cutting plane* algorithms are also an alternative to *Branch-and-Cut.* While the former methods add cuts to the root node until an optimal solution is found, the latter may perform branching to subdivide the problem into simpler problems.

Refinement of the Exact Method

Even if an exact solution is ensured by BC, the optimisation process might take excessive time. Thus, although the proposed method was initially conceived for benchmarking purposes, it is still aimed to find near-optimal solutions under operator's time constraints. Ideally, an instance of the CPAP must be solved every time a new cell or PCU is added to the network. Thus, an operator usually faces a set of CPAP instances (i.e., one per BSC in the network) that must be solved in a given period of time (e.g., several days). Hence, each problem instance must be solved under loose runtime constraints.

To speed up the optimisation process, an initial heuristic solution is fed to the BC algorithm. This solution gives an upper bound of the optimal value, which can be used to fix variables and to discard branches within the search tree. The number of visited subproblems is thus reduced. Ideally, such a heuristic method might identify the optimal solution and the BC algorithm would then only be used to verify the optimality. It is worth noting that, even though an existing solution may be configured in the network, it does not necessarily satisfy the formulated constraints. On the contrary, field trials have shown that solutions currently implemented in the network result in poor edge-cut performance and high imbalance amongst PCUs of the same BSC [20]. Therefore, the existing solution is of limited value. Instead, a multi-level refinement heuristic [73], described in the next section, is used to provide an initial solution to the problem.

Despite the previous improvement, the total runtime required by the BC algorithm to solve all instances optimally might still be too high. Therefore, the maximum runtime spent on each instance has to be limited. Although ideally this limitation might only entail a lack of an optimality proof, in practice, some performance impairment is observed. Thus, given the total available runtime, a heuristic must define the best share of runtime among instances to minimise the overall edge-cut impairment. In this work, two different time-sharing heuristics have been evaluated. Intuitively, those instances with a higher complexity should receive a larger share of the available time. However, the runtime of the BC algorithm on a particular instance is difficult to predict. As a rough estimation, the runtime can be assumed to be proportional to the size of the instance in terms of the number of variables and constraints in the ILP model. As will be shown later, the number of edges in CPAP graphs grows linear with the number of vertices and is much larger than the number of subdomains. Under this assumption, both the number of variables and constraints in (CMS) grows as $\mathrm{O}(k|E|)$, and so will be the share of runtime assigned to an instance. This strategy that assigns to instances a time-share proportional to its size is hereafter referred to as *size-based sharing* (SS) strategy. Alternatively, since the aim of the optimisation process is not the maximisation of the number of instances solved exactly, but the minimisation of the total edge-cut in the network, it might be preferable to spend most of the runtime in those instances where the edge-cut tends to be higher. This higher edge-cut might be a consequence of either a larger edge weight (e.g., higher user mobility) or greater difficulty of the optimisation problem (e.g., average TSL count per PCU close to PCU capacity limit). In this approach, instead of using the previous indicators directly, the edge-cut of a heuristic solution is used to predict the difficulties found by the BC algorithm to reduce the edge-cut in an instance. As a result, most of the computational effort in the exact method is spent on those instances where the heuristic approach failed to reduce the edge-cut, making the most of both approaches. This strategy that assigns to instances a time-share proportional to an estimation of the edge-cut of a heuristic solution is hereafter referred to as *edgecut-based sharing* (ES) strategy.

Given a period of time, T_{ov}, in which the CPAP must be solved for a network area that comprises several BSCs, the above-mentioned criteria result in the time shares

$$T_{ssj} = \frac{|E_j|\, k_j}{\sum\limits_{j=1}^{N_p} |E_j|\, k_j} \cdot T_{ov} \qquad \forall\, j = 1 : N_p\,, \tag{2.53}$$

$$T_{esj} = \frac{Q_j}{\sum\limits_{j=1}^{N_p} Q_j} \cdot T_{ov} \qquad \forall\, j = 1 : N_p\,, \tag{2.54}$$

where T_{ssj} and T_{esj} are the times reserved for problem instance j in SS and ES strategies, respectively, $|E_j|$ and k_j are the number of edges and subdomains in instance j, respectively, N_p is the number of problem instances (i.e., BSCs) and Q_j is the edge-cut of the heuristic solution to instance j.

2.3.3 Heuristic Method

The following paragraphs describe a method to solve the CPAP approximately based on heuristic techniques. For clarity, a classification of traditional graph partitioning techniques is given first. Then, the different techniques combined in the proposed method are progressively introduced. Finally, several refinements are proposed to consider the peculiarities of the CPAP.

Classification of Heuristic Graph Partitioning Techniques

Heuristic graph partitioning methods can be broadly classified into two categories: *geometric* and *combinatorial* approaches.

Geometric approaches make use of geometric coordinates associated with vertices to aid the partitioning of a graph. Geometric algorithms [74][75][76] attempt to group together vertices that are spatially near to each other, regardless of whether these vertices are not highly connected. In these algorithms, it is assumed that spatial proximity and vertex connectivity are strongly correlated. By nature, these algorithms are fairly easy to understand and to implement, but they rely on the availability of coordinates for vertices in the graph, which sometimes is not possible. Even if this piece of information is currently available for vertices in CPAP graphs (i.e., cell/site coordinates), its use entails the access to several NMS databases. This action would increase both the latency of the algorithm and the load on the NMS. More importantly, experience shows that, even though geographical proximity and connectivity are related, the correlation is not perfect. This fact justifies that assigning cells to PCUs based on a pure distance criterion does not always lead to the minimum inter-PCU HO ratio (i.e., edge-cut).

Alternatively, *combinatorial* (or *structural*) approaches compute partitions by referring only to the adjacency information of the graph. These methods attempt to group together vertices highly connected, regardless of whether or not these are close to each other in space. For this reason, the resulting partitions tend to have lower edge-cut and are less likely to contain disconnected subdomains when compared to those produced by geometric schemes. This quality enhancement is achieved at the expense of a larger execution time.

Although combinatorial techniques lead to solutions of better quality in terms of edge-cut, the sensitivity of these solutions to changes in the scenario is higher. This drawback stems from the fact that connectivity information might occasionally vary with time. Changes in user mobility trends (e.g., new routes) and propagation scenario (e.g., loss of line-of-sight conditions due to new obstacles) can produce large variations in the connectivity information. After such events, the performance of a solution that was optimal in the past might degrade significantly. Therefore, the combination of mobility and geographic criteria can increase the robustness of the solution against such events. The solutions obtained by this hybrid approach would be valid for a longer period.

The core of the method developed in this thesis follows a combinatorial approach. Thus, the method combines several of these techniques, namely *refinement algorithms* [47][77], *graph-walking algorithms* [26], *multi-level algorithms* [26][49] and *adaptive multi-start algorithms* [28]. Nonetheless, a refinement of the method is proposed to seize the geographical information of the network. Hence, the proposed method can be considered as a hybrid approach. The remainder of the section describes the solution techniques in the proposed method.

Refinement Algorithms

The most intuitive method to build a partition of a graph starts with an initial random partition that is progressively refined. Given a graph with a sub-optimal partitioning, the problem is to improve the edge-cut while maintaining the balance constraint. Such methods are commonly referred to as *refinement* or *local-search based algorithms*, and can be used indistinctively to build a solution to the GPP or to refine solutions from any other method.

The simplest refinement algorithm is the *Greedy* (or *Steepest Descent*) *Refinement* (GR) algorithm. This algorithm consists of a series of iterations through the vertices of the graph. On each iteration, for every vertex, all possible movements to other subdomains are evaluated. For every movement, the potential new edge-cut and the weight of the source and target subdomains are calculated. Those movements that do not fulfill all the constraints are discarded. This capability to handle constraints (such as prefixed assignments, uneven subdomain weight limits or subdomain connectivity) in the assignment process is one of the main strengths of the refinement methods. Among all the valid movements, the algorithm selects the one that achieves the largest edge-cut reduction. The partition is then updated and the refinement process proceeds until no candidate exists that provides an edge-cut improvement.

In the previous explanation, it has been assumed that the initial solution that is refined is valid (i.e., it satisfies all the weight constraints). Under this assumption, checks in the refinement process ensure the validity of the new solution. When the previous assumption does not hold, the decisions in the refinement process must initially be oriented to find a valid solution. The refinement algorithm developed in this work first attempts to enforce the absolute weight constraints by removing vertices in the border of the largest subdomain. Among all candidate movements, the one with the smallest edge-cut increase (and not the one with the largest weight decrease) is selected in order to reduce the impairment of solution quality. Once the absolute weight constraint is satisfied, the algorithm attempts to enforce the imbalance constraint by removing (adding) vertices in the border of the largest (smallest) subdomain, following the same principle of minimum edge-cut deterioration.

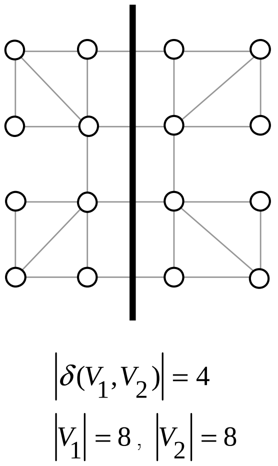


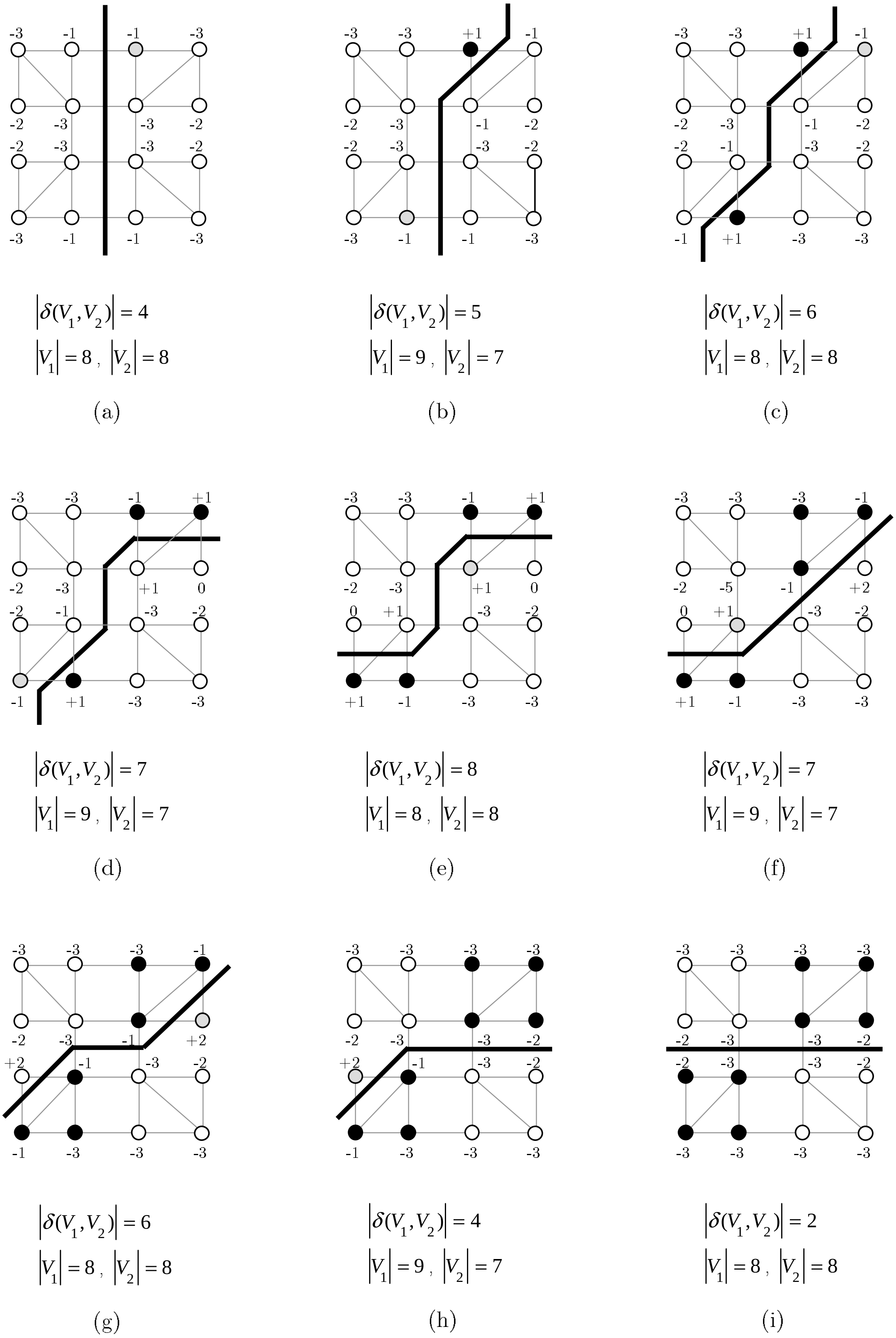
Figure 2.9: A bisection of a graph.

The greedy attribute of the algorithm stems from the fact that, on each step, the next movement is selected, regardless of the consequences that this decision might have in the final solution. Greedy algorithms belong to the family of local optimisation algorithms, and thus share with them the same drawbacks. The main disadvantage of these algorithms is their tendency to get trapped in local minima of the optimisation surface. These local minima are rather common in the CPAP, since vertices on the underlying graph tend to be clustered in groups that share a proximity relationship. This clustering effect implies that moving a single vertex of a cluster to an adjacent subdomain generally does not lead to any benefit, unless the remaining vertices in the cluster are also moved.

The previous problem can partly be solved by the *Kernighan-Lin* (KL) *refinement* algorithm [47]. Given a partition of a graph, the KL algorithm swaps subsets of vertices in different subdomains that yield the greatest possible edge-cut reduction. By considering pairs of vertices, and not individual vertices, the balance among subdomains is easier to maintain. Once a pair of vertices has been moved, neither is considered for movement in the rest of the pass. The pass ends when all vertices have changed their subdomains. Unlike the GR algorithm, where the refinement process ends if no movement can reduce the edge-cut, the KL algorithm explores moves that temporarily increase the edge-cut (and can thus be considered as a hill-climbing algorithm). This property enhances the capability of the method to escape from local minima. After each pass, the state of the partition at which the minimum edge-cut was achieved must be identified and restored. Experiments show that a small number of passes are normally needed (from 3 to 5) to achieve the best solution.

Since the KL algorithm must evaluate all possible pairs of vertices, every pass has a time complexity of $O(|V|^2 log|V|)$. To reduce runtime, the *Fiduccia-Mattheyses* (FM) *refinement* algorithm [77] is adopted in this work. This scheme differs from KL in that it moves only a single vertex at a time between subdomains, instead of swapping pairs of vertices. The time complexity is thus reduced from $O(|V|^2 log|V|)$ to $O(|E|)$ at the expense of a higher imbalance, which is acceptable in the application domain considered here. To reduce the execution time further, the evaluation is normally restricted to vertices that are in the border of subdomains (i.e., vertices whose incident edges contribute to the edge-cut) [26].

Figure 2.9 and 2.10 illustrate the capability of the FM refinement to escape local minima. For simplicity, a bisection of a planar graph has been considered in the example. The graph in Figure



$|\delta(V_1,V_2)| = 4$
$|V_1| = 8$, $|V_2| = 8$

(a)

$|\delta(V_1,V_2)| = 5$
$|V_1| = 9$, $|V_2| = 7$

(b)

$|\delta(V_1,V_2)| = 6$
$|V_1| = 8$, $|V_2| = 8$

(c)

$|\delta(V_1,V_2)| = 7$
$|V_1| = 9$, $|V_2| = 7$

(d)

$|\delta(V_1,V_2)| = 8$
$|V_1| = 8$, $|V_2| = 8$

(e)

$|\delta(V_1,V_2)| = 7$
$|V_1| = 9$, $|V_2| = 7$

(f)

$|\delta(V_1,V_2)| = 6$
$|V_1| = 8$, $|V_2| = 8$

(g)

$|\delta(V_1,V_2)| = 4$
$|V_1| = 9$, $|V_2| = 7$

(h)

$|\delta(V_1,V_2)| = 2$
$|V_1| = 8$, $|V_2| = 8$

(i)

Figure 2.10: A bisection of a graph refined by Fiduccia-Matheysses algorithm.

2.9 consists of 16 vertices and 24 edges, all of weight 1. Under these assumptions, the weight of a subdomain is the number of vertices in it and the edge-cut is the number of edges crossed by the partition line. Figure 2.9 represents the initial bisection of the graph by a bold line. Since there are eight vertices per subdomain, the initial bisection is perfectly balanced. However, by visual inspection, it is easy to figure out that the bisection of the graph by a vertical line is not optimal in terms of edge-cut. While the vertical line crosses four edges (i.e., edge-cut=4), a horizontal line would have crossed only two (i.e., edge-cut=2). The current bisection can therefore be enhanced by FM refinement if a slight imbalance is allowed between subdomains. In the example, the maximum allowed weight imbalance ratio between subdomains was 9/7. Figure 2.10 (a)-(i) show the results of the different steps in the FM refinement process. On every step, the algorithm computes the gain of assigning each vertex to a different subdomain. Since a bisection of the graph is desired, only one new target subdomain must be evaluated for every vertex. These gain figures are represented beside each vertex. Figure 2.10 (a) represents the initial refinement step. Since all gains are negative, any movement of a vertex will result in an edge-cut increase. Therefore, the optimisation process (or the bisection) is said to be in a local minima. In this situation, a greedy refinement approach would fail to move any vertex, since no vertex leads to an edge-cut decrease. However, the FM algorithm can explore moves that temporarily increase the edge-cut. For this purpose, the algorithm selects the vertex with the highest gain (even when it is negative) and moves it. Amongst the ones with equal gain, the final vertex is selected arbitrarily. The chosen vertex on each step has been highlighted in grey. The resulting bisection is represented in 2.10 (b). The edge-cut has increased from 4 to 5, as a result of moving a vertex with gain -1. This can be interpreted as the "hill-climbing" capability of the local optimisation process. Once moved, the vertex is discarded for the rest of the pass to prevent the vertex from returning back to the old subdomain. The discarded vertices are represented in the figures by a black fill. The algorithm then selects again the vertex with the highest gain. At this point, there is one vertex that has positive gain (i.e., +1), but it has been discarded. Likewise, even though several vertices have the same gain value (i.e., -1), not all of them are valid due to the imbalance constraint. It can be easily checked that the only movement that would not violate this constraint is a vertex that is moved from the left subdomain to the right subdomain. After such a movement, the bisection is balanced again. This weight constraint is the cause of the symmetry of the movements performed in Figure 2.10 (b)-(i). Figure 2.10 (i) represents the final bisection. It is observed that the resulting edge-cut has been reduced from 4 to 2 by means of 8 refinement steps. By visual inspection, it is easy to conclude that this bisection is globally optimal.

It is worth noting that the previous methods have limited capability to escape local-minima. Thus, the quality of the final solution strongly depends on the initial solution. Even though the optimal solution could be found, the number of iterations required would be affected by the selection of the initial solution. Hence, a good initial solution is crucial to improve the efficiency of these methods. The simplest way to achieve an initial solution is to use the solution configured in the network. Since this work deals with the problem of PCU re-assignment, an initial solution is normally available. However, experience with real networks has shown that this strategy based on the refinement of a previous solution is not the most effective one. In contrast, a repartitioning of a graph can also be computed by simply partitioning the graph from scratch. For this reason, refinement algorithms are normally used in combination with other graph partitioning algorithms that do not need any solution to the problem a priori. The remainder of the section is devoted to methods that build the initial partition from scratch.

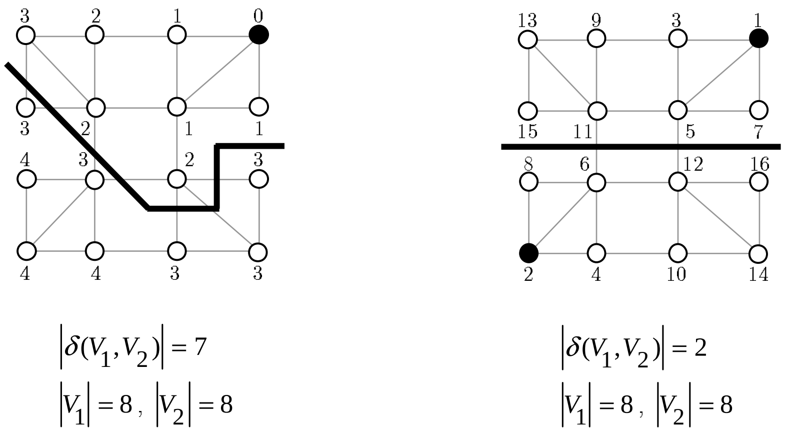
Graph-Walking Algorithms

It is obvious that the edge-cut of a partition is usually minimised if adjacent vertices are in the same subdomain. *Graph-walking* (GK) approaches attempt to put connected vertices together by initially assigning a single vertex to each subdomain and incrementally adding adjacent vertices. The initial vertex is referred to as *seed vertex* and the intermediate state of a subdomain is referred to as a *growing region*. The process of adding vertices to subdomains continues until all vertices in the graph are included in one of the growing regions.

In the *Recursive Level Bisection* (RLB) algorithm [74], a bisection of a graph is built by visiting the vertices of the graph in a breadth-first manner. A subdomain in the graph is grown around a seed vertex, which is assigned the number zero. In subsequent iterations, all vertices that are not assigned a number and are adjacent to any vertex that has been numbered are assigned the latter number plus one. This growing process ends when half of the vertices have been visited (or the subdomain weight is half of the overall weight of the graph). The numbered vertices are assigned to the first subdomain, leaving the remainder in the second one. This technique is extended in *Farhat's greedy* algorithm [78] to consider partitions of several subdomains. In this algorithm, the subdomains are constructed one at a time. For each subdomain, the growing process continues until $|V|/k$ vertices have been assigned a number (or the subdomain weight is $\sum_{i=1}^{|V|} \omega_i/k$). The process is repeated for the remaining subdomains by selecting the boundary vertex with the smallest number of unexplored edges as seed vertex. Both of the previous algorithms share the time complexity of the BFS algorithm, which is $\mathrm{O}(|V| + |E|)$.

In the previous algorithms, no preference is established among the vertices adjacent to a growing region. By contrast, the *Greedy Graph Growing Partitioning* (GGGP) algorithm [26] adds the vertices in an order determined by the potential edge-cut reduction. In its original version, a bisection of the graph is constructed. At each step, vertices out of the growing region are ranked based on the sum of the weight of the edges incident to the vertices in the growing region. The vertex with the highest overall edge weight is then assigned to the growing region following a greedy approach. This technique is also extended in the *k-way Greedy Graph Growing Partitioning* algorithm [38] to consider several growing regions in parallel. In this variant, k (instead of 2) initial vertices are selected and the remaining vertices are alternately assigned to the subdomain with the minimum weight. Thus, the imbalance among subdomains is minimised. The complexity of this algorithm proves to be $\mathrm{O}(|V|^2)$.

Figure 2.11 illustrates the result of different GK methods over the simple planar graph considered so far. Figure 2.11 (a) depicts the result of the RLB algorithm. The index on top of the vertices represents the step in which the vertex was reached (i.e., the hierarchy level in the BFS tree). Thus, the initial node (highlighted in black) is assigned index 0. Adjacent vertices are progressively reached, until half of the vertices have been assigned. From the figure, it can be concluded that the bisection achieved is of average quality (i.e., edge-cut=7). Figure 2.11 (b) depicts the results of the GGGP algorithm. In this case, two seeds are defined (highlighted in black) and the remaining vertices are progressively assigned to the subdomain with the lowest weight. The index on top of each vertex represents the order in which they are included in any of the two growing regions. Thus, the two subdomains in the bisection are represented by odd and even, respectively. From the figure, it is observed that the optimal partition of the graph has been achieved (i.e., edge-cut=2). It should be pointed out that, in the latter method, the convention to select among vertices of similar gains was chosen arbitrarily. Thus, other conventions equally valid might lead to different partitions.



(a) Recursive Level Bisection (b) Greedy Graph Growing Partitioning

Figure 2.11: A bisection of a graph by means of Graph-Walking methods.

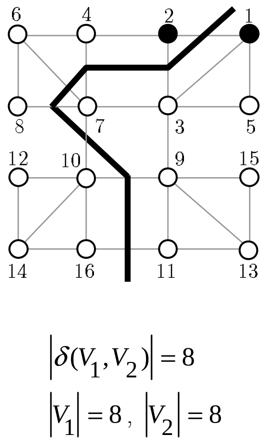


Figure 2.12: A bisection of a graph by means of Greedy Graph Growing Partitioning.

From the previous examples, it is obvious that the performance of GK algorithms depends strongly on the selection of seed vertices. Figure 2.12 presents an extreme case where a bad seed selection in GGGP leads to a bad partition. In this case, the chosen seed vertices (i.e., vertices 1 and 2) are adjacent to each other. Again, the index sequence reflects the order in which vertices are assigned to the two growing regions, represented by odd and even numbers, respectively. From the indices, it is evident that both growing regions come into conflict during the growing process. Due to the alternation in the assignment of vertex to regions, the set of adjacent vertices to the even subdomain gets exhausted after the assignment of vertex 9 to the odd subdomain. This causes that a distant vertex (i.e., 10) has to be assigned to the even subdomain to keep the balance between subdomains. As a result, the final bisection contains a disconnected subdomain, as the even subdomain consists of vertices {2, 4, 6, 8} and {10, 12, 14, 16}. It might be tempting to conclude that RLB and Farhat algorithms do not suffer from this problem, since they build each subdomain at a time. However, they suffer from a similar problem: since every subdomain is grown independent of the others, the set of vertices that remain unassigned does not necessarily have to be connected, but it might consist of isolated vertices.

Two different strategies can be used to choose the seed vertices in the GGGP algorithm. The first one aims to maximise the average distance between seeds, where the distance between a pair of vertices is defined as the number of edges in the shortest path between them in the graph. Thus, the likelihood of assigning adjacent vertices to different subdomains from the very beginning is minimised. For this purpose, the method described in [38] is adopted here. The method starts by selecting an initial vertex at random. Subsequent seed vertices are selected such that they have the greatest average distance to previously selected vertices. As a result, the robustness of GGGP is improved so that acceptable results are obtained even with only one attempt. As main drawback, the minimum distance between each pair of vertices must be computed. For this purpose, the *Floyd-Warshall* (FW) algorithm [69] is commonly used, giving rise to the FW-GGGP algorithm. The FW algorithm computes, for each pair of vertices, the length of all possible paths between the two vertices by multiplying the adjacency matrix multiple times. The main idea behind this algorithm is the construction of a series of matrices $W^{(0)}$, $W^{(1)}$, ..., $W^{(n)}$, where the element w_{ij} of matrix $W^{(k)}$ represents the length of the shortest path between vertices i and j, using only vertices $v_1, v_2, ..., v_k$ as interior vertices in the path. The matrix $W^{(0)}$ is a matrix where $w_{ii}^{(0)} = 0$, $w_{ij}^{(0)} = \gamma_{ij}$ if there exists the edge (i, j) and $w_{ij}^{(0)} = \infty$, otherwise. The core of the process is the construction of $W^{(k)}$ from $W^{(k-1)}$, based on the observation that $w_{ij}^{k} = \min\{w_{ij}^{(k-1)}, w_{ik}^{(k-1)} + w_{kj}^{(k-1)}\}$. This operation must be performed for $k = 1 : |V|, \forall\, i, j = 1 : |V|$. Since the algorithm requires the construction of $|V|$ matrices of size $|V| \cdot |V|$, the complexity of this algorithm is $\Theta(|V|^3)$. Such a complexity is clearly a limiting factor when applied to graphs of reasonable sizes.

In the second strategy, the seed vertices are chosen arbitrarily. To improve the robustness of the method, a limited number of partitioning trials are made with randomly selected seed vertices [26] and the best solution in terms of edge-cut is selected. The effectiveness of this naïve multi-start approach, referred to as *Random-GGGP* (R-GGGP), is limited by the fact that random local optima in large problems tend to all have average quality and little variance. This means that the chance to improve a solution with this approach quickly diminishes from one iteration to the next.

Multi-Level Algorithms

Most of the current graph partitioning algorithms are based on the *Multi-Level* (ML) (or *Hierarchical*) [48] approach. Figure 2.13 shows the main idea behind the ML paradigm. While traditional graph partitioning methods work directly on the original graph, ML techniques first coarsen the graph by collapsing vertices and edges to reduce the size of the graph. The resulting series of graphs $G^{(0)}$, $G^{(1)}$,..., $G^{(m)}$, defines a hierarchy of graphs as successive simplifications of the original graph, $G^{(0)}$. Over these smaller versions, an initial partition is efficiently computed and later uncoarsened to obtain the partition of the original graph. After each uncoarsening step, refinement techniques are applied on small portions of the graph that are close to the partition boundary. Results show that this technique not only reduces the execution time, but also improves dramatically the partition quality.

The ML algorithm consists of three stages: *graph coarsening*, *initial partitioning* and *graph uncoarsening*. The following paragraphs describe the algorithms used on each stage.

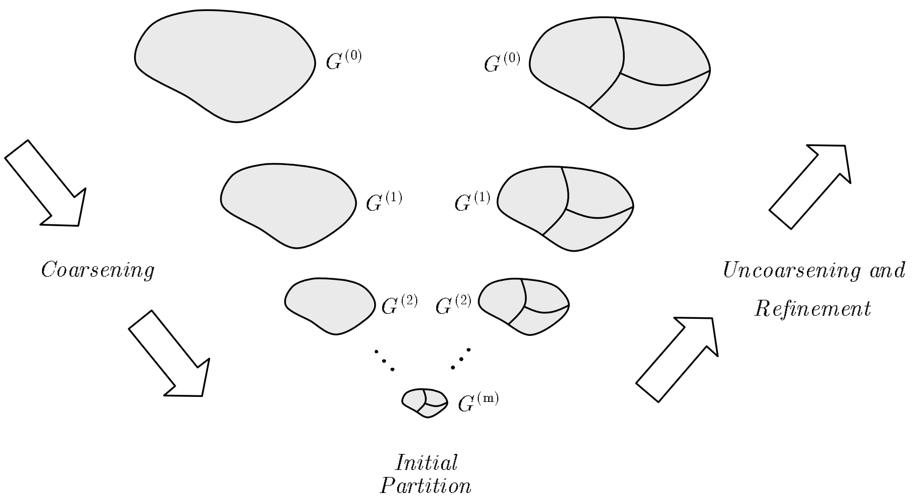


Figure 2.13: The multi-level graph partitioning algorithm.

a) Coarsening

During the first stage, the original graph is progressively coarsened. In most coarsening schemes, sets of vertices (often pairs) are collapsed to form single vertices on the next level coarser graph. The weight of the new vertex is the sum of the weights of the collapsed vertices. To maintain the connectivity information in the coarsened graph, the set of edges incident to the new vertex is the union of the edges incident to the collapsed vertices and their weights are the sum of the associated weights of the underlying edges. Thus, it is ensured that: a) the edge-cut of a partition of a coarser graph is equal to the one in the finer graph; b) a balanced partition in the coarser graph remains so for the finer graph.

The performance of ML techniques strongly depends on the method used to group vertices. The effectiveness of a grouping scheme depends on how successful it is in removing a significant amount of edge weight from the successive coarser graphs. On each coarsening step, the algorithm must define the groups of vertices that will be collapsed. This action is made by defining a subset of edges that are not incident on the same vertex. Thus, it is ensured that a vertex will not belong to several vertices in the coarsened version of the graph. This set of independent edges, M, is referred to as a *matching* (and the algorithm to define it is the *matching algorithm*). The vertices on the ends of edges in M are called *matched* by M and vertices not incident on any edge of M remain *unmatched* by M.

In this work, two alternative representations are used to describe a matching. In the case of matching pairs of vertices, the matching will be denoted by the list of edges that it contains. If more than two vertices can be matched on a single vertex, the matching will be denoted by a list of lists. The outer list represents the vertices in the coarsened graph and the inner lists represent the subset of vertices of the original graph that are grouped in each vertex of the coarsened graph.

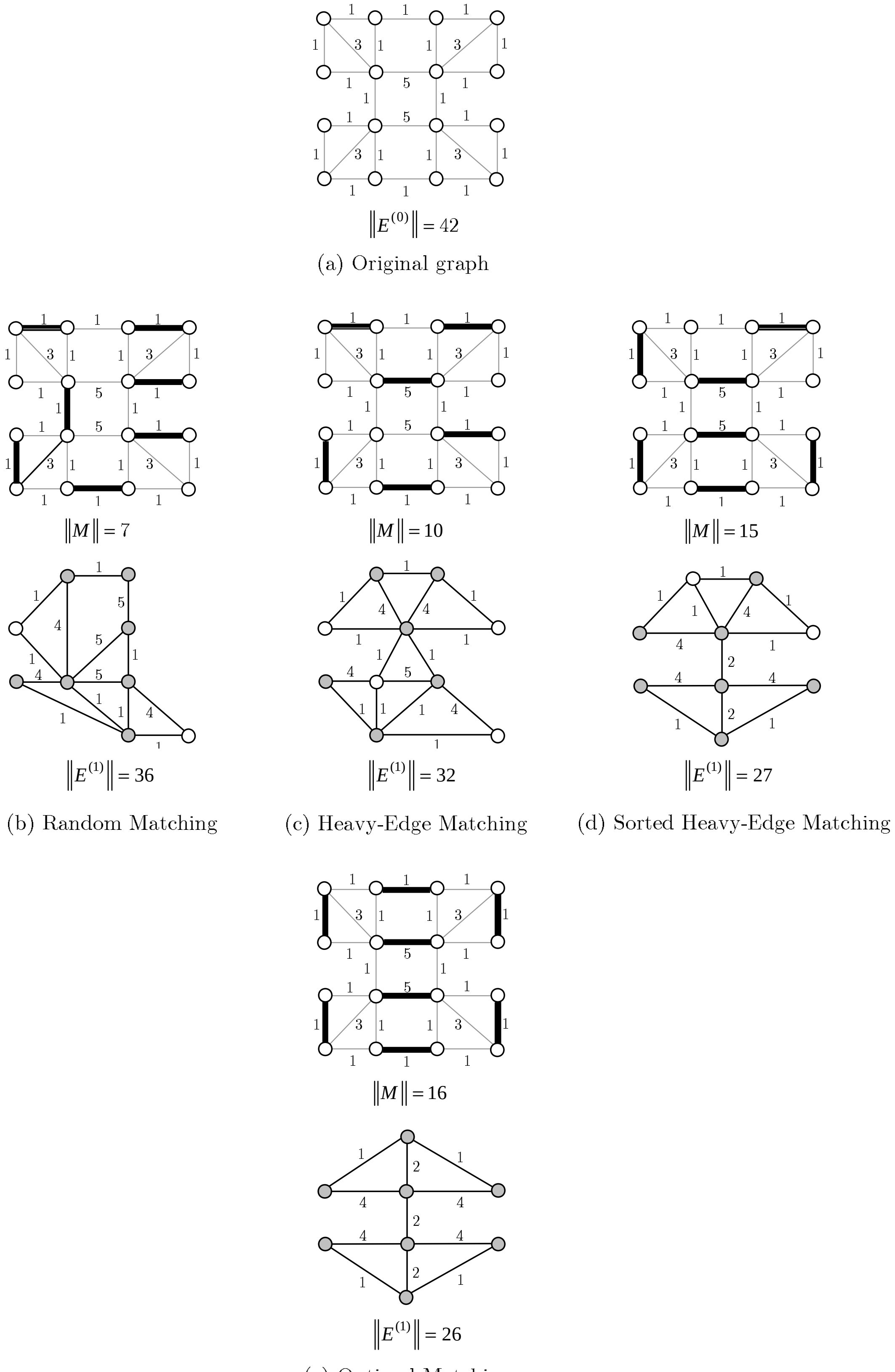
It is obvious that a matching is better as the number of hidden edges is larger. To maximise the number of pairs, the defined matching should be *maximal*, i.e., it should not be possible to include any other edge without making two edges incident on the same vertex. In the common case where vertices are matched by pairs, the number of vertices of the graph $G^{(i+1)}$ is never less than $G^{(i)}/2$. Therefore, at least $\mathrm{O}(\log_2(n/n'))$ coarsening stages are needed to reduce the size of the graph from n to n' vertices. At this point, it is worth noting that the size of a maximal matching is not fixed, but depends on the algorithm used to define it. Depending on graph connectivity and the order in which vertices are paired, some vertices might be left unmatched, thus reducing the effectiveness of the matching algorithm.

In previous work, two matching schemes are commonly used. In *Random Matching* (RM) [26], the vertices are visited in random order. If the vertex has not been matched before, one of its unmatched adjacent vertices is randomly selected. If such an adjacent vertex does not exist, the vertex remains unmatched. This technique is simple and efficient to minimise the number of coarsening levels. However, it does not intend to minimise the edge-cut of the partitions. On the contrary, the final matching only include edges of average weight. By contrast, *Heavy-edge matching* (HEM) [26] also visits vertices in random order, but, instead of selecting a random adjacent vertex to be matched, it selects the adjacent vertex that shares the heaviest incident edge with the original vertex. The matching thus obtained tends to be of a higher weight, since it normally (but not always) includes the heaviest edges in the graph.

In this work, a maximal matching is computed by selecting the heaviest edges on the graph during each coarsening step [79]. By hiding the heaviest edges in a greedy fashion, a larger edge-cut reduction is normally achieved when partitioning the coarser version of the graph. For this purpose, on each version of the graph, edges are sorted by weight and selected in decreasing order. A maximal matching is then obtained by collapsing vertices on their ends if they have not been matched before, until no vertex is left unmatched or all edges have been selected. This matching method is hereafter referred to as *Sorted Heavy Edge Matching* (SHEM). It should be pointed out that this greedy algorithm does not guarantee the maximum overall weight for the matching, since this problem is equivalent to the *vertex-cover problem*, which is known to be $\mathcal{NP}$-complete [69].

Figure 2.14 shows the result of the different matching approaches over a simple graph. As shown in Figure 2.14 (a), the graph considered so far has been modified to include edges of different weights (otherwise, edge selection in HEM and SHEM would be the same as in RM). Figure 2.14 (b)-(d) show the matchings (up) and the coarsened graphs (down) built by the different matching strategies. For comparison purposes, the figure shows the sum of edge-weights in the matching, $||M||$, and in the coarsened graph, $||E^{(1)}||$. It can be easily checked that the larger $||M||$, the lower $||E^{(1)}||$, since $||E^{(1)}|| = ||E^{(0)}|| - ||M||$. From the results, it is clear that SHEM outperforms the other schemes, since it achieves a larger reduction of edge weight in the coarser graph. It can also be verified that the SHEM solution is fairly close to the optimal one, presented in Figure 2.14 (e).

Experiments show that SHEM improves the quality of the matching at the expense of runtime efficiency. The time complexity of the SHEM coarsening algorithm is dominated by that of the edge sorting process over the finest graph. The sorting algorithm used in this work is the *Quicksort* algorithm [69], whose average time complexity is $\mathrm{O}(|E|log|E|)$ in practice. This complexity is not linear in the number of edges, which is the main reason why commercial graph partitioning packages, oriented to very large graphs, adopt other strategies that do not need to order the edges. By randomly selecting vertices, RM and HEM achieve time complexity $\mathrm{O}(|V| + |E|)$. Since CPAP graphs are of limited size and sparse, it is expected that the time to sort edges is not excessively high. The availability of very efficient implementations of this sorting algorithm helps that the latter assumption holds.



(a) Original graph

(b) Random Matching

(c) Heavy-Edge Matching

(d) Sorted Heavy-Edge Matching

(e) Optimal Matching

Figure 2.14: Comparison of different matching heuristics.

b) Initial partitioning

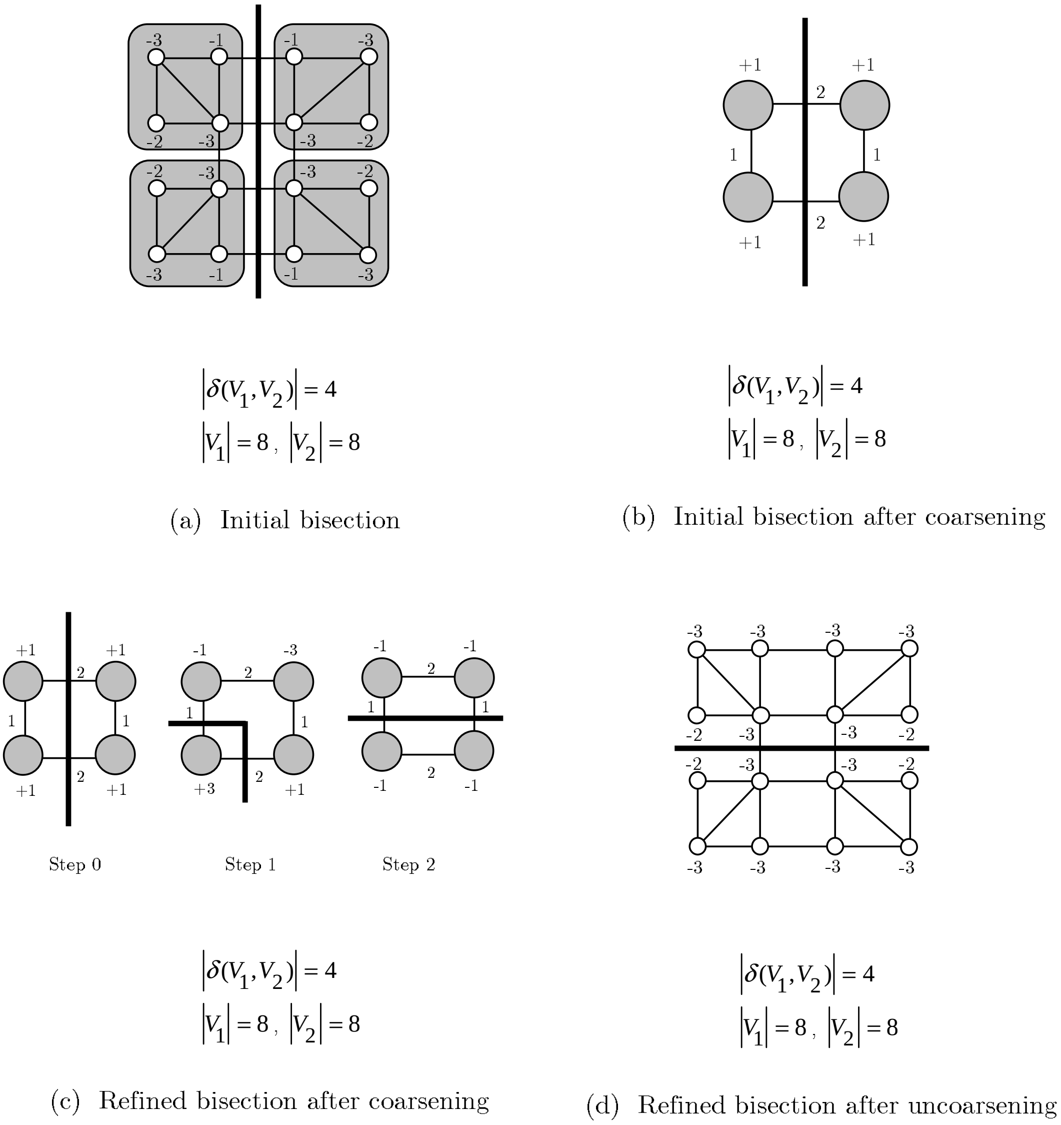
The second stage of the ML algorithm computes an initial partition of the coarsest graph. In the simplest strategy, the coarsening process continues until the number of vertices in the coarsest graph is the same as the number of subdomains, k [79]. Thus, the partition is built by assigning each vertex in the coarsest graph, v_i, to subdomain V_i ($i = 1 : k$). This method is considered in this work as the *standard* ML algorithm.

The previous strategy requires some mechanism to ensure that the final partition is balanced, as vertices of the coarsest graph are generally not homogeneous. Likewise, experiments show that the effectiveness of most matching algorithms decreases after a few coarsening stages. To cope with these problems, the coarsening stage might end when the number of vertices in the coarsest graph is below a certain threshold (e.g., c times k). Over the coarsest graph, an initial partition can be constructed with any of the previous graph partitioning methods (usually the R-GGGP algorithm). It is worth noting that the complexity of the latter algorithm is not a critical issue, since it is applied over the coarsest version of the graph. This fact leaves the door open to the use of more sophisticated algorithms to build the initial partition, such as the one discussed next. This variant in which the coarsening process ends prematurely will be referred to as *early-stop* ML algorithm.

c) Uncoarsening/Refinement

During the final stage, the partition is projected onto the original graph. This process is carried out by unfolding the vertices based on the matching information gathered during the coarsening stage. The unfolding algorithm is trivial: if a vertex v is in subdomain V_i in the initial (i.e., coarsest) partition, the matched vertices that the former represents are also assigned to V_i. As the graph gets finer, the additional degrees of freedom can be used to improve the partition. In this work, the FM refinement algorithm is applied after each coarsening step. This approach is adopted, regardless of the limited benefit in terms of edge-cut when compared to the situation where the refinement is performed only after the last uncoarsening operation. The reason for this option is that, in the coarsest graphs, the movement of a single vertex is equivalent to the movement of a large number of vertices. Less vertex movements are therefore needed to reach the final solution, which counterbalances the larger number of times the refinement algorithm is executed.

The discussion so far has stressed the reduction of computational load that is achieved by working on simplified versions of the graph. However, nothing has been said about the improvement in solution quality achieved by ML techniques. By reducing the exposed edge weight, the task of computing a good quality partition becomes easier. Equally important, the matching process smoothes the optimisation surface, reducing the number of local minima [80]. Thus, trajectory-based optimisation methods, such as the GGGP, KL and FM algorithms, increase their capability to escape from local minima. Figure 2.15 illustrates this property of ML techniques over the bisection problem considered so far. Figure 2.15 (a) shows the initial bisection of the graph in Figure 2.9. The gain of assigning each vertex to a different subdomain is again represented beside each vertex. As shown previously, this bisection is in a local minima, since there is no movement of a single vertex that would enhance the edge-cut (i.e., with a positive gain). The shaded area represents vertices that are matched during the coarsening stage. After coarsening, the graph consists of four vertices, each one comprising four vertices of



(a) Initial bisection

(b) Initial bisection after coarsening

(c) Refined bisection after coarsening

(d) Refined bisection after uncoarsening

Figure 2.15: The refinement of a bisection in a local minima after coarsening.

the original graph, as depicted in Figure 2.15 (b). In this graph, the movement of any vertex is equal to the movement of the whole set of vertices represented by that vertex. The new gain values are all positive, which highlights that any re-assignment would lead to an edge-cut decrease. Consequently, the bisection after coarsening is not in a local minima. A simple greedy refinement would suffice to reach the optimum bisection of the problem, as shown in Figure 2.15 (c). The later uncoarsening stage unfolds the matching to get the final bisection with full resolution, which is presented in Figure 2.15 (d). It is worth noting that, in this particular case, a larger imbalance between subdomains has to be allowed during the refinement stage. Since the coarsened graph consists of a small number of vertices of large weight, any vertex movement leads to a large imbalance between subdomains. This problem is a common drawback of ML approaches under strict balancing constraints among subdomains. To cope with this problem, several authors (e.g., [49]) have proved that it is beneficial to relax the balancing constraints for the refinement of the coarsest levels and progressively strengthen them as the uncoarsening stage progresses. Experiments showed that this approach gave negligible improvement over CPAP graphs and it has thus been discarded. From the previous example, it can be concluded that

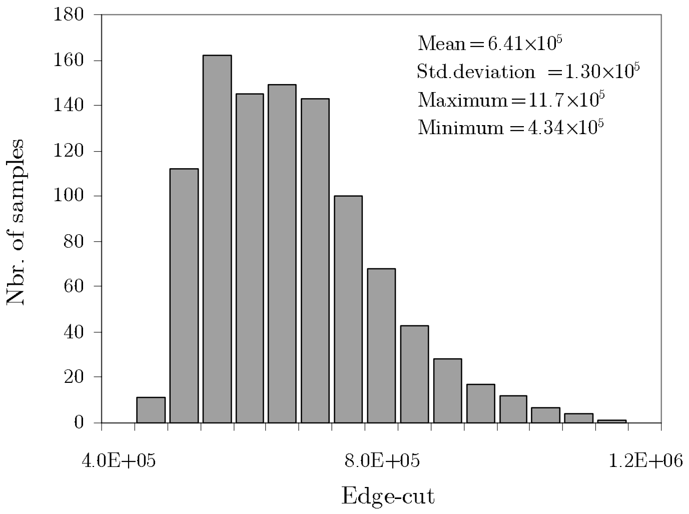


Figure 2.16: Histogram of the values of local minima from R-GGGP algorithm.

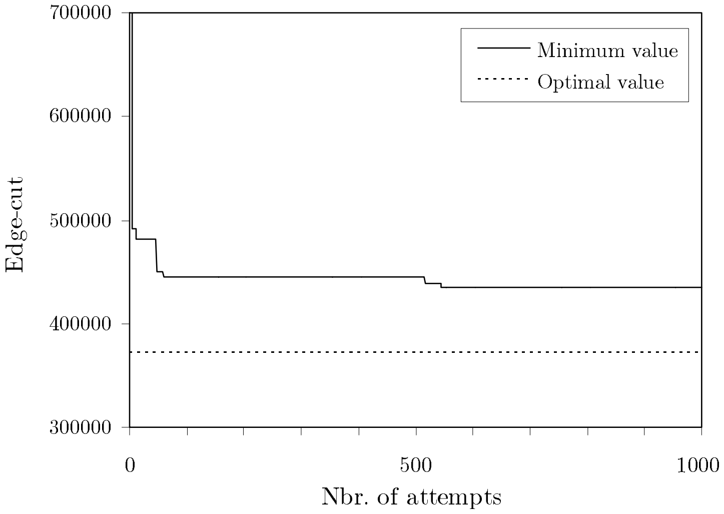
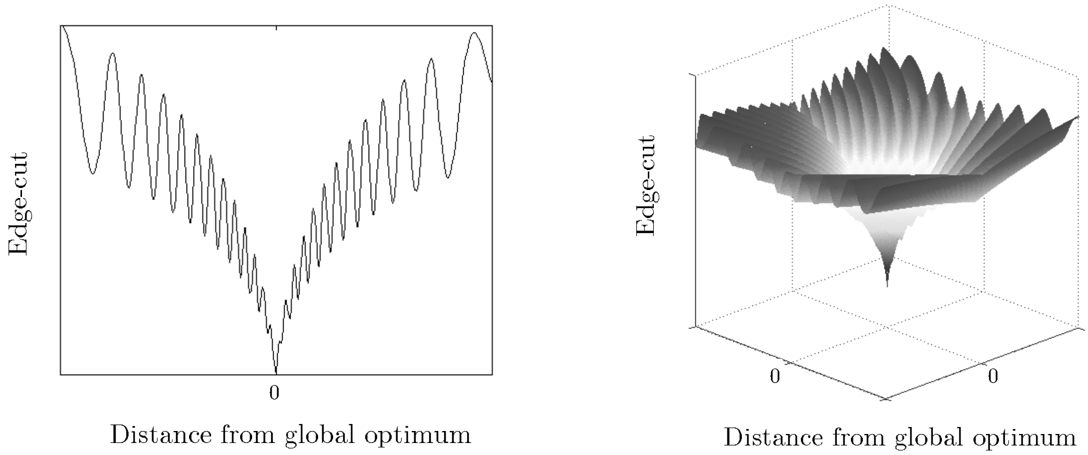


Figure 2.17: Evolution of minimum edge-cut value across attempts.

the optimisation surface is smoothed as a result of the coarsening process. This effect proves beneficial for techniques that take advantage of the regularities of the optimisation surface, which are discussed next.

Adaptive Multi-Start Algorithms

The above-described algorithms build an initial partition that is later refined, and can thus be classified as local optimisation methods. Consequently, the quality of the final solution is heavily influenced by that of the initial solution. It is then clear that the performance of these methods can be greatly improved by performing several attempts. Such an approach is referred to as *multi-start* (MS) technique [81]. By sampling the solution space, the diversity of the search is improved at the expense of a loss of intensity in the search. In its simplest version [82], the initial solutions are randomly selected, which, in the graph partitioning case, can be easily implemented by selecting seeds randomly in GGGP.



(a) 1-D

(b) 2-D

Figure 2.18: A globally-convex optimisation surface.

Unfortunately, the effectiveness of this strategy is limited, as random local minima in large combinatorial optimisation problems tend to all have intermediate quality with very little difference between them [52]. For instance, Figure 2.16 presents the histogram of the distribution of edge-cut values of 1000 different solutions achieved by the R-GGGP algorithm with greedy refinement over a real instance of the CPAP. Some relevant statistics have also been superimposed on the figure. It is observed that most local minima show values close to the average value (i.e., $6.41{\cdot}10^5$), which is significantly worse than the best value of the sample (i.e., $4.34{\cdot}10^5$) and the optimal value (i.e., $3.72{\cdot}10^5$). This result implies that the probability to improve a previous solution quickly diminishes from one attempt to the next. Therefore, a large number of attempts are needed to get a near-optimal solution (i.e., to hit the tail of the distribution). This behaviour is clearly evidenced in Figure 2.17. This figure represents the edge-cut of the best local minima found by R-GGGP as the number of attempts increases. In the figure, it is observed that the solution quality quickly improves in the first attempts, but no further improvement is achieved after 600 attempts, even if the quality is far from the optimal value.

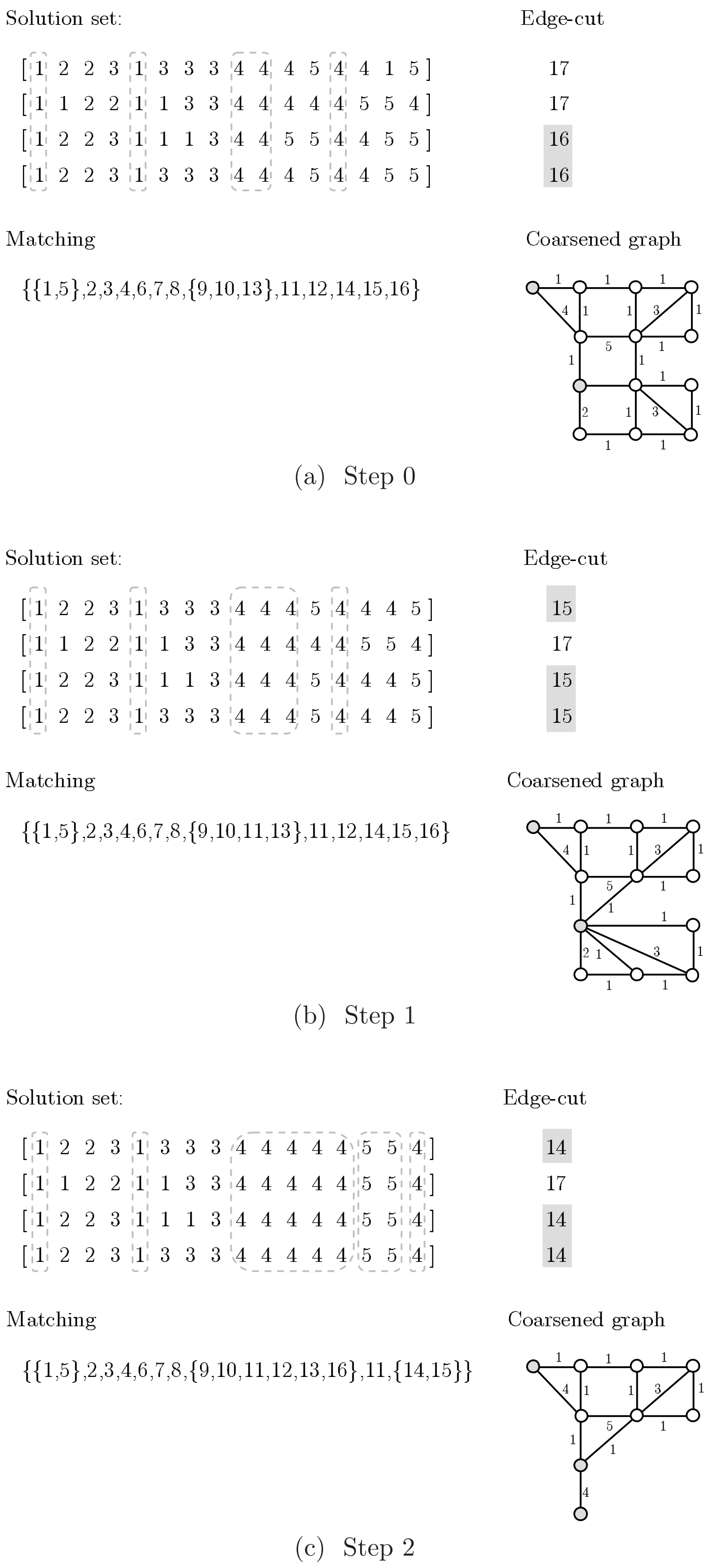
The shortcomings of the previous approaches can be overcome by taking advantage of the regularity of the search space. In large combinatorial problems, the optimisation surface displays a structure where the best local minima are grouped in a central position in the search space [52]. Such a surface with a "big central valley", as the ones depicted in Figure 2.18 (a)-(b), is said to be *globally-convex*. In both figures, it is observed that the optimisation surface comprise multiple local minima, and hence the shortcomings of local search methods. However, the best local minima are close to each other, which can be used to improve the efficiency of the search.

The previous observation is the basis of the *Adaptive Multi-Start* (AMS) techniques. These methods exploit the regularity of the optimisation surface to direct the search to regions where it is more likely that the best solution is found. This can be easily achieved by selecting starting points for local search methods from previously found local minima. Thus, the efficiency of the search is greatly improved, provided that regularities are correctly detected. In the case of GPP, the regularity of the optimisation surface is translated into similarities between the best solutions. Thus, the best partitions of a graph often share the same assignment of a large number of vertices, i.e., the distance amongst the best solutions is small.

Amongst all AMS methods for the GPP, the *Clustered Adaptive Multi-Start* (CAMS) algorithm [28] is the most widely used. In this method, new partitions are generated from the best partitions previously computed. The core of the method is the detection of similarities among previous solutions, which is used to simplify the graph. Being a ML method itself, this method performs coarsening based not only on the graph structure, as traditional coarsening schemes do, but also on the similarities of previous solutions. Thus, coarsening is not performed statically, but dynamically, based on the solutions built so far.

The CAMS algorithm starts with the construction of a limited set of random solutions by means of the naive MS method (i.e., R-GGGP). By dealing with several initial solutions, the diversity of the search is ensured. On each subsequent iteration, the algorithm identifies clusters of vertices in the same subdomain in all solutions. Once these groups of vertices are identified, a simpler version of the graph is constructed by collapsing all vertices in a cluster into a single vertex. This matching operation is also performed over the existing set of solutions. A refinement algorithm is then applied over all existing solutions in the simplified version of graph. It is worth noting that, even though the matched solutions are essentially the same as their non-matched counterparts, the application of the refinement algorithm over a simpler graph does not necessarily lead to the same solutions. On the contrary, by collapsing new groups of vertices, the capability of local-search methods to escape local minima is improved. As a result, a new set of solutions is obtained, over which the previous steps are repeated. As iterations pass, the graph is progressively simplified by detecting new similarities. This iterative process ends when all solutions coincide or the refinement process does not produce any change.

Figure 2.19 shows an example of partitioning performed by CAMS. In the example, the graph in Figure 2.14 must be divided into five subdomains, provided that the weight imbalance ratio is less than three. Figure 2.19 (a)-(c) present the results of the different iterations of the method. Figure 2.19 (a) shows the result of the first step (step 0). The algorithm starts by computing four initial solutions by repeated use of R-GGGP. These solutions correspond to different local minima in the optimisation surface. The set of solutions is described by showing the assignment of vertices to subdomains. Thus, each column corresponds to one of the sixteen vertices in the graph and each row represents one partition. Among the resulting partitions, two of them show the minimum edge-cut value of 16 (highlighted by a shaded box). In this set of solutions, the algorithm identifies clusters of vertices that are grouped under the same subdomain in all solutions. Concretely, two clusters have been identified in this case: one in subdomain 1 and another in subdomain 4 (inside dashed lines). It is worth noting that this process is not concerned with the assignment of a single vertex to the same subdomain in all solutions (e.g., 3rd vertex), but with entire clusters of vertices sharing the same subdomain in all solutions (e.g., {1,5} and {9,10,13}). By matching these clusters of vertices, a simplified version of the graph is obtained. Thus, part of the edge weight in the graph is hidden, as in any classical coarsening algorithm. Step 0 ends with the refinement of the solutions over the coarsened graph, which leads to the new set of solutions presented in Figure 2.19 (b). It is observed that, although some solutions do not change after the refinement process (e.g., 2nd solution), most of them are improved as their edge-cut is decreased (e.g., 1st, 2nd and 3rd solution). Likewise, vertex 11 is now assigned consistently in all solutions, and is thus added to the cluster already identified in subdomain 4. After refinement in step 1, the new solution set, presented in Figure 2.19 (c), has vertices {12,15,16} assigned consistently in all solutions, which is used to simplify the graph further. Finally, the refinement in step 2 results in no change in the solution set and the algorithm stops, even though the graph has not been fully coarsened (i.e., the number of vertices is not exactly the number of subdomains).



Solution set: Edge-cut

[1 2 2 3 1 3 3 3 4 4 4 5 4 4 1 5] 17
[1 1 2 2 1 1 3 3 4 4 4 4 4 5 5 4] 17
[1 2 2 3 1 1 1 3 4 4 5 5 4 4 5 5] 16
[1 2 2 3 1 3 3 3 4 4 4 5 4 4 5 5] 16

Matching

{{1,5},2,3,4,6,7,8,{9,10,13},11,12,14,15,16}

Coarsened graph

(a) Step 0

Solution set: Edge-cut

[1 2 2 3 1 3 3 3 4 4 4 5 4 4 4 5] 15
[1 1 2 2 1 1 3 3 4 4 4 4 4 5 5 4] 17
[1 2 2 3 1 1 1 3 4 4 4 5 4 4 4 5] 15
[1 2 2 3 1 3 3 3 4 4 4 5 4 4 4 5] 15

Matching

{{1,5},2,3,4,6,7,8,{9,10,11,13},11,12,14,15,16}

Coarsened graph

(b) Step 1

Solution set: Edge-cut

[1 2 2 3 1 3 3 3 4 4 4 4 4 5 5 4] 14
[1 1 2 2 1 1 3 3 4 4 4 4 4 5 5 4] 17
[1 2 2 3 1 1 1 3 4 4 4 4 4 5 5 4] 14
[1 2 2 3 1 3 3 3 4 4 4 4 4 5 5 4] 14

Matching

{{1,5},2,3,4,6,7,8,{9,10,11,12,13,16},11,{14,15}}

Coarsened graph

(c) Step 2

Figure 2.19: Example of partition by the Clustered Adaptive Multi-Start algorithm.

It is worth noting that matched vertices remain unchanged by the refinement process, since they are treated as a single vertex. Consequently, the refinement of an old set of solutions leads to a new set of solutions that, at least, share the same degree of similarity as the previous set. As a result, the similarities between solutions increase with iterations and the graph is progressively simplified, extending the capability of the ML technique to escape from local minima. This trend is clearly observed in step 2, where 2 out of 4 solutions are exactly the same (i.e., 1st and 4th solution), and the remaining ones only differ in 2 vertices (i.e., vertices {2,6} and {6,7}, respectively).

The main parameter of the previous algorithm is the number of initial solutions, g. This parameter controls the trade-off between diversity and intensity of the search. A large g ensures that most regions of the solution space are explored, increasing the diversity of the search. The main drawback is a reduced likelihood of a group of vertices being in the same subdomain for all solutions. As a result, a large number of iterations is needed to ensure that all solutions finally coincide, which might be considered as a loss of intensity in the search. Alternatively, a small g ensures a fast convergence to the final solution at the expense of a higher variability of solution quality. Experiments have shown that a value of g=5 provides a good trade-off between diversity and intensity in the search.

It is worth noting that CAMS has important similarities with evolutive approaches [83]. Similar to the latter, there exists a population of solutions that is improved across generations. On each generation, an offspring is produced from parent solutions. Unlike most evolutive methods, where every new individual is generated from two parent solutions, CAMS combines several solutions to generate new better ones. This combination process leads to a progressive loss of diversity, since all solutions converge to a unique solution. However, it is worth remarking that the refinement process can lead to new solutions after each new coarsening operation, and some diversity is thus gained. This trade-off between intensity and diversity is the basis of the excellent performance of CAMS in terms of solution quality and runtime efficiency.

Adaptation to the Cellular Environment

The previous discussion has dealt with classical graph partitioning algorithms, which were mainly conceived to solve the GPP in supercomputing applications. In contrast, the CPAP (and, in general, any partitioning problem in the cellular environment) has several peculiarities that must be taken into account. These differences affect both formulation and solution techniques. The following paragraphs discuss the limitations of classical graph partitioning algorithms to solve the CPAP. The solution techniques described here will all be used in the proposed algorithm described later.

1) Imbalance between subdomains

Most graph partitioning algorithms in supercomputing aim for a perfect balance among subdomains, since the performance of the final solution heavily relies on the fulfilment of this constraint. This restriction decreases the number of candidate solutions, thus reducing the degrees of freedom in the optimisation process. In the CPAP, some imbalance between PCU loads is permitted, provided that the number of inter-PCU CRSs can thus be reduced.

Even if some algorithms allow for the control of the maximum imbalance between subdomains, the definition of this indicator differs from the way it is currently understood by cellular network operators. This difference does have an influence on the mechanism that controls the imbalance among subdomains. Conventional graph partitioning algorithms define the imbalance as the ratio between the weight of the largest subdomain and the weight of a subdomain under perfect balance. By contrast, operators are more interested in the ratio between the maximum and minimum subdomain weights (i.e., the weight imbalance ratio). Although it might seem that both definitions provide similar results, it is worth noting that the former definition only entails the control of the largest subdomain in the partition, whilst the latter also requires the control of the smallest subdomain. As a consequence, conventional algorithms accept partitions with almost empty subdomains, as long as the weight of the largest subdomain is below a certain percentage of the overall weight of the graph. Such a solution has the same problems as the solutions manually configured by the operator. Therefore, unlike conventional approaches, the weight imbalance ratio between the largest and the smallest subdomain must be controlled explicitly.

In addition, experiments have shown that, although classical algorithms usually find partitions of very good quality with a small weight imbalance ratio between subdomains, a large imbalance is observed in some cases. This behaviour is more frequent in graphs with a small number of vertices with large weight (as it is the case for the CPAP). To avoid this situation, the refinement algorithm proposed in this work first tries to build a valid partition, which is later refined. This is the reason why, in some isolated cases, the refinement process might lead to an edge-cut impairment in order to ensure the balance among subdomains.

2) Connectivity

As already stated, operators prefer solutions where cells in the same PCU are geographically close to each other. Although this property is not strictly required, it makes checking of the PCU plans on a map easier. As no distance information is included in CPAP graphs, distance must be inferred from connectivity information in the adjacency matrix. Thus, the only way to check that two clusters of vertices in the same subdomain are geographical neighbours is by checking that there exists an edge (or path) that links them. If such an edge does not exist, clusters can be arbitrarily far from each other. It can thus be concluded that, in order to keep geographical consistency, it is necessary that all vertices in a subdomain are connected (i.e., there exists an internal path between every pair of vertices in the subdomain). This constraint is hereafter referred to as the *connectedness property*.

The connectivity of vertices in a subdomain is hardly ever considered in supercomputing, as this is not a critical issue. Although some post-processing is performed by most algorithms to avoid disconnected subdomains, experiments show that disconnected subdomains are still present in the final partition. The origin of this problem is the lack of explicit connectedness checks during the refinement stage. Any re-assignment of a vertex to a different subdomain might occasionally break the connection of vertices that are left in the source subdomain. To prevent such an event, the refinement algorithm in this work checks the fulfilment of the connectedness property after any potential movement. This variant is known in the literature as *connected refinement* [38].

The connected refinement algorithm starts with the identification of subdomains that are initially connected. To maintain connectivity, only vertices in the borders of subdomains should

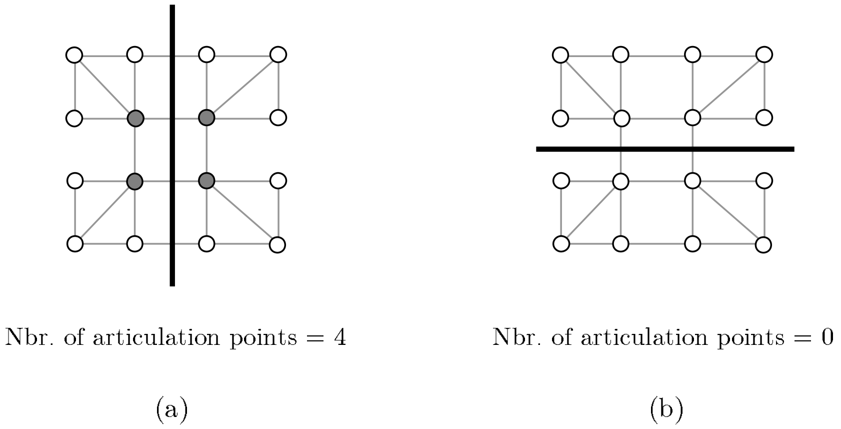

Nbr. of articulation points = 4
Nbr. of articulation points = 0
(a)
(b)

Figure 2.20: Articulation points in a graph.

be moved to another subdomain. For that purpose, articulation vertices in connected subdomains must first be identified. A vertex u is defined as an *articulation point* of the subdomain V_n to which it belongs, if there exists a pair of vertices $v, w \in V_n$, such that $u \neq v \neq w$ and u is included in every path between v and w [38]. From this definition, it is easily deduced that if u is moved to a different subdomain, there is no path between v and w inside V_n and V_n becomes disconnected. To avoid such an event, articulation points cannot change subdomain and must be excluded from the candidate vertex list during the refinement process. Thus, a vertex can be moved to a different subdomain only if it: (a) does not destroy the connectedness of the subdomain it leaves, (b) does not lead to violation of weight constraints, and (c) does not empty the partition it leaves.

For instance, Figure 2.20 shows the articulation points of two different bisections of the same graph. The bisection of Figure 2.20 (a) has two articulation points per subdomain (highlighted in grey). If one of these vertices is moved to the other subdomain, the upper cluster of vertices becomes disconnected from the lower cluster. In contrast, no articulation point is observed in the subdomains of Figure 2.20 (b). This example proves that articulation points do not only depend on the graph structure, but also on the current partition of the graph. Consequently, the set of articulation points is not fixed, but changes during the refinement process, and it must be updated after every vertex move. It is clear that any re-assignment of a vertex only affects the source and target subdomains. Nonetheless, the management of articulation points proves to take a significant part of the computation, so it must be carefully designed.

Checking that a subdomain is connected and identifying its articulation points can be performed by the same algorithm. This commonality is obvious, as the simplest method to determine if a vertex is an articulation point of a connected subdomain is to check the connectedness of the subdomain without the considered vertex. The connectedness of a subdomain can be checked by traversing the subdomain following a DFS strategy. This algorithm starts with the definition of an initial random vertex as a root. On each subsequent step, a new adjacent vertex is chosen. The search explores as far as possible along each branch and a new branch is chosen only when all vertices in a branch have been visited. The group of vertices that are reached by this search is called a *connected component*. If all vertices in the subdomain belong to the same connected component, the subdomain is connected; otherwise, it is disconnected. The time complexity of this algorithm is that of the DFS algorithm (i.e., $\Theta(|V| + |E|)$).

The previous algorithm is appropriate for the identification of connected subdomains and articulation points at the beginning of the refinement process. However, this method proves extremely inefficient to keep the set of articulation points updated after every vertex movement, since a full DFS has to be performed as many times as there are vertices in the source and target subdomains. By defining some basic rules, the number of checks can be kept to a minimum and the efficiency of the DFS can be increased. In the target subdomain, the addition of a vertex can only affect the set of articulation points in two ways: (a) causing that its adjacent vertices become articulation points, and (b) the old articulation points are not so anymore. Thus, only these vertices should be checked for being articulation points in the target subdomain. By contrast, these checks must still cover all vertices in the source subdomain. At the same time, the computational load of these checks can be minimised by reducing the depth of the DFS. To check if a vertex is an articulation vertex, only its adjacent vertices (and not every vertex) in the subdomain must be checked for connectedness. If the original vertex is not an articulation vertex, such a reduced set of adjacent vertices is normally reached in the early steps of the DFS, so that the search can be interrupted. Only in the case of an articulation vertex, a full DFS must be performed.

It should be pointed out that it is sometimes impossible to satisfy the connectedness constraint if the original graph consists of two or more isolated clusters of vertices, and hence is not itself connected. In a live network, this situation is likely to happen in rural areas, where discontinuous cell coverage is common. Under these circumstances, the best solution is the one that has the minimum number of disconnected subdomains and the connected refinement algorithm should strive to achieve this solution. A special case of the latter situation is the existence of isolated cells in the network (i.e., cells with no incoming or outgoing HOs). These cells are represented in the graph by vertices without edges. This event is not considered in previous connected refinement algorithms reported in the literature. On the contrary, it is always assumed that the original graph is connected, even if subdomains built by a partition might not be so. However, this situation is rather common in live networks, where coverage is not always seamless. Although these isolated vertices do not contribute to the overall edge-cut, they do have an influence on the weight and connectivity of subdomains to which they are assigned. By definition, a subdomain with an isolated vertex would strictly be considered as disconnected, since all its vertices are not in the same connected component. Hence, no articulation points would be defined for such a subdomain in the refinement process. As vertices in this subdomain could be re-assigned freely, the subdomain would tend to disaggregate into isolated clusters of vertices due to the elimination of vertices that interlink them. To prevent this situation, the proposed refinement algorithm extracts isolated vertices before checking the connectivity of a subdomain. After this simple step, it is observed that most subdomains would be connected, were it not for these isolated vertices. Thus, the articulation points of these modified subdomains can be identified, preventing disaggregation during the refinement process.

3) Granularity of the assignment

In principle, the smallest network entity that can be assigned to a PCU is a cell. Thus, the original graph that models the CPAP is said to have *cell resolution* (i.e., every vertex stands for a cell). Although not common, it might happen that the methods discussed so far assigned cells in the same site (e.g., sectors of a tri-sectorised cell) to different PCUs. To avoid this situation, cells in a site can be forced to be in the same PCU, which is referred to as *site constraint*. There are several reasons to justify this choice. On the one hand, solutions with only one PCU per site are easier to check on a map by maintenance personnel. Equally important, these solutions can

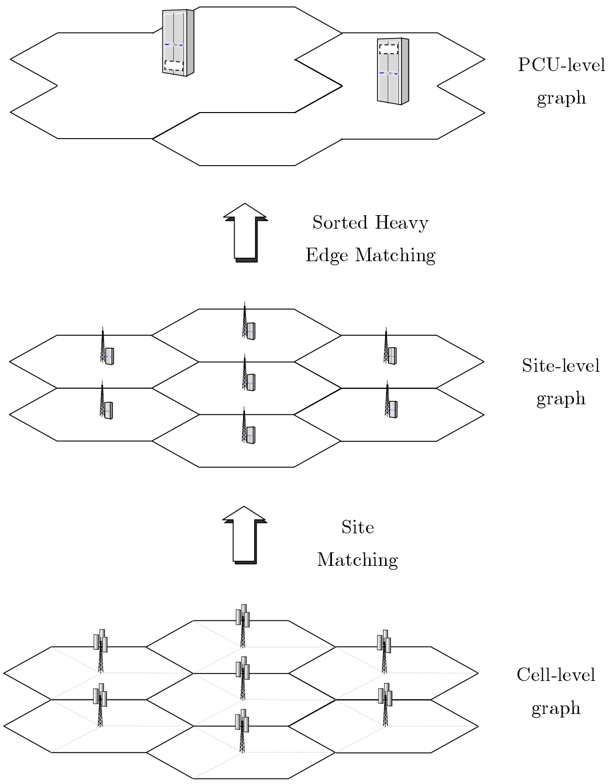


Figure 2.21: The hybrid graph partitioning method.

be viewed as a hybrid approach that combines both mobility (i.e., edge-cut) and geographical (i.e., site grouping) criteria. These solutions are expected be more robust against changes in user mobility trends and propagation scenario at the expense of an increased edge-cut.

In essence, this hybrid approach solves the CPAP in a BSC over a simplified version of the graph whose vertices represent the sites of the BSC. Since the partitioning process deals with sites (and not cells), co-sited cells are assigned as a whole to the PCU and such a solution is said to have *site resolution*. Formally, this approach can be viewed as an ML technique with a new coarsening algorithm in the first (and maybe the last) coarsening step. In this coarsening algorithm, referred to as *Site Matching* (SM), vertices of the original graph corresponding to co-sited cells are matched into a single vertex, which represents the whole site. Over this graph of site resolution, any of the partitioning algorithms described so far can be applied. In case of the classical ML algorithm, the graph would be simplified further by the standard SHEM coarsening algorithm. The combination of both coarsening algorithms is hereafter referred to as SM-SHEM ML algorithm. Figure 2.21 shows the main idea behind the SM-SHEM ML approach. As in other ML methods, the graph is progressively simplified to reduce the computational load and escape from local minima. The only difference lies on the first coarsening step, where the matching is defined based on geographical data (i.e., site information in the NMS) and not on the graph structure (i.e., HO statistics). Once cells on each site have been matched, the method proceeds as in the standard ML algorithm. After several coarsening-uncoarsening stages, a site-level solution is obtained. Finally, the last uncoarsening step (i.e., from site-level to cell-level resolution) does not include any refinement to ensure that cells in a site remain in the same PCU.

From the runtime perspective, SM is faster than SHEM. On the one hand, the matching is pre-defined by the network infrastructure and does not have to be computed. On the other hand, the size of the graph obtained by SM is smaller than with SHEM, as sites normally comprise more than two cells. However, there is no guarantee that this matching is optimal (or, at least, better than SHEM) from the edge-cut perspective. On the contrary, experiments show that matching vertices based on geographical information produces solutions with higher edge-cut than solutions based on structural information. The restriction of the degrees of freedom in the assignment unavoidably leads to solutions of worse quality. Nonetheless, it is still interesting to check if these simpler solutions can perform competitively with more refined solutions.

It should be pointed out that there exist several definitions of the term *site*. For simplicity, this work deals with the logical identifier of the site in the BSC, in contrast to the geographical site. In practice, this logical identifier manages to group sectors of a tri-sectorised cell. Unfortunately, depending on network configuration, this strategy might fail to group co-sited cells of different frequency bands located in the same geographical position. This issue could be easily corrected by considering the geographical data currently available in the NMS.

4) Number of changes in the network

As stated previously, it is beneficial from the performance point of view to build the solution to the CPAP from scratch, without considering the solution currently implemented by the operator. This is expected since it is better to use a state-of-the-art graph partitioning algorithm to compute the new partition than trying to refine a poor solution. In the context of re-planning processes, the solution thus obtained might lead to a large number of changes in the network, since some cells might need to be reassigned to other PCUs. Although these changes can currently be performed automatically, there are a number of reasons to reduce the number of changes in the network. Apart from the ease of management, the number of cells that are re-assigned to a different PCU must be minimised to increase service availability, since any PCU re-assignment might require temporary disabling of packet-data services in the cell.

To reduce the number of changes in the network, the *scratch-and-remap* (SR) technique [23] is adopted here. The method first builds the partition of a graph from scratch without considering the original solution. This intermediate solution is remapped to a new solution by changing the labels of the subdomains so as to minimise the differences between the old and new solution. By simply changing subdomain labels of the new partition in accordance with the old partition (without modifying the partition structure), the number of changes can be significantly reduced. The core of the mapping process is the comparison of the subdomains of the old and intermediate solution. For this purpose, the method builds a *similarity matrix*, S, of size $k{\cdot}k$, where k is the number of subdomains, whose rows and columns correspond to the subdomains of the old and new partitions, respectively. Each element in S, s_{qr}, represents the sum of the weight of the vertices that are in subdomain q of the old partition and in subdomain r of the intermediate partition. Every subdomain in the intermediate partition is re-labelled to the subdomain in the old partition with which it has the largest similarity. Thus, the mapping process is a bijective function that defines a one-to-one correspondence between subdomains of both solutions. The mapping is defined by selecting k elements in S, such that every row and column contains exactly one selected element and the sum of the selected elements is maximised. This choice corresponds to the mapping that maximises the amount of overlap between the original and the remapped partition. It is worth noting that the mapping process

Old partition: [1 1 1 2 2 3 3 2]

Intermediate partition: [1 2 2 3 3 3 3 3]

Similarity matrix:

$$\begin{array}{c|c|c|c|} & 1 & 2 & 3 \\ \hline 1 & 1 & 2 & 0 \\ \hline 2 & 0 & 0 & 3 \\ \hline 3 & 0 & 0 & 2 \\ \hline \end{array}$$

Mapping: 1→3 , 2→1 , 3→2

Re-labelled partition: [**3** 1 1 2 2 **2** **2** 2]

Nbr. of changes: 3

Figure 2.22: Example of re-labelling by means of a similarity matrix.

considers the sum of vertex weight affected by changes, instead of the number of vertices. The application of this criterion minimises the number of users (and not the number of cells) that might be affected by a cell disabling due to a PCU re-allocation.

Figure 2.22 illustrates an example of re-labelling. The inputs to the algorithm are the old and intermediate partitions of an unweighted graph with 8 vertices and 3 subdomains. As shown in the figure, both partitions differ in 5 out of 8 vertices. From this data, the algorithm constructs the similarity matrix, S, whose rows and columns represent the subdomains of the old and intermediate solution, respectively. For instance, if subdomain 2 in the intermediate solution was re-labelled as 1, at least two vertices would be assigned to the same subdomain in both solutions. Although it might seem that, after the change, both solutions would coincide in the first three vertices, this is not the case, since the first vertex has to change its subdomain. Therefore, s_{12} is 2 (and not 3). Once all elements in S have been computed, the algorithm selects three of them, such that every row and column contains exactly one selected element and the sum of the selected elements is maximum. For this purpose, the elements s_{qr} are sorted in decreasing order and the first three of them that fulfill the former constraint are selected in a greedy fashion (i.e., s_{23}, s_{12} and s_{31}, highlighted by a shaded box). The sum of the remaining elements is the distance between the old and new (i.e., re-labelled) partition. The differing vertices in the re-labelled partition are highlighted in bold. From the comparison of all partitions, it is clear that, after re-labelling, the number of changes has decreased from 5 to 3.

Algorithm Template

All the previous techniques can be combined into a single graph partitioning algorithm. The basic heuristic algorithm proposed in this work follows a Scratch-Remap ML approach, as shown in Figure 2.23. The coarsening stage simplifies the original graph by SHEM. Over the coarsest graph, the initial partitioning is performed by CAMS. The subsequent uncoarsening stage projects the partition on the coarsest graph back to the original graph by unfolding the matched vertices. During this stage, connected FM refinement is applied after each uncoarsening operation. Finally, subdomain indices are re-labelled to minimise the number of changes in the network. This method will hereafter be referred to as ML-CAMS algorithm.

```
Stage 1) Coarsening stage
    1.1) Repeat Match vertices by Sorted Heavy Edge Matching algorithm
            i. Rank edges based on decreasing weight by Quicksort algorithm
            ii. Select next edge and match endvertices if not already matched, until no edge is left
                unchecked
         until the average number of vertices per subdomain in the coarsest graph is below a certain
         threshold.

Stage 2) Initial partitioning
    2.1) Build an initial set of $g$ partitions of the coarsest graph by Random Multi-Start algorithm
            i. Build the different partitions by Greedy Graph Growing Partitioning algorithm
            ii. Refine each partition by connected Fiduccia-Matheysses refinement algorithm, identifying
                articulation points by Depth-First Search analysis on each subdomain.
    2.2) Repeat Generate a new set of $g$ partitions from the old set of partitions
            i. Find groups of vertices assigned to the same subdomain in the current set of partitions
            ii. Simplify the coarsest graph by matching vertices in a group into a unique vertex and
                simplify the current set of partitions accordingly
            iii. Refine each partition in the current partition set by connected Fiduccia-Matheysses refinement
                algorithm over the coarsest graph
            iv. Build the new set of solutions by uncoarsening the new solutions from the previous step
         until a number of generations have been reached or the new set of partitions coincide to the
         old one.

Stage 3) Uncoarsening stage
    3.1) Repeat Progressively refine the initial partition on the coarsest graph
            i. Uncoarsen the coarser graph based on the matching scheme
            ii. Refine the current partition by connected Fiduccia-Matheysses refinement algorithm
         until the finer graph is the original graph.

Stage 4) Post-processing stage
    4.1) Build similarity matrix $S$ between old and new solution
    4.2) Select $k$ elements in $S$ such that every row and column has one selected element
    4.3) For every element $s_{qr}$ selected, rename subdomain $r$ in the new solution as subdomain $q$.
```

Figure 2.23: Template of the proposed heuristic graph partitioning algorithm.

2.3.4 Time-Complexity Analysis

The previous sections have presented two methods of solving the CPAP: one exact method based on the application of the BC algorithm to an ILP model of the problem, and one heuristic method based on the ML-CAMS algorithm. Before testing them in practice, it is interesting to evaluate their computational complexity in theory.

ILP-BC method

Exact methods follow an enumerative approach to search the entire solution space. Thus, their theoretical worst-case time complexity has an upper bound in the pure brute-force approach. In the case of the *min k-cut* problem, this naive approach requires the enumeration of the different ways to assign the $|V|$ vertices to the k subdomains. The number of solutions can thus be computed as the number of variations with repetitions of k elements taken in groups of $|V|$. The size of the solution space is therefore $k^{|V|}$, which makes this approach impractical for graphs of non-trivial size. Although the search space can be significantly reduced in BC methods, several studies prove that most BC methods still have exponential time complexity in the worst case [84]. This exponential dependence is observed on both the number of variables and constraints of the ILP model, which in turn were proved to be $\mathrm{O}(|V| + |E|)$.

ML-CAMS method

The heuristic method proposed in this work is built upon several algorithms. Therefore, the complexity of the algorithm depends, to some extent, on all individual algorithms. Table 2.1 presents the theoretical worst-case time complexity for each algorithm in the heuristic method. From the table, it can be deduced that the overall worst-case time complexity is that of the connected FM refinement algorithm, and all the algorithms that make use of it. The complexity is $\mathrm{O}(|V|^2(|V| + |E|))$, as a DFS must be performed for every vertex in the source subdomain after each vertex exchange. Although this time complexity seems to be high, in practice runtime does not follow that trend for two reasons. Firstly, the worst-case value is often a very pessimistic estimation of the runtime of an algorithm, which is especially true for the connected FM refinement algorithm with the type of graphs used here. Secondly, the graphs handled in this application are of medium size, so the hidden constants and terms in the O-notation might have an influence on the runtime. Nonetheless, it is expected that most of the computational load is due to the connectedness checks for larger graphs. Finally, it is worth noting that some of the algorithms are only applied over simplified versions of the graph (e.g., initial partitioning by CAMS) and should therefore have a small impact on the overall execution time. All these theoretical results are confirmed by the runtime analysis presented in Appendix A.

2.4 Field Trial

Once the algorithms to solve the CPAP have been described, this section is devoted to the analysis of the field trial results presented in [20]. The purpose of this initial test was to justify the need for the PCU re-planning process over a live GERAN. By comparing the current operator configuration with a simple heuristic solution, the gain of the optimisation process

Component	Worst-case
Quicksort	$O(\|E\|^2)$
Sorted Heavy Edge Matching	$O(\|E\|^2)$
Coarsening stage	$O(\|E\|^2)$
Depth-First Search	$O(\|V\| + \|E\|)$
k-way Greedy Graph Growing Partitioning	$O(\|V\|^2)$
Non-connected FM refinement	$O(\|E\|)$
Identification of articulation points	$O(\|V\|(\|V\| + \|E\|))$
Connected FM refinement	$O(\|V\|^2(\|V\| + \|E\|))$
Clustered Adaptive Multi-Start	$O(\|V\|^2(\|V\| + \|E\|))$
Multi-Level Clustered Adaptive Multi-Start	$O(\|V\|^2(\|V\| + \|E\|))$
Uncoarsening stage	$O(\|E\|log\|E\|)$
Remapping	$O(k^2\|V\|)$

Table 2.1: Worst-case time complexity of algorithms in the heuristic method.

could be roughly estimated. For clarity, the trial set-up is described first and the trial results are discussed later.

2.4.1 Trial Set-up

The following description gives a brief outline of the scenario, the experiments carried out and the criteria adopted to assess the value of the method during the trial.

Trial Scenario

The trial area consisted of one BSC providing seamless coverage. The trial BSC comprised 139 cells distributed over 58 sites and 6 PCUs. Figure 2.24 shows the graph that models the CPAP in the trial BSC. Each vertex is represented on the site where the BTS is physically located, as if it were on a map. Graph weights have been removed for the sake of clarity. From the figure, it is clear that CPAP graphs are non-planar. Likewise, it is observed that CPAP graphs can be highly heterogeneous, especially in terms of local structure, unlike graphs derived from meshes in the supercomputing area, which have frequently been used to develop and test partitioning algorithms.

Assessment Methodology

A computer programme was created to test a simple heuristic method in a real environment. The inputs of the programme were the configuration of GPRS (i.e., number of PCUs per BSC and GPRS TSLs per cell) and the CS-HO statistics for a 9-day period, both located in the NMS. As optimisation constraint, the maximum number of GPRS TSLs in a PCU was set to 256 due to physical hardware limitations (i.e., $B_{aw} = 256$). Following operator demand, a maximum weight imbalance ratio of 2 was permitted, so that the weight of the smallest subdomain had to be, at least, half of the largest one (i.e., $B_{rw} = 2$).

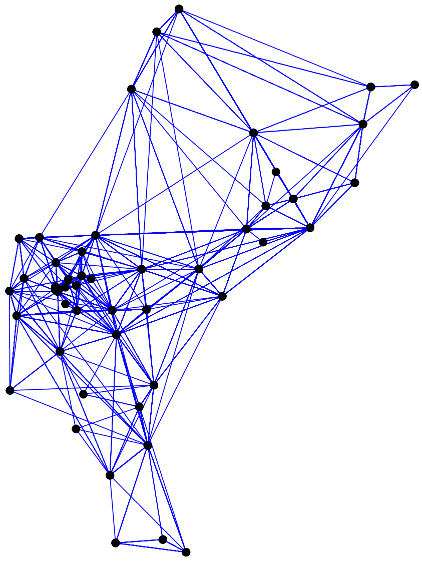


Figure 2.24: The graph of the trial BSC.

Based on CS-HO statistics and GPRS configuration, a new PCU plan had to be created for the trial BSC. It is clear that the complete enumeration approach is impractical, as, even in the site-level assignment case, it requires evaluating 6^{58} solutions. Instead, a heuristic method was used. To reduce the development effort in this preliminary trial, the core of the method was the graph partitioning algorithm in METIS software. METIS [64] is a high performance graph partitioning package from the University of Minnesota, which is available in the public domain. This package provides several state-of-the-art graph partitioning algorithms, based on the approach used in most commercial codes: the ML algorithm with HEM coarsening, initial partitioning by R-GGGP and non-connected FM refinement. As will be shown in the last section of this chapter, this approach proves to find solutions much better than the one currently implemented in the network. Unfortunately, the stand-alone version of METIS does not have support for generating bounded partitions (i.e., partitions with a slight weight imbalance between subdomains). On the contrary, the standard algorithm aims to achieve perfect balance among subdomains. To circumvent this limitation, the METIS solution was post-processed by a non-connected FM refinement routine built from scratch. The execution time of the overall algorithm was about 30s on a 2.4GHz 1GByte-RAM computer, most of which was spent on the refinement process.

Performance statistics were gathered before and after the new PCU plan was implemented in the network to assess the value of the method. As the core of the assessment methodology, drive surveys were carried out to quantify the benefits in terms of data throughput and service break duration. Two different routes (referred to as A and B) were defined, encompassing a total of 314 km (7 hours drive). Figure 2.25 depicts the routes on a road map. Dots on the map represent locations of BTSs in the BSC and lines represent roads in the BSC area. It is worth noting that the selected routes covered a variety of environments, ranging from dense-urban to open rural. Consequently, the drive survey included different driving conditions, from almost pedestrian in the city centre to very fast moving on the open motorways.

During the drives, a mobile terminal was configured to repeatedly download a 10MB data file via FTP (File Transfer Protocol). This test set-up was selected to utilise a traffic source offering

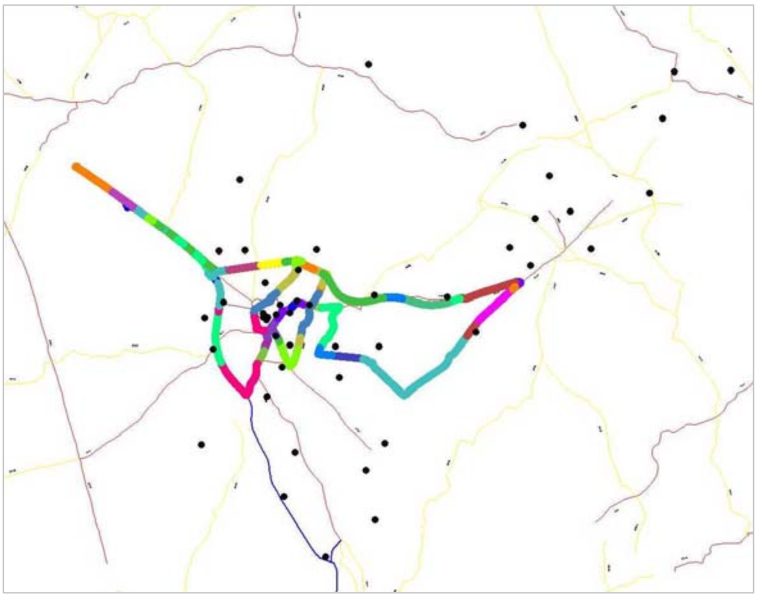


Figure 2.25: Map view of the routes in the trial area.

a steady traffic flow. Thus, measurement variability is reduced to the statistical variation of the service gap and the interpretation of results is simplified. The data logging equipment was used in a 3 Downlink (DL) + 1 Uplink (UL) TSL configuration.

Assessment Criteria

As main performance indicator, the DL application throughput was collected with a resolution of 1s. The term ”application” is used here to stress that this figure excludes headers associated to Radio Link Control and Medium Access Control protocols. Although a maximum DL throughput of 3*12=36kbps can be achieved by such a configuration with the coding schemes CS1-CS2 currently implemented in the trial network, the actual throughput depends on the propagation conditions experienced by the user and the overall traffic demand in the trial area. It is worth noting that these conditions were not controllable during the trial, since coordination with ordinary subscribers in the network was impossible for obvious reasons. Nonetheless, it should be pointed out that the traffic demand remained virtually unchanged during the trial. Concretely, the total number of temporal connections for data transmission (called temporary block flows) established for the DL in the trial area differed less than 0.5% between the before and after periods. Likewise, CS traffic, which has a higher priority than PS traffic, and might thus have an influence on the available TSL resources, varied less than 2.5% between periods.

The impact of the new PCU plan on data rate was analysed based on the data collected. The analysis focused on every CRS event (and not on the entire route), since the influence of the new plan on data performance should be confined to the vicinity of these events. Thus, the influence of propagation and traffic conditions is minimised. Figure 2.26 reflects how the observation period is defined around a CRS event. A time period of 20s around CRS events was considered appropriate to reflect the influence on data performance. This period proves large enough to encompass the largest service gaps (i.e., up to 15s), but small enough to reject the period of bad propagation conditions that might occur before cell change. For this period, the mean DL throughput and the service break duration were computed for each CRS.

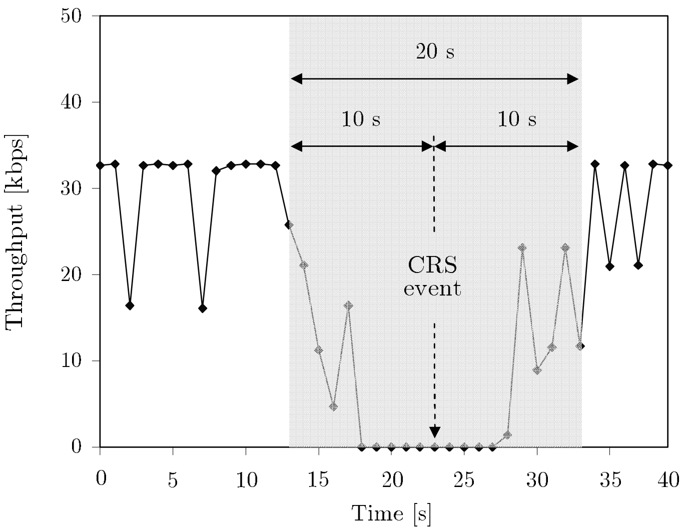


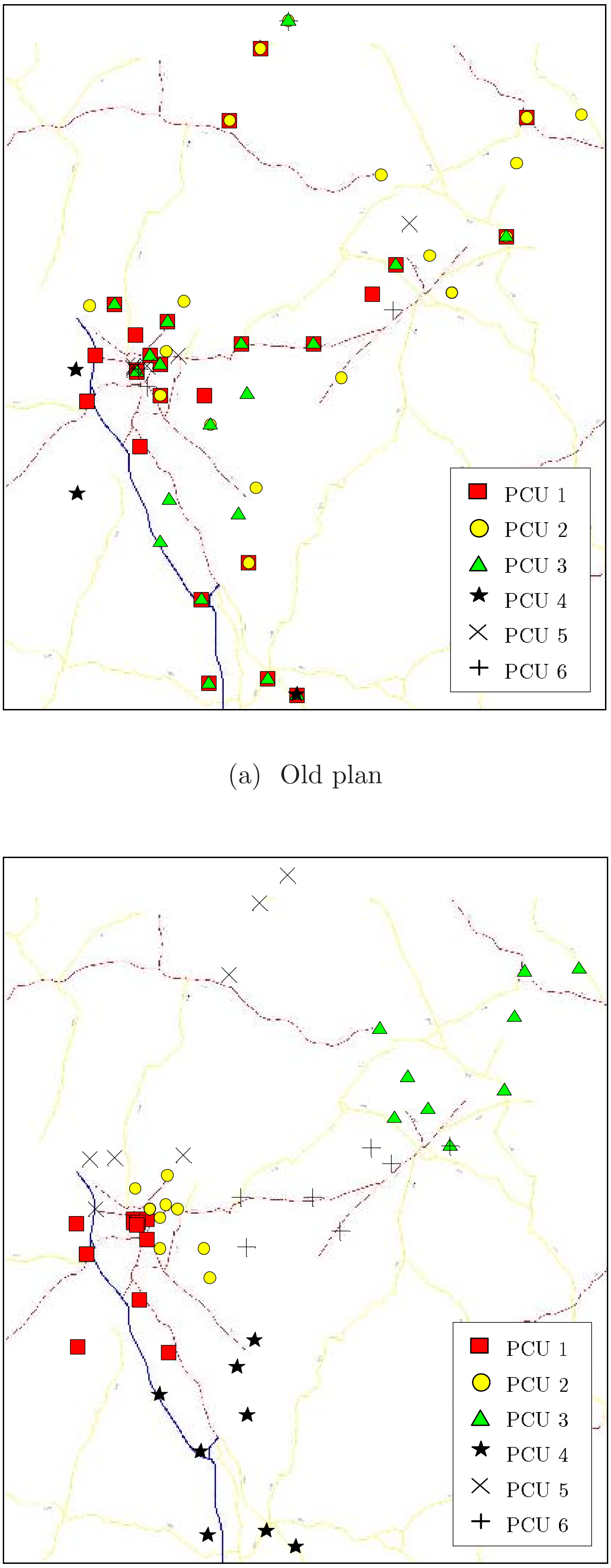
Figure 2.26: Measurement period around a cell re-selection event.

2.4.2 Trial Results

From the application of the method, a new PCU plan was built for the trial BSC. To visually inspect the impact of the new PCU plan, Figure 2.27 (a)-(b) provide a map view of the old and new PCU plans over the geographical area of the BSC. A symbol on the map displays the location of a BTS. Each of the six different symbols represents a different PCU on the BSC. Figure 2.27 (a) shows that, in the old plan, cells on the same PCU (denoted by the same symbol) do not always form a contiguous area. This issue is especially serious in the vicinity of main roads, where most CRSs will take place. It is also remarkable that cells on the same site are sometimes assigned to two or even three different PCUs. By contrast, the new plan in Figure 2.27 (b) shows that cells belonging to the same PCU are grouped in geographical clusters. Therefore, for a randomly selected drive route in the BSC area, the number of inter-PCU CRSs in the new PCU plan should be lower than the respective value in the old PCU plan.

Table 2.2 presents the main performance indicators collected during the drive tests, broken down by PCU plan and route. The different types of CRS are first compared based on the figures at the bottom row of Table 2.2. The average application throughput during intra-PCU and inter-PCU CRS events was 18.36 and 11.05 kbps, respectively. Similarly, the mean duration of the service gap in the intra-PCU and inter-PCU cases was 4.55 and 9.32s, respectively. These figures clearly indicate that it is beneficial to increase the number of intra-PCU CRSs while decreasing the number of inter-PCU CRSs, which is the aim of the PCU planning activity. It is worth noting that these observations remain valid, regardless of the PCU plan and drive route.

In Table 2.2, it is observed that the share of intra-PCU CRSs was doubled with the new PCU plan (i.e., 33.2% to 67.6%). End-user throughput figures provide evidence of the benefit from the optimised PCU plan. Thus, the overall average throughput increased from 12.94 to 16.58 kbps (i.e., 28%). Likewise, the average service break duration decreased from 8.10 to 5.69s (i.e., 30%). This performance enhancement was mainly obtained because a large number of inter-PCU CRSs were converted to intra-PCU CRSs. It should be pointed out that the quoted throughput benefit is for the 20s period around the CRS event and not for the entire drive route. The gain obtained on the entire drive route is obviously lower and strongly depends on the ratio of time spent in CRS to the time outside CRS, which is influenced by the drive route.



(a) Old plan

(b) New plan

Figure 2.27: Map view of the old and new PCU plans of the trial BSC.

PCU Plan	Route	Avg. throughput[kbps]		Avg. break duration[s]		Nbr. of		Overall intra-PCU CRS ratio [%]	Overall average throughput [kbps]	Overall average break duration[s]
		Intra-PCU CRS	Inter-PCU CRS	Intra-PCU CRS	Inter-PCU CRS	Intra-PCU CRS	Inter-PCU CRS			
Old	A	18.71	9.94	4.42	11.12	31	59	34.4	12.96	8.81
Old	B	17.34	10.84	5.39	8.42	33	70	32.0	12.92	7.45
Old	A+B	18.00	10.43	4.92	9.68	64	129	33.2	12.94	8.10
New	A	18.34	13.20	4.3	7.71	61	31	66.3	16.61	5.45
New	B	18.81	11.59	4.4	9.42	58	26	69.0	16.58	5.95
New	A+B	18.55	12.47	4.35	8.49	119	57	67.6	16.58	5.69
Old+New	A+B	18.36	11.05	4.55	9.32	183	186	-	-	-

Table 2.2: Performance comparison of the different PCU plans and routes during the trial.

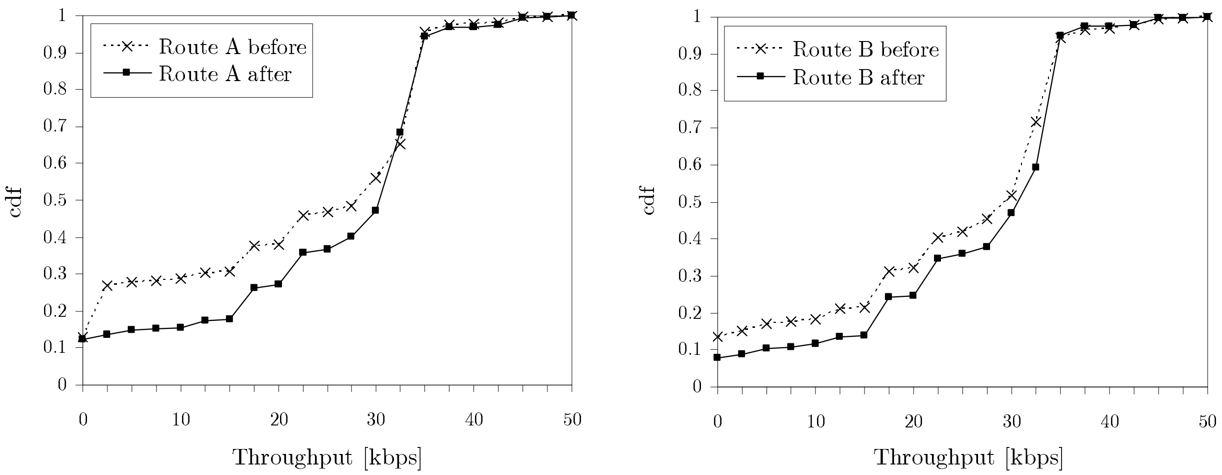
PCU Plan	PCU id.	Nbr. of Intra-PCU HOs	Nbr. of Inter-PCU HOs	Intra-PCU HO Ratio[%]	Nbr. of cells	Nbr. of GPRS TSLs
Old	1	786059	604490	56.5	44	253
	2	562922	558241	50.2	42	254
	3	384128	460415	45.5	35	214
	4	3732	45637	7.6	6	36
	5	114	16116	7.0	5	42
	6	1737	37809	4.4	7	48
	Overall	1738692	1722708	50.2	139	847
PCU Plan	PCU id.	Nbr. of Intra-PCU HOs	Nbr. of Inter-PCU HOs	Intra-PCU HO Ratio[%]	Nbr. of cells	Nbr. of GPRS TSLs
New	1	117649	739051	86.3	20	112
	2	66771	175231	72.4	18	108
	3	115178	530676	82.2	24	148
	4	44777	459687	91.1	23	147
	5	52179	374230	87.8	21	129
	6	125656	660315	84.0	33	203
	Overall	522210	2939190	84.9	139	847

Table 2.3: Results of the graph partitioning algorithm based on CS-HO statistics.

However, once the focus is on CRS events, this increase in data throughput is independent of the drive route.

Although the selected drive routes cover a large part of the BSC area, the drive survey can still only provide a rough indication of the change in the ratio of intra-PCU to inter-PCU CRS for the whole BSC. To provide such information, Table 2.3 shows the estimated performance of the graph partitioning method based on NMS CS-HO statistics in the entire trial area. The intra-PCU HO ratio increases after optimisation from 50.2% to 84.9% (i.e., an increase of 34.7% in absolute terms and 69% in relative terms). While the intra-PCU HO ratio varied significantly from PCU to PCU in the old plan, small variations are observed in the new plan. Likewise, the number of GPRS TSLs is more evenly balanced in the new plan, which translates into a reduction of the maximum weight imbalance ratio from 6.05 (=254/42) to 1.88 (=203/108), i.e., a threefold reduction. Thus, the need for additional PCUs in the future is minimised, as spare capacity now exists in all PCUs. From this data, it can be concluded that the same trends observed in the drive tests are also seen in the NMS-based estimations.

To complete the analysis, some figures of the entire drive route (and not only on the CRS event) are also presented. Figure 2.28 (a)-(b) show the empirical cumulative density function of user throughput over routes A and B. Both figures show that the probability of having less than a certain throughput is lower (i.e., better) after optimisation. It is clearly observed that the ratio of low throughput samples in both routes decreases after optimisation. This effect yields an increase of the overall mean data throughput from 23.4 to 24.8 (i.e., 6%), and a decrease of the standard deviation from 12.5 to 11.5 (i.e., 8%). It is worth noting that this performance improvement was not only due to a reduction of the CRS delay, but also came from a reduction of the GPRS territory upgrade rejection ratio in the trial area, which decreased from 19.6% to 6.6%. This benefit was a side effect from the load balance among PCUs, as explained below.



(a) Drive route A

(b) Drive route B

Figure 2.28: Empirical cumulative density function of user throughput over the entire route.

The capacity of a PCU is limited by the maximum number of GPRS TSLs that can be used simultaneously in the cells assigned to it. In contrast to what has been assumed so far, the number of active GPRS TSLs in a cell (i.e., GPRS territory) is not fixed, but fluctuates depending on the traffic demand of both CS and PS services and the available TSL resources. Thus, large fluctuations of traffic demand in some isolated cells might occasionally reach the PCU capacity limit. Under this condition, any new request for extending the GPRS territory in the cells of the PCU (i.e., GPRS territory upgrade) would be rejected, even if there were available TSLs in the cells. This event, called a *rejection due to PCU congestion*, can take place in any of the two scenarios of territory upgrade: the automatic recovery of the default GPRS territory after being occupied by CS traffic, or the extension of the GPRS territory into the CS default territory if the number of GPRS users per TSL is excessive. The main effect of this event is an under-utilisation of TSL resources and a degraded performance for GPRS users.

After equalising the load among PCUs, less territory upgrades were rejected, as it was less likely that a PCU became overloaded, even for peaks of GPRS traffic demand. Consequently, the inclusion of TSLs into the GPRS territory became faster. This effect brought a global performance enhancement in the BSC, which should affect every single GPRS users, whether mobile or static. For instance, the ratio of time when the number of DL TSLs assigned to the trial terminal was less than its maximum capacity (i.e., 3 TSLs) decreased from 4.7 to 2.1% with the new PCU plan. Although the increase of the average number of assigned TSLs in the entire route was small (i.e., from 2.91 to 2.96), this gain proved to be concentrated after CRS events, where the GPRS territory upgrade is more frequent. This result would explain that the average service break for both inter- and intra-PCU CRSs were slightly shorter after optimisation, as observed in Table 2.2. In particular, the service break for the intra-PCU case decreased from 4.92 to 4.35 (i.e., 11%), whilst it decreased from 9.68 to 8.89 (ie. 8%) for the inter-PCU case. This reduction is thought to be due to a faster assignment of GPRS TSLs to the new user when entering the target cell.

From the trial results, it can be concluded that it is beneficial to keep strongly-related cells under the same PCU, since inter-PCU CRSs cause longer service breaks than intra-PCU CRSs. Likewise, HO statistics can give a rough estimation of the benefit from the PCU re-planning process. The previous statements are true, regardless of the selected BSC. However, the previous results fail to give an accurate estimation of the maximum benefit that can be achieved by optimising the PCU plan, since only a simple heuristic algorithm was tested. In this sense, it is worth stressing that routines in METIS are designed to give the best results in graphs from the supercomputing area, which have important differences with CPAP graphs. This motivates the development of new algorithms. In addition, it might be argued that results could be dependent on the selected BSC. Thus, a single BSC covers a small network area, which might not be representative of the whole network. For both reasons, the analysis is extended by the experiments reported in the next section.

2.5 Analysis over Measurement-Based Network Model

Once trial results have shown the potential of PCU re-planning, the following analysis aims to find the most suitable method to solve the CPAP. In particular, the main concern is to prove the value of the ILP-BC method for benchmarking purposes and the capability of the ML-CAMS method to provide fast and high quality solutions. For that purpose, an extensive set of graphs is constructed from HO statistics of a live GERAN. Over these graphs, the performance of the proposed methods is estimated and compared with several classical approaches. For clarity, the preliminary conditions are described first and the results of the analysis are subsequently presented.

2.5.1 Analysis Set-up

The following paragraphs outline the network area where the statistical data is extracted, the experiments carried out during the analysis and the criteria adopted to assess the value of the different methods.

Analysis Scenario

The network area under study comprises 8952 cells (4216 sites) distributed over 61 BSCs providing seamless coverage. As the CPAP is solved on a per-BSC basis, the set of problem instances considered in the analysis consisted of 61 CPAP graphs. It is worth noting that robust performance estimations are expected, since the collection of graphs covers a large geographical area with very different propagation and mobility environments. Table 2.4 presents some relevant statistics of the network area under analysis. These values provide some insight into the attributes of the GPP instances behind the CPAP. Preliminary analysis indicates some significant differences against other application areas. In general, the size of CPAP graphs is not very large, since the number of vertices per graph (i.e., cells per BSC) is several orders of magnitude smaller than graphs in other fields. While graphs of 10^5-10^7 vertices are usually reported in the graph partitioning literature, the CPAP deals with graphs of only hundreds of vertices. Even if all the instances of the CPAP (i.e., BSCs in the network) are taken into account, the problem cannot be considered of extreme size, since the size of the problem only grows linearly with the number of instances. Regarding the number of subdomains per instance (i.e., PCUs per BSC), despite its small absolute value, this figure is high when compared to the number of vertices in the graph. As a consequence, the average number of vertices per subdomain (i.e., cells per PCU) is one order of magnitude below the values commonly reported in the literature. The

	Avg	Std	Min	Max
Nbr. of cells per BSC	146.8	27.5	85	213
Nbr. of adjacencies per BSC	929.3	322.3	241	1907
Nbr. of PCUs per BSC	5.3	1.0	4	8
Nbr. of GPRS TSLs per BSC	361.6	76.7	208	593
Nbr. of cells per PCU	27.8	4.5	15.9	37.4
Nbr. of GPRS TSLs per PCU	68.2	11.4	41.6	91.0

Table 2.4: Main statistics of the scenario in a BSC level.

small number of vertices per subdomain affects the performance of classical graph partitioning algorithms for several reasons. On the one hand, the benefit from ML approaches decreases, as the original graph cannot be simplified further during the coarsening stage (note that the termination condition for coarsening is often expressed in terms of the average number of vertices per subdomain in the coarsened graph). On the other hand, refinement algorithms have less degrees of freedom, since only a small number of vertices can be moved without affecting the balance between subdomains. This problem is exacerbated by the presence of articulation vertices in subdomains.

Figure 2.29 shows the histogram of the number of adjacencies per cell to other cells in the same BSC. An average of 12.7 adjacencies per cell experienced HOs to cells in the same BSC. Since this figure is much lower than the average number of cells in the BSC (i.e., 146.8), the density of the graphs in the CPAP is small. Concretely, the average density is 0.09, which is obviously much lower than 1, but still larger than values reported in the literature. To further clarify this issue, Figure 2.30 presents the number of active adjacencies versus the number of cells in the 61 BSCs. It is observed that, despite the variations in the number of active adjacencies per cell, the total number of active adjacencies per BSC remains proportional to the number of cells in the BSC (or, at least, does not show a quadratic relationship with the latter parameter). Hence, the graphs handled in the CPAP can be broadly classified as sparse, but sparsity is less than in graphs from supercomputing applications.

All the previous features suggest that the instances of the CPAP can be considered of limited size, but harder to solve than other instances of the GPP of the same size. These facts justify the use of solution techniques more sophisticated than the classical ones.

Assessment Methodology

A programme has been created to test different methods with real data. The inputs of the programme are the HO statistics for a 9-day period and the configuration of GPRS, both residing on the NMS of a live GERAN. As optimisation constraints, the maximum number of GPRS TSLs per PCU is 256 (i.e., B_{aw}=256) and the maximum weight imbalance ratio is 2 (i.e., $B_{rw} = 2$).

During the analysis, the main concern is to evaluate the two graph partitioning methods proposed in this work: on the one hand, the exact method based on the application of the BC algorithm, initialised with the solution of the standard ML algorithm, over the ILP model of the problem (denoted as BC); on the other hand, the heuristic ML method that uses SHEM for

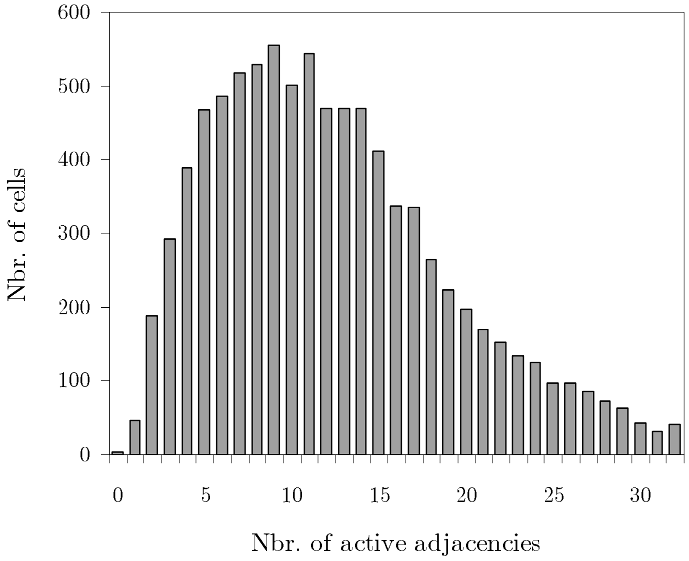


Figure 2.29: Histogram of number of active adjacencies per cell.

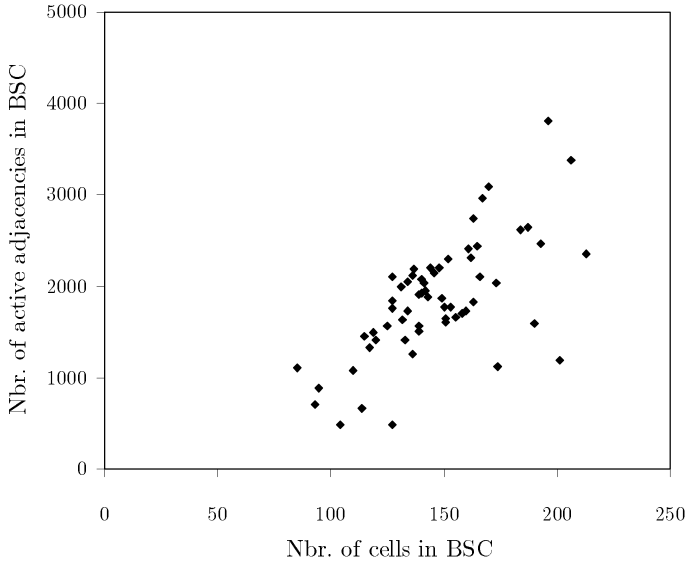


Figure 2.30: Number of active adjacencies versus number of cells in the BSCs.

coarsening, CAMS for initial partitioning and connected FM refinement during uncoarsening (denoted as ML-CAMS).

The first experiments verify several properties of the CPAP, which are the origin of the superior performance of adaptive MS approaches. For that purpose, a naive MS approach is used to explore the optimisation surface. Concretely, a set of 1000 local minima is built in a problem instance by repeated use of the GGGP algorithm with random seeds and greedy refinement (denoted as R-GGGP).

The next experiments quantify the performance benefit that can be achieved by optimising the PCU plan. For this purpose, the performance of the different methods is evaluated over the whole set of CPAP instances. First, the current operator solution (denoted as IO) is compared with the solution of the exact method. Thus, an upper bound for the improvement is obtained.

In this approach, the entire set of ILP models is solved by the BC algorithm under loose runtime constraints. Two different runtime sharing strategies are considered: size-based sharing (denoted as BC-SS) and edgecut-based sharing (denoted as BC-ES).

Subsequently, the analysis is extended to heuristic methods. The first method is the FM refinement of the operator solution (denoted as RO). The next three ones are variants of the GGGP algorithm. The first variant is the FW-GGGP method, which uses the FW algorithm to select seed vertices. The implemented version of this method is deterministic (i.e., the output solution is the same in any run of the algorithm). The second variant is the R-GGGP method, based on the repeated use of GGGP with random seed selection and greedy refinement. The third variant is the adaptive multi-start algorithm, CAMS, where R-GGGP is used to build a set of initial solutions, which are checked for similarities to simplify the original graph iteratively. Another method is the standard ML method (denoted as ML), where initial partitioning is performed by coarsening the graph until the number of vertices is the same as the number of subdomains. Finally, the analysis considers the proposed ML-CAMS method, which coarsens the graph partially and uses CAMS only for the initial partitioning.

All heuristic methods share the connected FM refinement algorithm, except R-GGGP, which uses the greedy variant. Unless stated otherwise, the number of passes in the FM algorithm is 4. Likewise, all ML methods use the SHEM algorithm for coarsening, except CAMS, where matching is based on similarities of previous solutions. Finally, it is worth noting that OR, FW-GGGP and ML are deterministic. Consequently, a single run of these algorithms is performed. In contrast, CAMS, ML-CAMS and R-GGGP are randomised (and hence produce a different solution for each different random seed). To ensure the statistical confidence of results, performance figures for CAMS and ML-CAMS are computed from the average of 100 independent runs, while figures for R-GGGP correspond to 1000 independent runs of the algorithm (i.e., the edge-cut of the best attempt and the runtime of the whole series of attempts).

Assessment Criteria

Several performance indicators are evaluated during the assessment process. From the operator perspective, the main figure of merit is the intra-PCU HO ratio. Formally, this quantity represents the edge-cut normalised by the total sum of edge weights, which is hereafter referred to as *edge-cut ratio*. The PCU load-imbalance ratio and the number of PCUs with disconnected cells are considered as secondary criteria. Finally, the processing time is also evaluated. For that purpose, the different routines are run on a Windows-based computer with a clock frequency of 2.4GHz and 1GByte of RAM.

To assess the value of a given algorithm, the trade-off between runtime and solution quality is evaluated following the methodology described in [36]. Most optimisation algorithms contain a parameter that allows the user to specify how long the search for an optimal solution should continue before giving up (e.g., passes in the refinement algorithm, attempts in the multi-start algorithm). This parameter is commonly denoted as the *intensity* of the algorithm. For each intensity value, ρ, and problem instance, i, the edge-cut, Q, and the runtime, T, are averaged across different runs of the algorithm, r, to give

$$\overline{Q}_{\rho,i} = \frac{1}{N_r}\sum_{r=1}^{N_r} Q_{\rho,i,r} \ , \qquad \overline{T}_{\rho,i} = \frac{1}{N_r}\sum_{r=1}^{N_r} T_{\rho,i,r} \ , \tag{2.55}$$

where N_r is the number of independent runs.

To estimate the overall performance, relevant indicators are aggregated (and not averaged) across the different instances of the problem, i, as

$$Q_{\rho T} = \sum_{i=1}^{N_p} \overline{Q}_{\rho,i} \ , \qquad T_{\rho T} = \sum_{i=1}^{N_p} \overline{T}_{\rho,i} \ , \tag{2.56}$$

where N_p is the number of problem instances (i.e., BSCs) and the subindex $(\cdot)_T$ stands for total.

For clarity, normalisation against the performance of some reference solution is used to calculate the final performance figures. Thus, the normalised edge-cut, Q_ρ, and normalised runtime, T_ρ, are defined as

$$Q_\rho = \frac{Q_{\rho T}}{{Q_{\rho T}}^*} \ , \qquad T_\rho = \frac{T_{\rho T}}{T_{\rho T}^*} \ , \tag{2.57}$$

where ${Q_{\rho T}}^*$ is the total edge-cut of the best heuristic solution a priori (i.e., R-GGGP) and $T_{\rho T}^*$ is the total runtime of the fastest method (i.e., ML with no refinement). By using different intensity values, ρ, the trade-off between Q_ρ and T_ρ can be investigated to give an indication of the performance of the different approaches.

Implementation Issues

Although several codes are available in the public domain to solve the GPP, several limitations prevent them from being applied in the problem considered here. For this reason, almost all graph partitioning routines in this work have been developed from scratch. This approach has several advantages that are listed below:

a) *Problem formulation*: Several formulations exist for the GPP and not all are considered by the available tools. The main issues are related to the formulation of problem constraints. Thus, it is sometimes not possible to adjust the maximum weight imbalance between subdomains, since perfect balance is almost always targeted. At the same time, the definition of the weight imbalance ratio in these tools does not coincide to the one used by operators. Likewise, no geographical restrictions can be imposed. By coding new routines, the CPAP problem can be formulated more precisely, resulting in solutions that comply better with the requirements of cellular network operators.

b) *Solution techniques*: In the design of standard graph partitioning methods, some compromise is reached to ensure that adequate performance is obtained under a wide variety of graphs. Thus, most public codes use simplified algorithms to achieve small runtimes in large graphs. A first example of this trend is the avoidance of sorting edges in HEM, which might cause that the heaviest edges were not included in the matching. Another example is the lack of connectedness checks in most FM implementations. Although some actions are normally taken to avoid disconnected solutions, experiments carried out in this work have shown that, in some cases, disconnected subdomains still exist in the final solution. Hence, it is clear that adapting to the peculiarities of graphs handled in a particular application can give a significant performance improvement. In particular, the limited size of CPAP graphs allows the use of more sophisticated algorithms, such as SHEM for coarsening, CAMS for initial partitioning and connected FM for refinement.

c) *Validation process*: A number of codes are available in the form of libraries, which can easily be integrated into other programs as independent modules. This approach would be suitable if the assessment process was restricted to the quality of solutions. However, runtime also has to be evaluated. Thus, the combination of modules that are coded in different programming languages complicates (or even prevents) the time complexity analysis. By writing all methods in the same programming language, the influence of implementation issues on algorithm performance is reduced. This allows a fair comparison among methods in terms of runtime.

The exact method based on the ILP formulation of the CPAP is an exception, as it is based on the BC algorithm in the CPLEX optimisation package [59]. In this case, the development work was restricted to the routines involved in the construction of the ILP model. The main aim of this decision was to reduce runtime as much as possible, as this method is extremely computationally expensive. Since this method was initially conceived as a benchmark for solution quality, runtime analysis was secondary. Nonetheless, it is interesting to check if this method can be applied in a live situation under operator's time constraints with state-of-the-art routines.

2.5.2 Analysis Results

The following discussion is first focused on the optimisation surface and later on the performance of the different solution techniques.

Optimisation Surface

The initial goal of the analysis is to obtain some prior knowledge about the optimisation surface in the CPAP. In particular, the preliminary analysis examines two main concerns: a) the identification of a statistical model for the values of random local minima, and b) the proof of correlation among the best solutions in the solution space. For that purpose, a set of 1000 random local minima is initially computed on several instances of the CPAP by means of R-GGGP with connected greedy refinement. This set of solutions is expected to be representative of the optimisation surface, provided that seed vertices in the GGGP algorithm are selected randomly.

The simplest approach to build a distributional model for a sample is by means of graphical techniques. In this approach, the first step is to generate a histogram for the sample data. Figure 2.31 shows the histogram of edge-cut values of the local minima achieved by R-GGGP on the CPAP instance considered in Figure 2.16. The only difference here is the inclusion of connectedness checks in the refinement algorithm. It is observed that the histogram exhibits a fairly regular structure, but it is clearly asymmetrical, with a tail extending to the right. Therefore, the distribution cannot be classified as "normal", since the skewness coefficient is very different from 0.

The previous histogram suggests that members in the extreme value distribution family might be a better distributional model for the local minimum values. An *extreme value distribution* is the distribution of the extreme order statistic (i.e., the maximum or minimum) for a

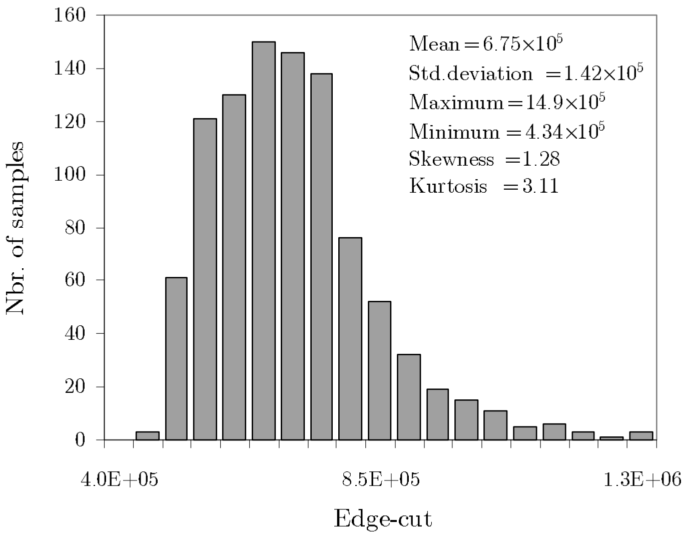


Figure 2.31: Histogram of values of local minima from R-GGGP algorithm.

very large collection of random observations from the same arbitrary distribution. In particular, the *extreme value distribution - type I* (also known as *Gumbel distribution*) proves to be an adequate model for the edge-cut values. The probability density function (PDF) of the Gumbel distribution (maximum) is

$$f(x;a,b) = \frac{1}{b}\exp\left[-\frac{x-a}{b} - \exp\left\{-\frac{x-a}{b}\right\}\right] \qquad -\infty < x < \infty, \tag{2.58}$$

where a is the location parameter and b is the scale parameter. The corresponding cumulative density function (CDF) is

$$F(x;a,b) = \exp\left[-\exp\left\{-\frac{x-a}{b}\right\}\right] \qquad -\infty < x < \infty. \tag{2.59}$$

Figure 2.32 depicts the PDF of the standard Gumbel distribution, where $a = 0$ and $b = 1$. In the figure, it is clearly observed that the location parameter is also the mode (i.e., the most frequent value) of the distribution.

To assess how well the distributional model fits the data set, the CDF of both the sample data and the theoretical model are compared. For this purpose, the *empirical cumulative distribution function* (ECDF) of the observed random variable X (i.e., the edge-cut) is defined as

$$F_n(x) = \frac{\#(X \leq x)}{N_s} \tag{2.60}$$

where $\#(X \leq x)$ represents the number of samples where X is less than or equal to x, and N_s is the sample size. In practice, the ECDF is computed by ordering the observations $x^{(1)} \leq x^{(2)} \leq \cdots \leq x^{(n)}$ and slightly modifying (2.60) as [85]

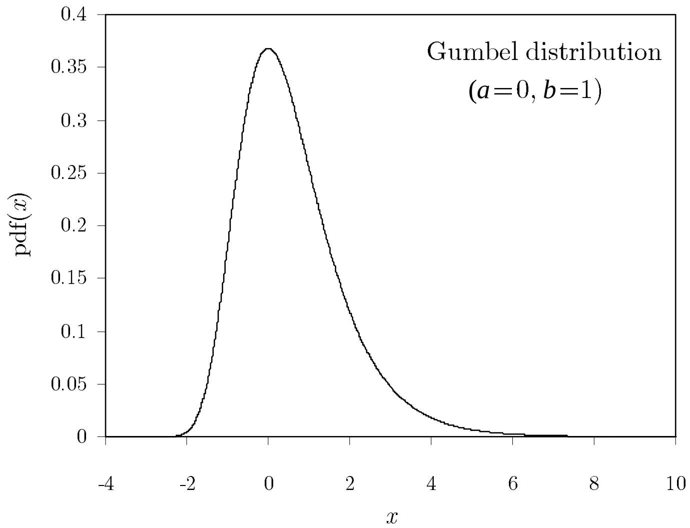


Figure 2.32: Probability density function of the standard Gumbel distribution.

$$F_n(x^{(i)}) = \begin{cases} 1 - F_n(x^{(N_s)}) & \text{for } i = 1 \\ \frac{i-0.3175}{N_s+0.365} & \text{for } i = 2, \cdots, N_s - 1 \\ 0.5^{1/N_s} & \text{for } i = N_s. \end{cases} \tag{2.61}$$

Direct comparison of ECDF and CDF values is not normally performed, as it is difficult to visually check the similarity of two curves. Instead, a probability plot eases the comparison between the theoretical and practical model. Given the theoretical CDF, $T(z)$, a *probability plot* is a plot of $z = T^{-1}(F_n(x))$ on x, i.e., a representation of the ordered observations $x^{(1)} \leq x^{(2)} \leq \cdots \leq x^{(n)}$ against the values $z^{(1)} \leq z^{(2)} \leq \cdots \leq z^{(n)}$ that display the same CDF value in the theoretical model [85]. The latter values are easily computed if an analytical expression of the inverse of the CDF (i.e., the percent point function) is available for the theoretical model. This sort of representation offers the possibility to judge the fit based on the deviation from a straight line, which is much easier than visually inspecting the closeness of two curves. At the same time, the correlation coefficient of the series in both axis gives an indication of the linear fit, which can be used as an objective measure of the goodness of fit.

Figure 2.33 presents the Gumbel probability plot of the local minima values in three instances of the CPAP. The linear relationship between both axis indicates clearly that the selected distribution does in fact fit the data very well in all instances. To reinforce this statement, the values of the squared sample correlation coefficient, R^2, are superimposed on the figure. The values of R^2 close to 1 give evidence of the strong correlation. This good fitting is remarkable for the lowest edge-cut values, which is the interval receiving most of the attention from the optimisation perspective. Although the previous figure only presents the results of three instances, the same trend is observed in the remaining instances. More precisely, the average and minimum value of R^2 in the set of instances are 0.980 and 0.903, respectively. From this result, it can be concluded that the local minimum values in any CPAP instance follow a Gumbel distribution.

From Figure 2.33, it can also be deduced that, although the distributional model remains valid for all instances, the parameters in the model depend on each specific instance (otherwise, the curves of different instances would coincide). Hence, the parameters of the distribution must

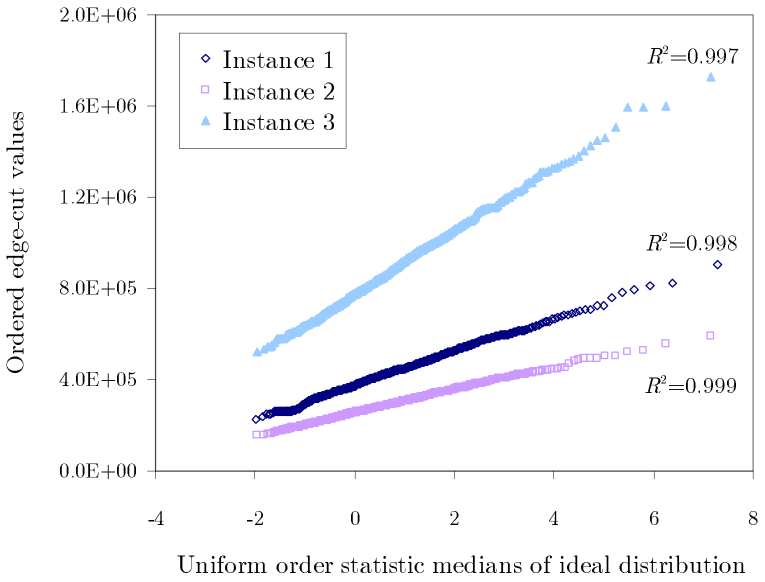


Figure 2.33: Probability plot of values of local minima values in several problem instances.

still be estimated on a per-instance basis in order to fully exploit the information supplied by the analytical model. The maximum likelihood estimates for these parameters can be computed from the sample as [85]

$$\widehat{b} = \sum_{i=1}^{N_s} \frac{X_i}{N_s} - \frac{\sum_{i=1}^{N_s} X_i \exp(-\frac{X_i}{\widehat{b}})}{\sum_{i=1}^{N_s} \exp(-\frac{X_i}{\widehat{b}})} \tag{2.62}$$

$$\widehat{a} = -\widehat{b} \ln \frac{\sum_{i=1}^{N_s} \exp\left(-\frac{X_i}{\widehat{b}}\right)}{N_s}, \tag{2.63}$$

where X_i is the edge-cut value of solution i and N_s is the sample size. Equation (2.62) must be solved iteratively for $\widehat{b}$ and then used to solve (2.63) for $\widehat{a}$. An initial estimate of both parameters can be computed based on the properties of the ideal Gumbel distribution as

$$\widehat{b} = \frac{\sqrt{6}\sigma_s}{\pi} \tag{2.64}$$

$$\widehat{a} = m_s - 0.57722\widehat{b} \tag{2.65}$$

where m_s and σ_s are the sample mean and standard deviation, respectively. These initial estimates prove to be close to the maximum likelihood values in most cases. However, the accuracy relies heavily on the validity of the fit to the Gumbel distribution. Thus, instances with a worse fit showed a larger difference between the values obtained with (2.62)-(2.63) and (2.64)-(2.65). For this reason, the maximum likelihood estimates are preferred.

The analytical tool described so far can be used to show how the probability of improving a previous solution quickly diminishes from one attempt to the next in the naive MS approach. Thus, the number of R-GGGP attempts that must be performed to obtain a solution with a certain minimum quality (i.e., maximum edge-cut) is a random variable geometrically distributed. From (2.59), it can be deduced that the probability of success on each individual attempt is

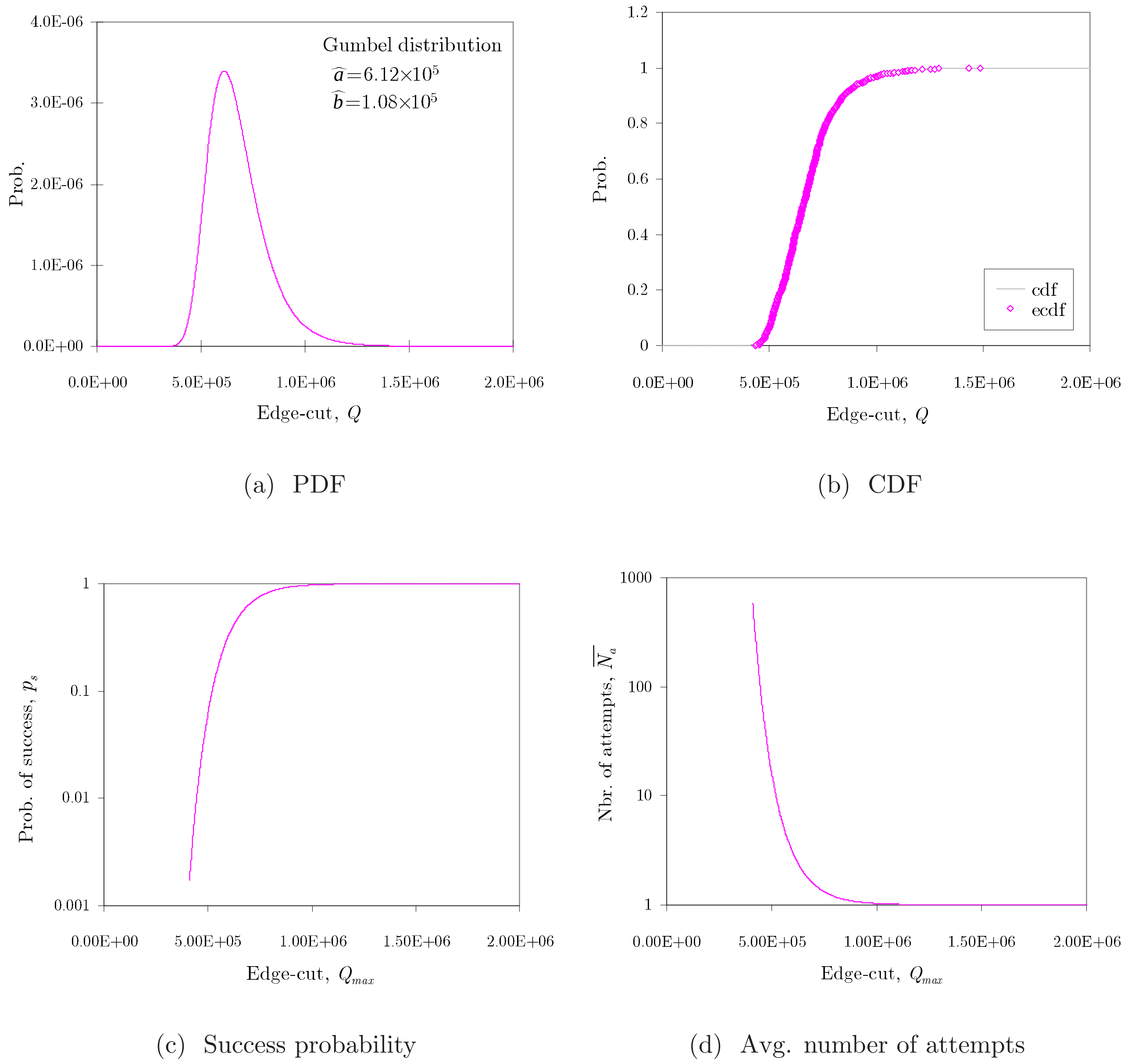
$$p_s = F(Q_{max}; a, b) = P(x \leq Q_{max}) = \exp\left[-\exp\left\{-\frac{Q_{max}-a}{b}\right\}\right] , \tag{2.66}$$

where Q_{max} is the target edge-cut value to be achieved. The average number of attempts until success can be calculated from p_s as

$$\overline{N_a} = \frac{1}{p_s} \quad . \tag{2.67}$$

Figure 2.34 shows how the previous formulas can be used to estimate the number of R-GGGP attempts required to reach a solution with a pre-defined edge-cut in a CPAP instance. The example is based on the set of local minima considered in Figure 2.31. First, the edge-cut PDF is modelled by a Gumbel distribution. For this purpose, the values $\widehat{a}$ and $\widehat{b}$ are computed as in (2.62)-(2.63). Figure 2.34 (a) presents the resulting PDF, where it is observed that the most frequent edge-cut value coincides with the location parameter ($\widehat{a} = 6.12 \cdot 10^5$). Figure 2.34 (b) shows the empirical and theoretical CDFs, where it is observed that both fully coincide. Figure 2.34 (c) shows the success probability in a single attempt, p_s, as a function of the desired edge-cut value, Q_{max}. The rapid decay of the PDF below the location parameter leads to a fast decrease of p_s for small values of Q_{max}. As a consequence, the average number of attempts increases exponentially as Q_{max} decreases below the location parameter, as shown in Figure 2.34 (d). This behaviour explains the stagnation of naive MS approaches after a few attempts.

Once a distributional model has been found for the local minimum values, the preliminary analysis concludes with the proof of the correlation among the best solutions in the search space. This property is verified over the previous set of 1000 random local minima from an instance of the CPAP. The scatter plot in Figure 2.35 (a) shows the relationship between edge-cut and average distance to all other local minima in the sample. Again, the distance between two solutions is defined as the number of vertices that are not assigned to the same subdomain in both solutions. It is observed that the best local minima (i.e., the ones with the lowest edge-cut) have the smallest average distance to other minima. It can thus be concluded that the best local minima are central to all other local minima, since other minima are located around them. In this example, it is worth noting that the best local minimum is exactly in the centre, since it has the smallest average distance to the other minima. Figure 2.35 (b) plots edge-cut versus distance to the best local minimum. It is observed that there exists a clear correlation between the quality of the solution and the distance to the best local minima. All these facts suggest a globally-convex structure for the optimisation surface of the GPP, where the best local minima are grouped in the middle of the solution space. This property is the origin of the good performance of adaptive MS approaches.



(a) PDF

(b) CDF

(c) Success probability

(d) Avg. number of attempts

Figure 2.34: Example of performance analysis of the R-GGGP method.

Exact Methods

This section presents the results of several variants of the BC method when applied to the set of 61 CPAP instances under loose runtime constraints. These constraints cause that these methods can only be broadly classified as exact. Nonetheless, the performance of these methods will be used as a benchmark for the heuristic methods evaluated in the next section.

The first experiment highlights the difference between the three ILP models of the CPAP described in Section 2.2.3. Table 2.5 presents the problem size statistics for the 61 CPAP instances. The number of variables and constraints in each model is computed from the number of vertices, edges and subdomains on the graph, using (2.30)-(2.31), (2.37)-(2.38) and (2.51)-(2.52). The table shows both the average and maximum (i.e., worst-case) values for each parameter in the instance set. From the table, it is evident that model (GM) leads to instances

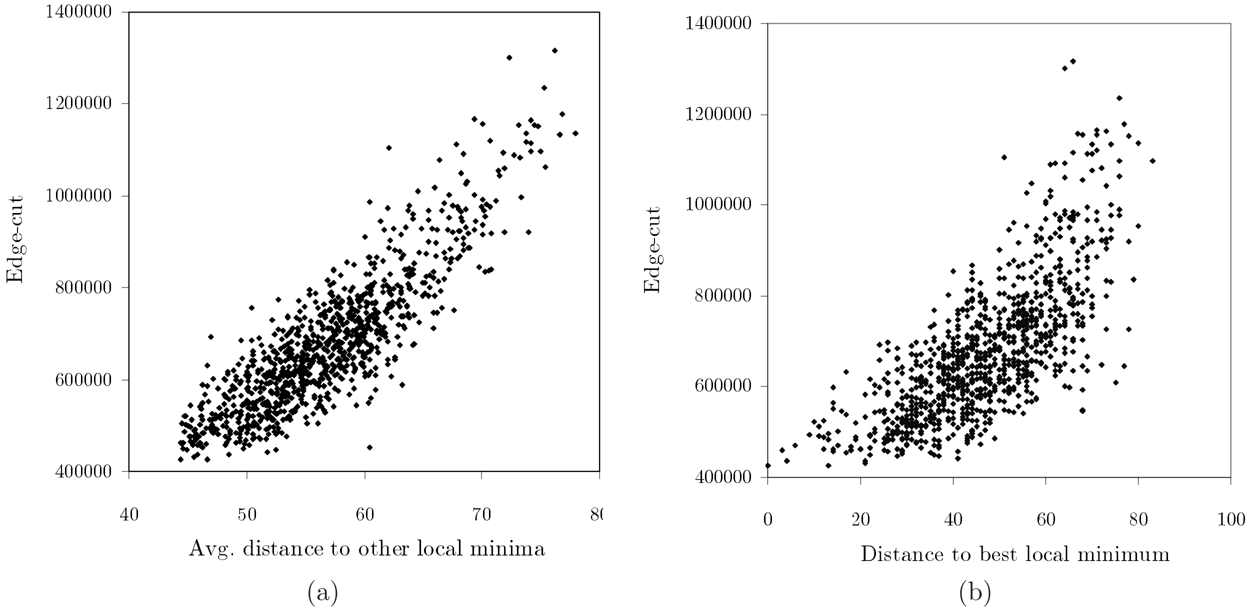


Figure 2.35: Analysis of 1000 random local minima over a CPAP instance.

Entity	Parameter	Average	Maximum
Graph	$\|V\|$	146.8	213
	$\|E\|$	929.3	1907
	k	5.34	8
ILP model (GM)	$N_{var_{GM}}$	61996	159537
	$N_{const_{GM}}$	245767	633937
ILP model (CM)	$N_{var_{CM}}$	5820	13293
	$N_{const_{CM}}$	21062	49101
ILP model (CMS)	$N_{var_{CMS}}$	5638	13048
	$N_{const_{CMS}}$	20351	48141

Table 2.5: Problem size statistics

of extreme size. In contrast, models (CM) and (CMS) reduce the size of the model by one order of magnitude. These results are due to the sparse nature of the adjacency matrix. Concretely, only 12.7 adjacencies are actively used on average for HO purposes on each cell, which is significantly less than the 146.8 cells per BSC (i.e., $|E| << |V|^2$).

From Table 2.5, it is clear that model (GM) is worse than the other models due to its large number of variables and constraints. However, the difference between models (CM) and (CMS) is more subtle. A deeper analysis proves that model (CMS) not only simplifies model (CM) marginally, but it also provides a significant performance improvement of the BC algorithm. To clarify this issue, both models are tested on the smallest problem instance. Table 2.6 presents the results of the BC algorithm over both models. The results of the original ILP formulation and the LP-relaxation of each model are given in columns ILP and LP, respectively. As previously explained, the LP-relaxation of the problem corresponds to the situation where the integrality constraints are eliminated. In the table, it is observed that both models result in a similar number of variables and constraints. Likewise, both approaches manage to solve the

	Model (*CM*)		Model (*CMS*)	
	ILP	LP	ILP	LP
Nbr. of variables	2604		2472	
Nbr. of constraints	9222		8705	
Nbr. of nodes visited	1235	-	447	-
Nbr. of iterations of simplex algorithm	90176	2926	24854	2930
Optimal edge-cut value	16151	0	16151	8524.7
Runtime [s]	229.1	2.1	56.6	2.1

Table 2.6: Performance of the Branch-and-Cut method over different models.

	IO	BC-SS	BC-ES	R-GGGP
Total edge weight [$\cdot 10^6$]	243.7			
Total edge-cut [$\cdot 10^6$]	82.3	15.3	15.0	16.4
Edge-cut ratio [%]	33.8	6.28	6.16	6.71
Avg. weight imbalance ratio	3.55	1.99	1.99	1.91
Nbr. of instances optimally proven	-	27	26	-
Runtime [h]	-	74.4	79.2	80.6

Table 2.7: Performance of different methods based on handover statistics.

ILP problem exactly and the edge-cut of both solutions coincides (i.e., 16151). However, model (*CMS*) provides a four-fold reduction in runtime (i.e., from 229 to 56s). This trend is also observed in the number of nodes visited in the search tree and, consequently, in the number of iterations of the simplex algorithm to solve the LP-relaxations. The reason for this improvement is found in the LP-relaxation of the original problem. The optimal LP value for model (*CMS*) is much closer to the one obtained for the original ILP problem (i.e., 8524.7 for model (*CMS*), 0 for model (*CM*)). Thus, the LP version of model (*CMS*) gives a finer bound for the ILP problem, which favours the discarding of nodes in the search tree. For this reason, model (*CMS*) is adopted for the rest of the analysis.

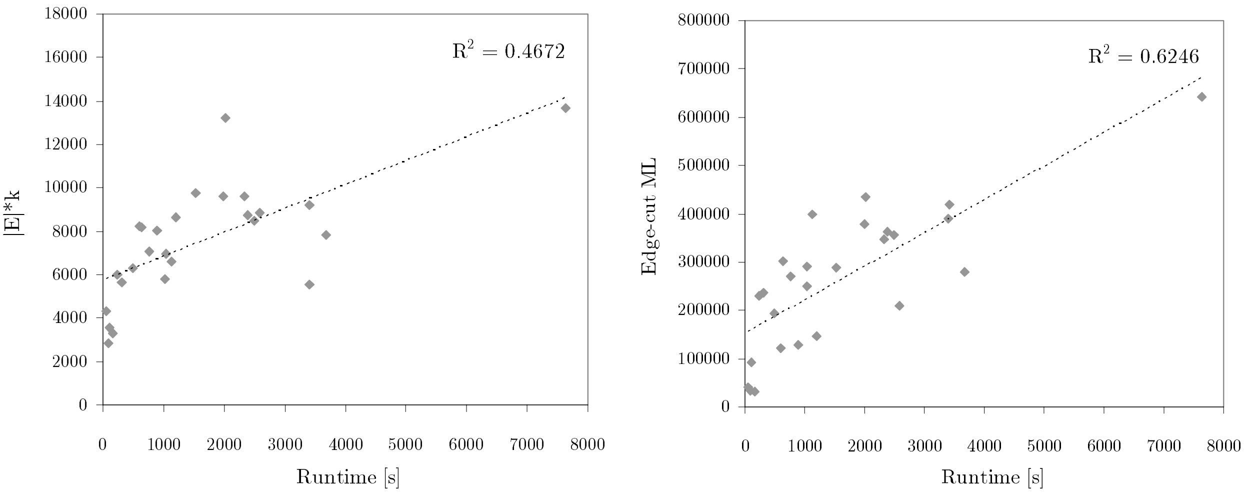
The next experiment shows the capability of exact methods to find the best solution. Table 2.7 presents the overall results of exact methods in the entire set of instances. To keep the execution time within a reasonable limit, the total runtime is set to 96 hours (i.e., $T_{ov} = 96$h). To estimate the maximum performance benefit, the current operator solution (denoted as IO) is included in the table. For comparison purposes, the result of the R-GGGP heuristic is also included in the table, as it is the simplest benchmarking approach. For a fair comparison, the number of attempts in R-GGGP is adjusted to achieve a similar execution time, and the connectedness constraint is eliminated, as exact approaches fail to consider this constraint. From the table, it is evident that the BC method gives the lowest inter-PCU HO ratio (i.e., lowest edge-cut ratio). Concretely, BC-ES reduces the total edge-cut by more than five times (i.e., 81%) when compared to IO. This result justifies the need for the optimisation process. Likewise, the total edge-cut is 9% lower than in R-GGGP, which is the heuristic method commonly used for benchmarking purposes. Not shown in the table is the fact that the heuristic approach is unable to find the optimal solution for any of the problem instances. This result reinforces the conclusion that the CPAP optimisation surface has many different local minima.

It is worth noting that the edge-cut reduction is obtained without impairing the balance between subdomains. On the contrary, the average weight imbalance ratio of the initial situation is significantly reduced. Concretely, BC-ES reduces the previous indicator from 3.55 to 1.99. At this point, it is worth noting that most optimisation methods improve the edge-cut by allowing a slight imbalance among subdomains. Therefore, the closer the weight imbalance ratio approaches to B_{rw} in the final solution, the better this flexibility is exploited. In the table, it is observed that both BC methods take full advantage of the allowed imbalance, since the average weight imbalance ratio is nearly 2. By contrast, R-GGGP displays a value of this indicator lower than 2. Experiments showed that any local-search based algorithm tends to get trapped in local-minima near the limit of the feasible region. This behaviour is a consequence of the limited number of possible vertex moves, once the current solution is close to violate the weight constraints. BC does not have this problem, as it is based on an enumerative approach.

Regarding the time-sharing strategies in BC, Table 2.7 shows that, under loose time constraints, it is advantageous from the edge-cut perspective to dedicate more time to those instances where the edge-cut tends to be higher. Thus, BC-ES slightly outperforms BC-SS, since it provides 2% less edge-cut. Concretely, BC-ES achieved less edge-cut in 11 of the 61 instances, amongst which were the 5 instances with the largest edge-cut. In contrast, BC-ES led to a higher edge-cut in only 4 instances, all of which displayed a low edge-cut figure (and, consequently, had a small influence on the total edge-cut). In the remaining 46 instances, both methods achieved the same result.

From Table 2.7, it is clear that BC does not manage to prove the optimality of all the solutions due to the time constraints. While BC-SS proves the optimality in 27 out of 61 (i.e., 44%), BC-ES only proved 26 (i.e., 43%). In BC, a solution is proved optimal only when the entire solution space has been evaluated. Thus, the presence of runtime constraints causes that, in some cases, part of the solution space remains unexplored. Nonetheless, the optimal solution is normally found in the early stages of the algorithm, provided an adequate branching strategy is used. Thus, the presence of loose time constraints should have a negligible impact on solution quality. The previous results also show that minimising the total edge-cut does not necessarily lead to the maximum number of problem instances optimally proven. On the contrary, more instances were optimally proved by BC-SS than by BC-ES.

From the runtime figures, it can be deduced that part of the available time is not used. This is a consequence of the static assignment of time to instances. Thus, 43% of the instances are solved by BC-ES before reaching their time limit and the extra time is wasted. Nonetheless, BC-ES total runtime is closer to the time available (i.e., 96h). While BC-SS does not make use of 23% of the time, BC-ES only leaves 18% unused. From this outcome, it might be inferred that the edge-cut of the ML solution, and not the instance size, provides a better estimation of runtime of BC, since the difference between the planned and actual runtime in BC-ES is less than in BC-SS. This statement is valid, at least, for the subset of problem instances that are solved exactly (i.e., the easiest instances), where the time is wasted. Although this conclusion might be considered unexpected, it is the confirmation that the complexity of a problem instance does not only depend on its size. This statement is especially true when the asymptotic assumption does not hold. To reinforce this conclusion, Figure 2.36 (a)-(b) depict the correlation of runtime with model size and edge-cut of the ML solution over the solved instances. A regression line has been superimposed, together with the squared sample correlation coefficient. In the figures, it can be observed that correlation is not strong in any case. However, the larger value of R^2 indicates that the edge-cut from a good heuristic solution provides a better estimation of time complexity in practice. It is worth noting that more refined regression models based on N_{var}



(a) Problem size

(b) Edge-cut

Figure 2.36: Correlation of BC runtime versus problem size and edge-cut of ML solution.

and N_{const} equally failed to predict the runtime of BC.

The next experiment evaluates the performance of exact approaches under stricter time constraints. For this purpose, the maximum execution time is gradually reduced from 96 to 12h. This reduction is carried out by progressively halving T_{ov} (i.e., T_{ov} = 96, 48, 24 and 12h). Figure 2.37 depicts the edge-cut ratio of both time-sharing approaches for different runtimes. The results of R-GGGP have also been superimposed for comparison purposes. It is worth noting that the x-axis in the figure represents the actual runtime, which is different from the planned T_{ov}. As stated previously, this discrepancy in exact methods is due to the early solving of some problem instances, which is more frequent for larger values of T_{ov}.

On the right side of the figure, it is evident that both BC strategies outperform R-GGGP for large values of T_{ov}, as shown in Table 2.7. The figure also shows that the benefit from additional attempts quickly diminishes in R-GGGP, even if the connectedness constraint is relaxed. From this observation, it can be concluded that BC is a better method for benchmarking purposes. At the same time, it is observed that the total edge-cut in BC methods seems to stagnate beyond 80 hours. This behaviour is worth remarking, since Table 2.7 showed that only 27 instances can be optimally proven after that period. This result reinforces the idea that, in most instances, the optimal solution has already been found and the additional runtime would only be used to prove the optimality of the solution.

On the left side of the figure, it can also be observed that both exact strategies degrade gracefully when runtime is reduced. In particular, the edge-cut for BC-ES only increases by 10% when T_{ov} decreases from 96 to 24h. Concretely, 21 out of 61 instances still maintain the same solution. For these instances, the reduction of runtime entails, at most, the loss of an optimality proof and not an impairment of the optimal value. However, when T_{ov} falls below 24h, R-GGGP becomes more efficient than BC. This result indicates that exact approaches do not behave well under severe time constraints. Thus, BC-ES is unable to find a valid solution (different from the one provided by the heuristic solution) for 2 instances when T_{ov} is set to 12h.

The overall performance difference between the two time-sharing strategies in the exact method is small under all time constraints. For values of T_{ov} above 24h, BC-ES outperforms BC-SS, while the opposite is true for values of T_{ov} below 24h. Thus, BC-SS would be the

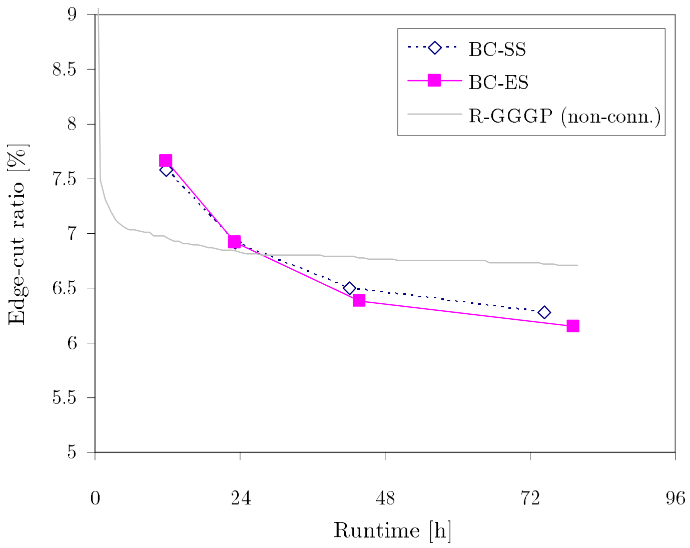


Figure 2.37: Performance of exact methods under different runtime constraints.

preferred option under strict time constraints, while BC-ES should only be used under loose time constraints. Since the latter case is common in benchmarking applications, ES might be considered as the default time-sharing strategy in the exact method.

To make the most of the available runtime, a dynamic variant of BC-ES is finally tested. In this method, time bounds are adjusted every time an excess of time is detected (i.e., every time an instance is proved to have been solved exactly). After such an event, the share of the remaining instances is updated, taking into account the remaining time. To maximise the benefit in terms of edge-cut, the instances are first ranked based on the edge-cut of the ML solution and later solved in increasing order. This procedure enforces that those instances with a higher edge-cut receive the excess of time of the other instances (and not conversely). The full exploitation of T_{ov} (i.e., 96h) helps to decrease the edge-cut by only 2% in relative terms. Likewise, it is obvious that the benefit from dynamic approaches will be negligible under tight time constraints, since few of the instances are solved exactly in that case. Concretely, only 3% of the time is left unused by static approaches when T_{ov} is reduced to 24h. Since the benefits of this dynamic variant of BC proves limited, this method is discarded for the rest of the analysis.

Heuristic Methods

The previous experiments proved that exact methods can solve the CPAP under loose runtime constraints. However, heuristic methods are usually preferred by network operators for the sake of time efficiency. The rest of the analysis is devoted to the comparison of heuristic methods for the CPAP. The following explanation have been structured to provide the reader with a smooth introduction to the results, from the simplest to the most complex. The analysis begins with the comparison of overall performance indicators from the methods with standard parameter settings. The tradeoff between edge-cut and runtime in the methods is investigated later. Having identified the potential of ML approaches, the subsequent analysis focuses on these methods, and, in particular, on ML-CAMS. The study concludes with the analysis of the influence of problem constraints on the performance of the most relevant methods.

a) Overall Performance Comparison

The analysis is first focused on solutions with cell resolution and connectedness constraint. Table 2.8 breaks down the main overall performance indicators achieved by the PCU plans created by different heuristic methods. For comparison purposes, the performance of the current operator solution (i.e., IO) is included in the first column. Likewise, the sum of edge weight in the area is reflected on the third row, which corresponds to the edge-cut when every cell has its own PCU (i.e., worst-case).

The analysis results corroborate the trends suggested by the field trial. From the table, it is clear that all methods achieve a significant edge-cut reduction, when compared to the initial operator solution (i.e., IO). In particular, results show that the edge-cut ratio can be reduced by the best heuristic methods (i.e., ML-CAMS and R-GGGP) from 33.8% (=82.3/243.7) to 7.3% (=17.7/243.7) (i.e., five-fold reduction). This result highlights the bad quality of the manual solution. Although all methods share the same refinement algorithm, results show that it is beneficial to start from scratch with an initial partition where the load is well distributed among subdomains, which is done in all methods except RO. The large difference in edge-cut reduction ratio between RO and ML-CAMS or R-GGGP solutions (i.e., 78.5-54.7=23.8% absolute) proves this statement. Among the methods that partition the graph from scratch, ML-CAMS and R-GGGP show the lowest edge-cut ratio. It is worth noting that, although ML-CAMS shows similar performance on average to the best solution found by R-GGGP, some individual ML-CAMS attempts outperformed R-GGGP. Likewise, ML-CAMS solution has 10% less edge-cut ratio than the traditional ML method. From these results, it can be concluded that the edge-cut performance of ML-CAMS is similar to the best method a priori (i.e., R-GGGP).

At the same time, the weight of subdomains is more evenly balanced in any of the new solutions. This re-distribution of weight translates into a reduction of the average weight imbalance ratio. For ML-CAMS, the imbalance ratio is nearly halved (i.e., 3.55/1.91=1.86) when compared to the existing solution. In the table, it is observed that ML, CAMS and ML-CAMS also exploit the allowed imbalance, as the weight imbalance ratio is close to (but less than) 2. Not shown in the table is the fact that some methods are unable to ensure the imbalance constraint without producing disconnected subdomains in some BSCs (concretely, 3 in RO and 1 in ML). This is the reason why the average imbalance for these methods is so high.

Although a slight imbalance is present in most solutions, it is observed that enough spare capacity is still available in all subdomains to deal with future network growth. Concretely, the average maximum subdomain weight (i.e., the maximum number of TSLs per PCU) in the ML-CAMS solutions is still only 37% (i.e., 100·94.1/256)) of B_{aw} (i.e., 256 TSLs limit). From this data, it can be inferred that the network is overdimensioned, since the average load of the PCUs is well below its capacity limit.

Similarly, the number of disconnected subdomains is greatly reduced by most methods. This outcome stems from the sequence of connectedness checks within the refinement algorithm. Nonetheless, it is observed that the reduction is not large when the refinement is directly applied to the initial solution (i.e., the number in IO is only halved by RO). Thus, it is beneficial to start with a good initial partition. Concretely, ML-CAMS achieves a thirteen-fold reduction of the total number of disconnected subdomains from the initial solution. Nonetheless, in some isolated cases, the algorithm is still unable to correct the lack of connection in some subdomains, due to the fact that the original graph is disconnected. This situation takes place when the number of isolated cell clusters exceeds the number of PCUs in a BSC.

Heuristic method	IO	RO	FW-GGGP	ML	CAMS	ML-CAMS	R-GGGP
Assignment granularity	Cell						
Sum of edge weights [$\cdot 10^6$]	243.7						
Total edge-cut [$\cdot 10^6$]	82.3	37.3	29.6	19.3	18.0	17.7	17.7
Edge-cut ratio [%]	33.8	15.3	12.1	7.9	7.4	7.3	7.3
Normalised edge-cut	4.65	2.11	1.68	1.10	1.02	1.00	1
Total edge-cut reduction ratio [%]	-	54.7	64.0	76.5	78.1	78.5	78.5
Average maximum subdomain weight	91.4	93.2	90.5	90.9	92.1	94.1	91.5
Average weight imbalance ratio	3.55	2.66	1.88	2.00	1.90	1.91	1.82
Total nbr. of disconnected subd.	192	90	37	12	13.0	13.4	29
Total nbr. of changes	-	2870	4924	4536	4564	4550	4631
Runtime [s]	-	231	261	200	1703	455	74110

Table 2.8: Overall performance of different graph partitioning methods based on handover statistics.

Since the gain of the methods is dependent on the problem instance, it is interesting to check that the performance is consistent across the entire set of instances. The robustness of the different methods is analysed by breaking down the results achieved in the area on a BSC level. Table 2.9 presents the main statistical averages of the previous performance indicators on a per-BSC basis. The average, standard deviation and worst-case values are presented for the considered methods. From the table, it can be deduced that ML-CAMS not only gives the best average performance, but also provides the most stable results. No less remarkable is the minimum edge-cut reduction value of 61% attained by this method.

As an outcome of the PCU re-planning procedure, a large number of changes are suggested by those strategies that build the initial partition from scratch. For instance, Table 2.8 shows that 4550 out of the 8952 cells must be reallocated in the ML-CAMS solution. Although this figure might seem excessive, when compared to the number of cells in the trial area, it should be pointed out that a significant part of these must be performed to counteract the uneven PCU load in the initial solution. Concretely, 683 changes (i.e., 15%) are performed by the refinement algorithm to find a valid solution before starting the actual refinement stage. This result also points out the sub-optimality of the initial solution.

From the runtime perspective, Table 2.8 shows that several methods have execution times in the order of minutes for the whole area. Among them, the standard ML method proves the most efficient alternative. It is noticeable that ML builds a solution with half the edge-cut of the RO solution in less time. In contrast, CAMS and R-GGGP have a large execution time, making them less appealing for daily use. Despite the superiority of ML in terms of runtime, it is observed that the difference is not as large as in other applications. This is mainly due to the small size of CPAP graphs. Thus, the number of coarsening steps required to reduce the number of vertices up to the number of subdomains k is small. By assuming that the number of vertices is halved after each coarsening operation, a rough estimation of the average number of coarsening levels per graph, $\overline{N_l}_{ML}$, can be computed as

$$\overline{N_l}_{ML} \approx \log_2 \frac{E[|V|]}{E[k]} = \log_2 \frac{146.8}{5.43} = 4.76 \, , \tag{2.68}$$

where $E[|V|]$ and $E[k]$ are the average number of vertices and subdomains, presented in Table 2.5. In practice, the value of $\overline{N_l}_{ML}$ is somewhat larger, since the capability to simplify the graph degrades for the last coarsening levels. This trend is corroborated in Figure 2.38, which shows a scatter plot of the size reduction factor achieved by SHEM over different graph sizes. From the trend line, it is clear that the ratio between the size of consecutive versions of the graphs is always lower than 2, regardless of the graph size. This loss of efficiency of the coarsening algorithm is more pronounced for smaller graphs, which correspond to the last coarsening steps. As a result, the actual value of $\overline{N_l}_{ML}$ coarsening steps is 8.4 (almost twice the value in (2.68)). Nonetheless, most of the coarsening effect is concentrated on a reduced number of steps and this justifies the short runtime difference against non-hierarchical heuristic methods.

b) Edgecut-Runtime Tradeoff

The discussion now focuses on the edgecut-runtime trade-off in the heuristic methods, which is the core of the analysis. For this purpose, the intensity of methods is varied to show how the additional runtime is used to improve solution quality. While the intensity of single-run methods

Heuristic method	IO	RO	FW-GGGP	ML	CAMS	ML-CAMS	R-GGGP
Assignment granularity	Cell						
Avg (edge-cut reduction ratio) [%]	-	55.7	63.0	77.6	79.0	79.3	79.7
Std (edge-cut reduction ratio) [%]	-	16.4	13.5	9.6	8.7	8.6	9.6
Min (edge-cut reduction ratio) [%]	-	15.9	20.4	55.0	59.4	60.9	59.4
Avg (weight imbalance ratio)	3.55	2.66	1.87	2.00	1.90	1.91	1.83
Std (weight imbalance ratio)	6.47	5.45	0.32	0.99	0.14	0.09	0.19
Max (weight imbalance ratio)	49.5	44.5	3.78	9.5	2	2.04	2
Avg (nbr. of disconnected subdomains)	3.15	1.47	0.61	0.20	0.21	0.22	0.48
Std (nbr. of disconnected subdomains)	1.35	1.19	1.07	0.44	0.52	0.29	0.62
Max (nbr. of disconnected subdomains)	6	5	4	2	3	2	2

Table 2.9: Statistical averages of main performance indicators on the area in a BSC level.

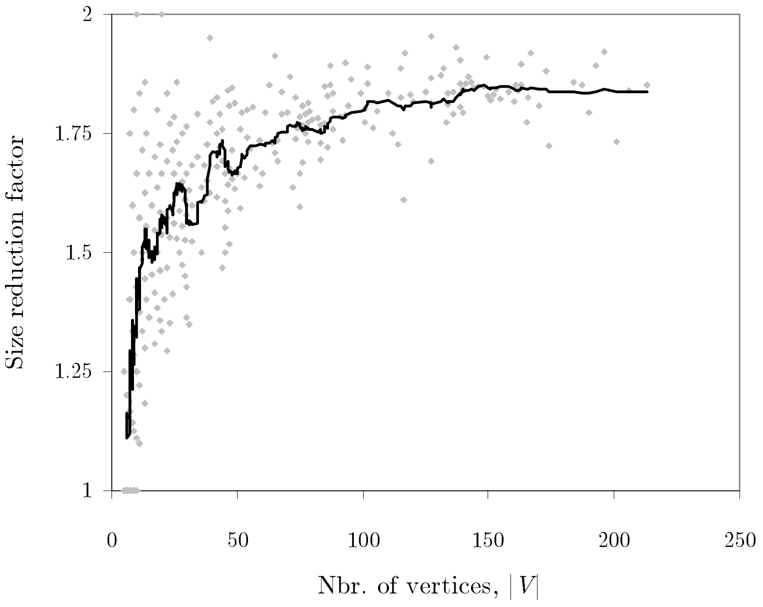


Figure 2.38: A scatter plot of size reduction factor versus graph size.

(i.e., RO, FW-GGGP, ML) is varied through the number of passes in the FM algorithm, the intensity of MS methods (i.e., CAMS and R-GGGP) is controlled by the number of attempts. As ML-CAMS is a hybrid approach, the intensity of this method is controlled by the adjustment of both parameters simultaneously. Performance results are aggregated across instances and later normalised against the reference values (i.e., the total edge-cut of the best of 1000 R-GGGP attempts and the total runtime of the standard ML algorithm with no refinement).

Figure 2.39 shows the convergence of each method to the best solution over time. Each curve corresponds to a different method and each point in the curve represents the runtime and edge-cut of a certain combination of algorithm and intensity. For single-run methods (i.e., RO, FW-GGGP and, ML), six points are presented on each curve, corresponding to the following settings in the refinement algorithm: no refinement, greedy refinement and FM refinement with 1, 2, 3 and 4 passes. Only five points are visible in some curves, as the performance of greedy refinement actually coincides with that of 1 pass of FM refinement in most methods. For MS methods (R-GGGP, CAMS and ML-CAMS), each point represents the edge-cut of the best attempt carried out so far. While each point in the R-GGGP curve denotes a new attempt, each point in CAMS corresponds to a new generation of solutions. In ML-CAMS, each point corresponds to a combination of number of generations in CAMS (initial partitioning) and passes in FM (uncoarsening). While the former parameter ranges from 0 to 5, the latter ranges from no refinement to 4 passes. Figure 2.40 extends the time axis to show the performance of the most computationally expensive methods.

Figure 2.39 confirms the results presented in Table 2.8. Thus, it is evident that the refinement of the existing solution (i.e., RO) performs worse than any other method that builds the partition from scratch. ML proves to be the best option for quick solutions, while R-GGGP, CAMS and ML-CAMS provide very high-quality solutions, which can be used for benchmarking purposes. It is also observed that single-run algorithms (i.e., RO, FW-GGGP and ML) converge quickly to their best solution. In these methods, most of the improvement in solution quality is achieved in the first pass of the refinement algorithm. Thus, greedy refinement performs competitively with FM refinement over this type of graphs. In contrast, MS algorithms (i.e., R-GGGP, CAMS

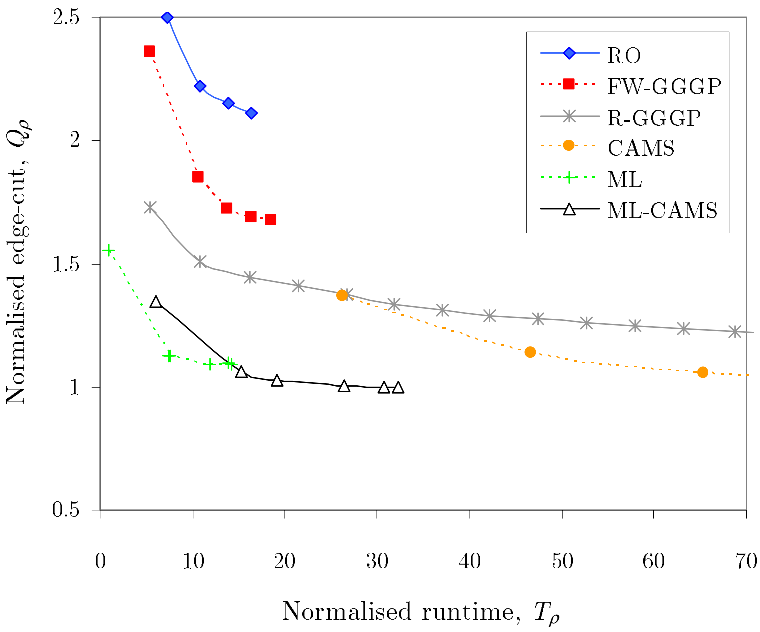


Figure 2.39: Convergence behaviour of different heuristic methods.

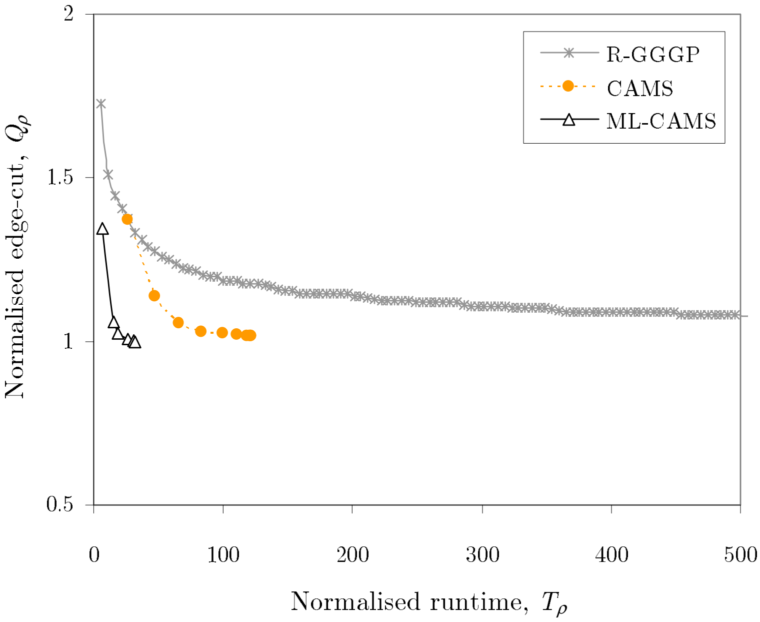


Figure 2.40: Convergence behaviour of multi-start heuristic methods.

and ML-CAMS) show slower convergence. Figure 2.40 shows clearly that R-GGGP stagnates after a few attempts, despite giving the best solution in the limit (i.e., the normalised edge-cut reaches the value of 1 in the limit, since it is the reference value against which all methods are normalised). At the same time, the first point in CAMS curve coincides with the fifth point in R-GGGP curve, as the first generation of solutions in CAMS consists of five solutions built by R-GGGP. Thereafter, the identification of similarities greatly enhances the convergence speed. Finally, it is observed that ML-CAMS brings forward the benefits from adaptive MS techniques, giving quick solutions whose quality is close to (or sometimes above) those achieved by R-GGGP in the limit.

Heuristic method	ML(standard)		ML(early-stop)		ML-CAMS	
Assignment granularity	Cell					
Matching technique	SHEM	SM-SHEM	SHEM	SM-SHEM	SHEM	SM-SHEM
Sum of edge weights [$\cdot 10^6$]	243.7					
Total edge-cut [$\cdot 10^6$]	19.3	20.4	24.3	25.7	17.7	18.7
Edge-cut ratio [%]	7.9	8.4	10.0	10.5	7.3	7.7
Runtime [s]	200	211	189	214	455	517

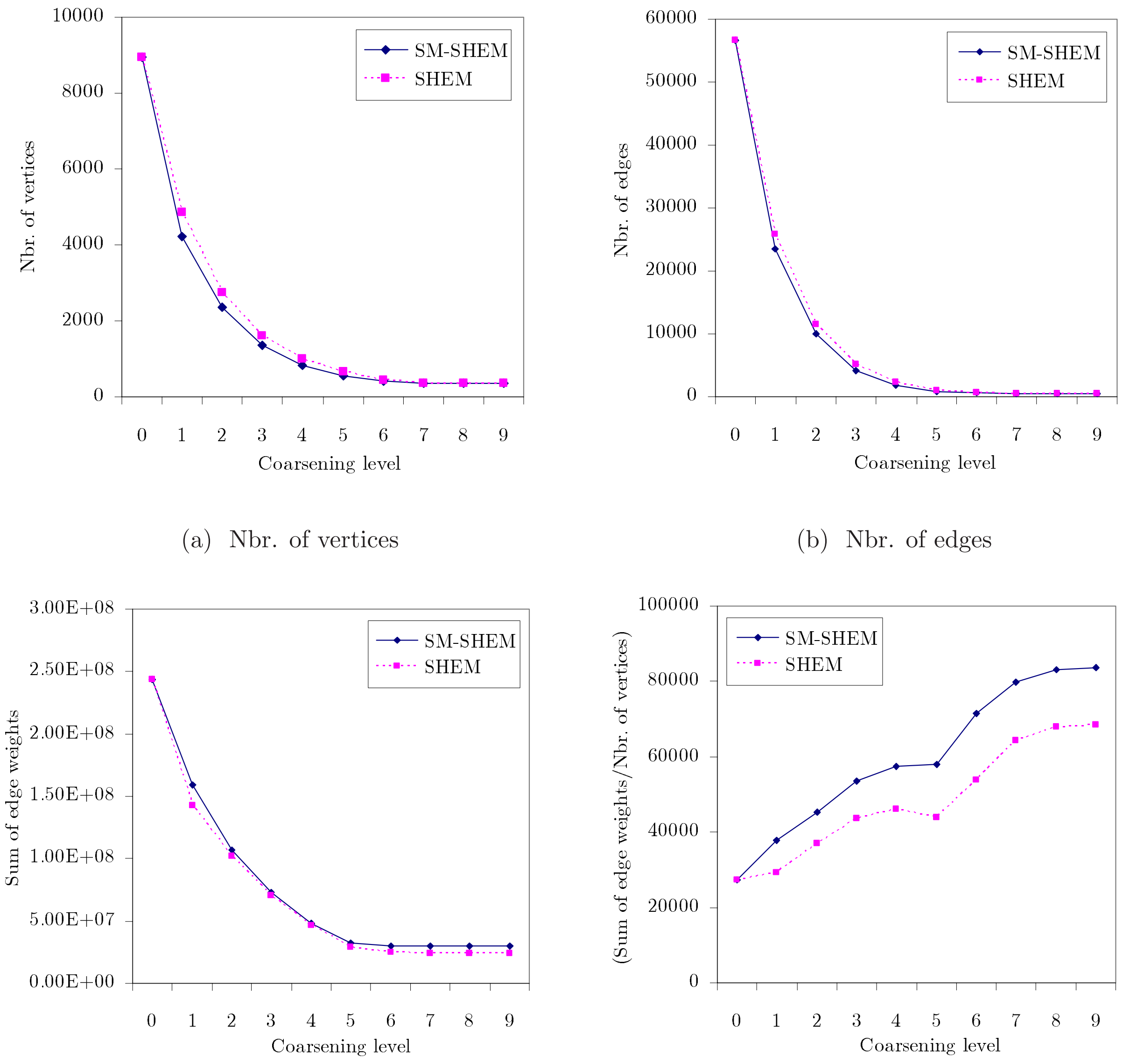
Table 2.10: Performance comparison of different matching schemes.

c) Performance of Multi-Level Methods

The following analysis compares the performance of several ML methods. On the one hand, it is interesting to evaluate the impact of terminating the coarsening stage prematurely, as it is done in the early-stop ML method. For this case, two initial partitioning algorithms are considered: FW-GGGP and CAMS. The resulting methods are denoted as ML(early-stop) and ML-CAMS, respectively, to differentiate them from the standard ML algorithm, denoted as ML. On the other hand, it is interesting to compare the performance of the two coarsening algorithms proposed in this work: SHEM and SM-SHEM.

Table 2.10 presents the results of the different ML methods with SHEM and SM-SHEM coarsening schemes. It is worth noting that this experiment does not intend to compare solutions with cell or site granularity, but only the use of SM or SHEM in the first coarsening step. Therefore, performance figures correspond to solutions that still maintain the cell granularity. From the table, it can be concluded that terminating the coarsening process earlier is detrimental for the edge-cut of the final solution. For instance, the edge-cut ratio of ML(early-stop) with SHEM suffers an absolute increase of 2.1% (i.e., 10.0-7.9%) when compared to ML. This impairment from the early termination of coarsening is counteracted in ML-CAMS by multiple trials. Thus, ML-CAMS reduces the edge-cut ratio of ML(early-stop) by 2.7% in absolute terms. Regarding the coarsening scheme, it is shown that the use of SM-SHEM slightly deteriorates the edge-cut of the final solution. Concretely, the use of SM-SHEM in ML-CAMS results in an absolute increase of the edge-cut ratio of 0.4%. Similar results are obtained in the other methods. More unexpectedly, the runtime of SM-SHEM is also larger. From the previous result, it can be concluded that SHEM clearly outperforms SM-SHEM.

A more detailed comparison of the performance of both coarsening algorithms gives evidence of the superiority of SHEM over SM-SHEM. Figure 2.41 (a)-(d) show the evolution of some relevant indicators in ML with SHEM and SM-SHEM schemes. Figure 2.41 (a) depicts the total number of vertices as the coarsening stage progresses. In this figure, it is observed that this indicator decreases by a factor of 2.1 (i.e., 8952 cells/4216 sites=2.1 cells/site) when SM is applied in the first coarsening step, while it only reduces by 1.8 for SHEM. Therefore, it is evident that SM achieves a faster coarsening of the graph. This result is mainly due to the larger number of vertices per matched group in SM, i.e., while up to 6 cells are grouped into a site, only 2 are grouped in the case of paired matching. In subsequent steps, the repeated use of SHEM in both schemes tends to reduce the difference. Finally, the size of the graph stagnates after a few coarsening steps around the same value in both schemes. This outcome is not to be considered as an indicator of poor matching performance, but only a consequence of the



(a) Nbr. of vertices

(b) Nbr. of edges

(c) Total edge weight

(d) Total edge weight per vertex

Figure 2.41: Performance of different coarsening algorithms.

inverse power-of-two law (i.e., $|V^{(n)}| \propto 2^{-n}$, where n is the coarsening level). The same trend is observed in the number of edges and total edge weight shown in Figure 2.41(b)-(c). From the latter figures, it might wrongly be concluded that SM has a superior performance, since it is able to reduce the total graph edge-weight faster. On the contrary, it is worth remarking that the best matching method in terms of edge-cut is not the one that achieves the highest reduction per coarsening step, but the one with the highest reduction per matched vertex. While the former approach aims to minimise the number of coarsening stages, the latter approach aims to minimise the number of vertices wrongly forced to be in the same subdomain by an improper matching. By taking the right matching option, SHEM hides a larger share of the total edge weight at the end of the coarsening process, although more coarsening stages are needed. This effect is observed in the evolution of the total edge weight per vertex presented in Figure 2.41 (d). The exposed edge weight per vertex with SM in the first coarsening step is higher than that after SHEM. The difference is maintained for the rest of the coarsening process. This result reinforces the conclusion that SHEM performs better, since it is able to hide a larger edge weight per matched vertex. Intuitively, this outcome is justified from the fact that the best matching is achieved between cells that have a strong HO relationship, which is not necessarily the case among cells in the same site.

From the runtime perspective, it is still tempting to conclude that SM-SHEM is faster than SHEM, since it has less coarsening/refinement steps. On the contrary, it is shown that wrong matching decisions performed by SM in the first coarsening step must be corrected by the refinement algorithm during the last uncoarsening step. The increased number of steps and passes in the refinement algorithm over the original graph is the reason for the higher runtime of SM-SHEM observed in Table 2.10. From these results, it can be concluded that SM should only be used to achieve solutions with site granularity.

Although the previous conclusions were drawn for ML, they are equally valid for ML-CAMS, as most of the coarsening process is shared by both methods. However, the influence on ML-CAMS is expected to be stronger, as the number of coarsening steps is less due to the early stop of the coarsening process. By considering that coarsening stops when the number of vertices per subdomain is less than a threshold c, the average number of coarsening levels per graph can be roughly approximated by

$$\overline{N_l}_{ML-CAMS} \approx \log_2 \frac{E[|V|]}{c \cdot E[k]} = \log_2 \frac{146.8}{15 \cdot 5.43} = 0.85 \;, \tag{2.69}$$

where again $E[|V|]$ and $E[k]$ are the average number of vertices and subdomains, and c is the minimum average number of vertices per subdomain in the coarsest graph (c=15 in this work). The previous formula suggests that ML-CAMS only performs one coarsening step on average in CPAP graphs. Figure 2.42 confirms this trend by presenting the histogram of the number of coarsening levels in ML and ML-CAMS. Although the actual average number of coarsening levels for ML-CAMS is somewhat larger than predicted (i.e., 2.55), no CPAP instance displayed more than 3 coarsening levels. Nonetheless, it is worth noting that, despite the small number of coarsening levels, graph coarsening still gives a significant speed benefit to CAMS. As shown in Table 2.8, CAMS runtime decreases from 1703 to 455s (i.e., four-fold reduction) when combined with an ML technique (as in ML-CAMS).

d) Performance of ML-CAMS

The good performance of adaptive MS methods (i.e., CAMS and ML-CAMS) stems from the "big valley" structure exhibited by the local minima of the GPP. However, it is worth noting that ML-CAMS not only outperforms CAMS in terms of runtime, but also in terms of solution quality. The origin of this performance enhancement is the smoothing of the optimisation surface achieved by the coarsening process. To verify this property, the distribution of local minima in a single problem instance is studied following the methodology suggested in [52].

Figure 2.43 (a)-(b) present the analysis of 1000 random local minima obtained by R-GGGP in an individual CPAP instance. These figures show again the correlation between edge-cut and distance to other local minima. Figure 2.43 (a) and (b) differ in the coarsening level where the method is applied: while the former presents the results for the original graph, the latter does the same for a coarsened version of the graph. In both cases, it is observed that the best local

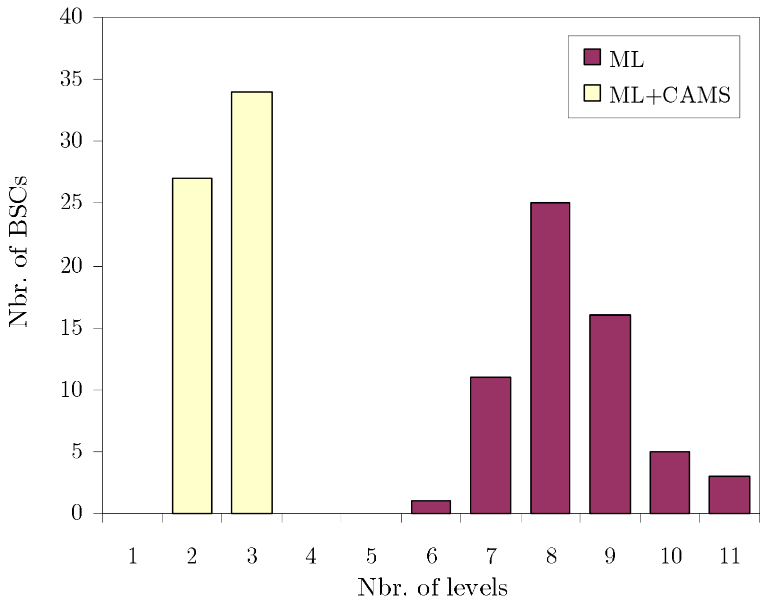
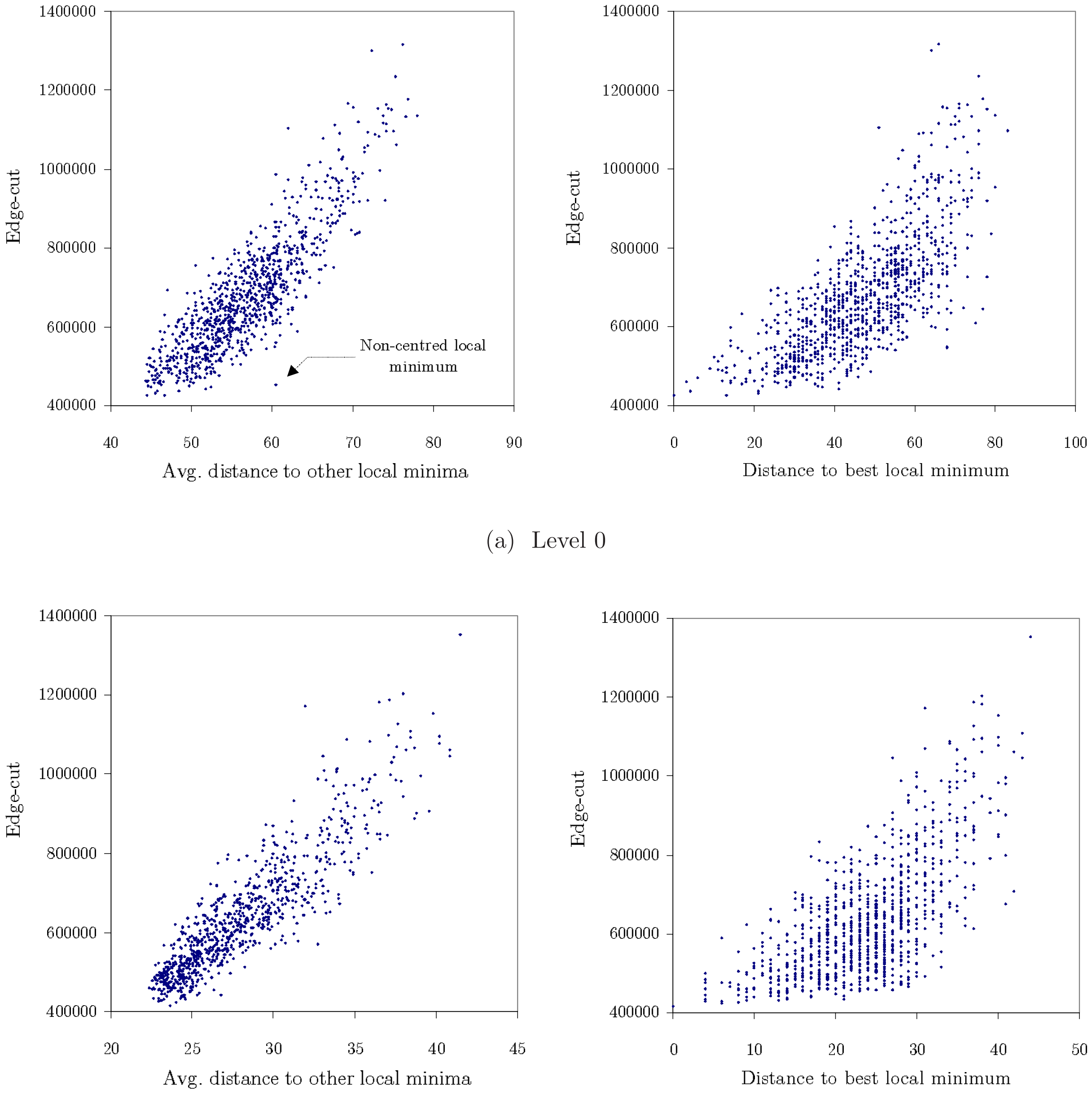


Figure 2.42: Histogram of the number of coarsening levels in multi-level algorithms.

minima (i.e., smallest edge-cut) show smaller average distance to other local minima. Thus, the best local minima tend to be central to all other local minima. This result indicates that the best solutions share common components, which is the basis of CAMS. However, in the original graph, the best 20 local minima are not fully clustered. On the contrary, Figure 2.43 (a) (left) shows that one local minimum with edge-cut close to the minimum value has an average distance of 60, which is much larger than the minimum one of 45. In contrast, the best local minima in Figure 2.43 (b) (left) are better grouped, which is evident from the narrower scatter plot. These figures also show that, after coarsening, most random local minima are clustered in the low edge-cut region. By hiding edges with large weights, coarsening the graph filters out the worst local minima, which improves the quality of R-GGGP solutions. In parallel, coarsening reduces the distance between local minima, due to the reduction of the number of vertices in the graph (note the different x-axis scale in Figure 2.43 (a) and (b)). The smaller distance between solutions leads to a more efficient refinement process, which is key to the runtime efficiency of ML-CAMS. All these facts justify the superiority of ML-CAMS over CAMS both in terms of solution quality and runtime.

Figure 2.43 (b) (left) also helps to explain a more subtle property of adaptive MS methods. In the figure, it is shown that the best local minimum is not necessarily in the centre of the solution space. In this case, the best minimum is not the one with the minimum average distance to other local minima (i.e., there exist other minima to the left of the lowest one in the figure). This result suggests that the best minimum is displaced from the centre of gravity of the set of local minima (otherwise, it would display the minimum average distance to all other minima). Thus, an algorithm that built a new partition by minimising the average distance to a previous solution set would fail to find the best solution if the latter was not completely centred. In contrast, CAMS does not intend to minimise the average distance to previous solutions, but is only guided by similarities among them. Thus, new solutions built by CAMS from an old set of solutions have lower average distance to the old solutions (i.e., they are interior points of the old set), but do not necessarily have the minimum distance (i.e., they might not be in the geometrical centre).



(a) Level 0

(b) Level 2

Figure 2.43: Analysis of 1000 local minima built by R-GGGP.

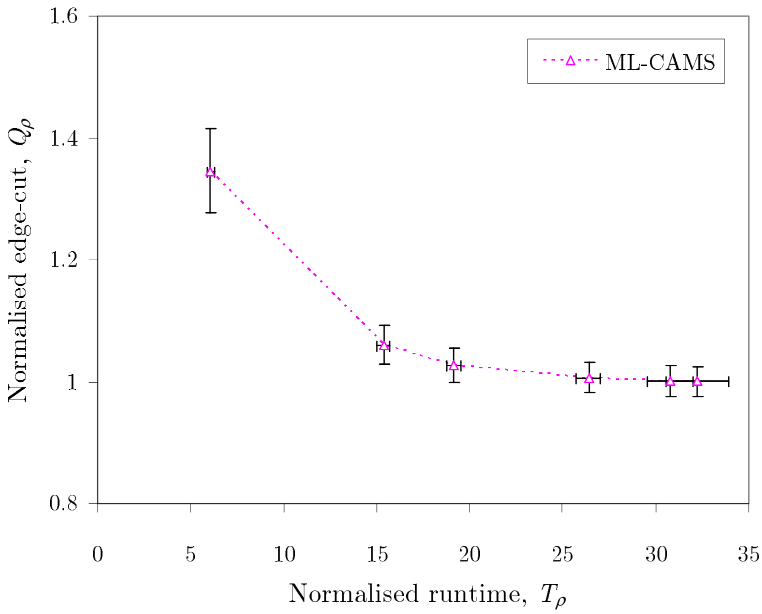


Figure 2.44: Plot of convergence of ML-CAMS with 99% confidence intervals.

So far, only performance averages have been taken into account. In a live situation, only a limited number of attempts can be performed for the sake of efficiency. Hence, it is important to verify the variability of ML-CAMS results. Thus, it is possible to check whether one attempt is enough to get reliable results, or, on the contrary, more attempts are needed. Figure 2.44 isolates the convergence curve of ML-CAMS, presented in Figure 2.39. For each setting in ML-CAMS, the 99% confidence intervals are computed from the standard deviation of performance figures from 1000 independent runs. The results are superimposed on each point of the curve. It is clear that the method provides consistent results across different attempts, since small confidence intervals are obtained. As expected, the edge-cut variability decreases as intensity increases, while the runtime variability increases due to the increased number of operations. Concretely, the width of the confidence interval for the normalised edge-cut in ML-CAMS with 5 generations in CAMS and 4 passes of FM refinement is only 4.8% in absolute terms. From this observation, it can be concluded that any ML-CAMS attempt has a small deviation from the average case, and, consequently, one attempt is enough to achieve a good solution. This good result is mainly due to the construction of a wide enough set of independent solutions in the initial generation (i.e., g=5). It is also worth noting that the latter confidence interval is centred around the reference value (i.e., 1) in the limit. This result gives evidence that, although the average performance of ML-CAMS is similar to the best of 1000 R-GGGP attempts, almost half of the ML-CAMS attempts end up with a solution better than R-GGGP.

Although the theoretical worst-case complexity for most algorithms in ML-CAMS was already stated, it is still important to evaluate runtime in practice. Unlike time complexity, runtime not only depends on problem size but also on the particular instance. Figure 2.45 depicts a scatter plot of the runtime of ML-CAMS against the number of edges in the 61 CPAP instances. In the figure, no strong relationship is observed between runtime and graph size. Nonetheless, the regression line suggests that the total runtime of the implemented algorithm is linear with $|E|$, which is far below the worst-case theoretical limit of $O(|V|^2(|V| + |E|))$. This result is just a consequence of the small size of CPAP graphs, for which the asymptotical assumption does not hold.

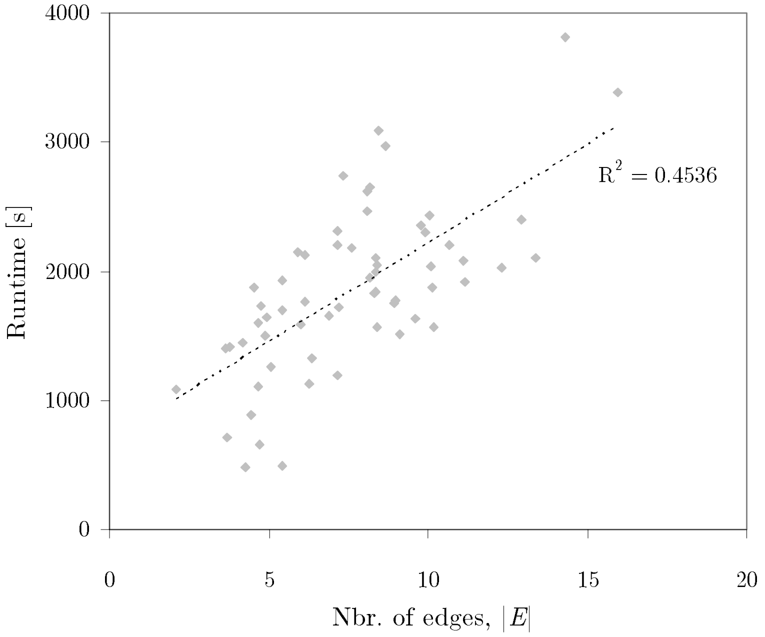


Figure 2.45: A scatter plot of runtime versus graph size in ML-CAMS.

The assessment of ML-CAMS concludes by evaluating the sensitivity to internal parameter changes. The focus is on the parameters that control the intensity and diversity of the search: the maximum number of generations and the number of solutions per generation. The final aim of the analysis is to find the best settings for these parameters. For simplicity, the following results correspond to the application of the algorithm to a single problem instance, although similar trends are observed for other problem instances.

To gain some insight into the influence of the maximum number of generations on solution quality, the first experiment shows how the solution quality improves across generations. For this purpose, 1000 ML-CAMS attempts are performed on a single problem instance and the quality of solutions on each generation is evaluated. For this experiment, no restriction is imposed on the number of generations (i.e., the maximum number of generations is set to ∞). Figure 2.46 illustrates the ECDF of edge-cut in the solutions of each generation. In the figure, it is observed that most of the edge-cut improvement takes place in the construction of the second generation of solutions (i.e., first generation after detecting similarities). Subsequent generations provide only marginal improvement, mainly from the elimination of the worst-quality individuals in the solution set (i.e., solutions to the right of the x-axis). This result proves that a small benefit is obtained after the first generation. Although it might be tempting to limit the maximum number of generations to increase the efficiency of the method, Figure 2.47 proves that this is not necessary. Figure 2.47 depicts the histogram of the number of generations in the 1000 ML-CAMS attempts, which can range from 1 to 10. From the frequency values, it is observed that 650 out of 1000 attempts had only one generation apart from the initial one built by R-GGGP. From the ECDF values in the secondary axis, it is also evident that 95% of the attempts had 3 or less generations. The reason for this early stop is the inability to generate a different solution set after 2-3 generations, since no additional similarities in the solution set tend to appear beyond that point.

Finally, the analysis evaluates the influence of the number of solutions per generation, g, on solution quality. For this purpose, 1000 independent runs of ML-CAMS are carried out

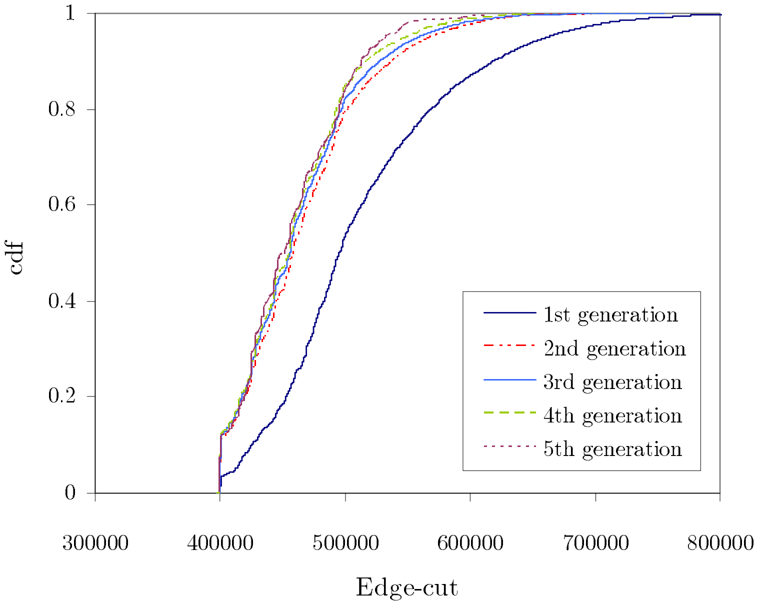


Figure 2.46: Evolution of edge-cut across generations in ML-CAMS.

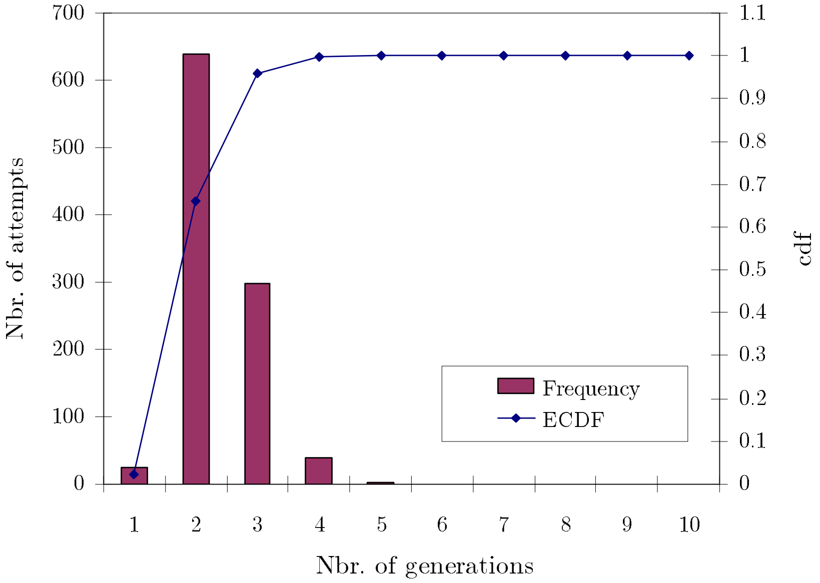


Figure 2.47: Histogram of the number of generations in ML-CAMS.

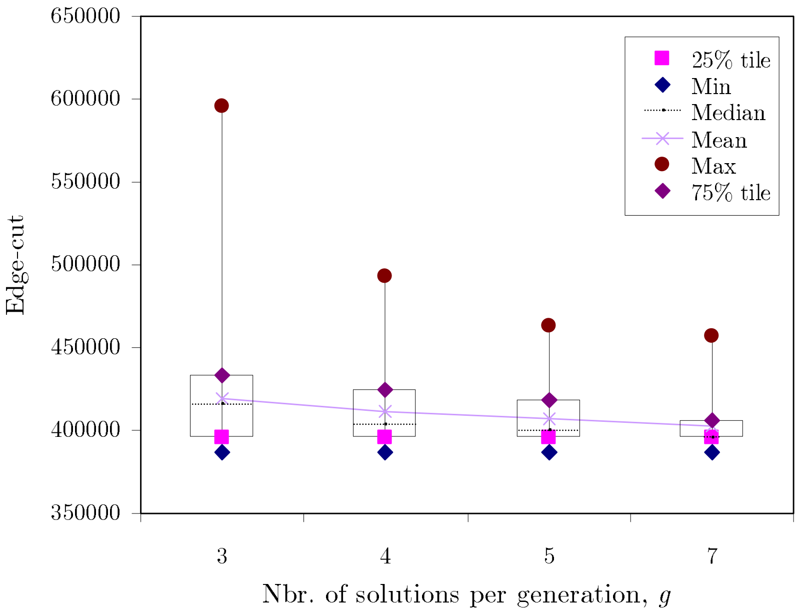


Figure 2.48: A box plot of the edge-cut distribution for different number of solutions per generation in ML-CAMS.

for different values of this parameter. Figure 2.48 depicts a box plot to show the effect of this parameter on edge-cut performance. A box plot [86] provides a visual summary of many important aspects of a distribution. The box stretches from the lower hinge (defined as the 25th percentile) to the upper hinge (the 75th percentile) and therefore contains the middle half of the values in the distribution. The median is shown as a dotted line across the box, while the mean and the extreme values are represented by different symbols. Thus, the plot includes a measure of central location (i.e., the median), two measures of dispersion (i.e., the range and inter-quartile range) and the skewness (i.e., from the orientation of the median relative to the quartiles). A trend line has also been superimposed to highlight the influence of g on the average edge-cut performance. In the figure, it is observed that the performance of the different parameter settings mainly differ in the higher edge-cut region, especially in the maximum (i.e., worst-case) value. Results show that ML-CAMS exhibit a large edge-cut variability for g=3, which confirms that three solutions per generation do not provide enough diversity. In contrast, setting g=5 largely reduces the worst-case value without an excessive increase in runtime. Beyond that value, the benefit from an increased diversity is significantly reduced and does not compensate for the increased runtime. Although it might be expected that runtime grew linear in g, experiments prove that the growth is significantly larger due to an increase of the number of generations in the algorithm. The histogram presented in Figure 2.49 shows that the number of generations increases with the number of individuals per generation. This result is mainly due to the difficulty of finding similarities in a large set of solutions. As more solutions are refined during an increased number of generations, the convergence to the final solution is much slower.

e) Influence of Optimisation Constraints

The analysis is completed with the influence of constraints on the optimisation process. From the operator side, it is essential to know the sensitivity of solution quality to the variation of constraints. As the number of trials is limited for practical reasons, the constraints cannot

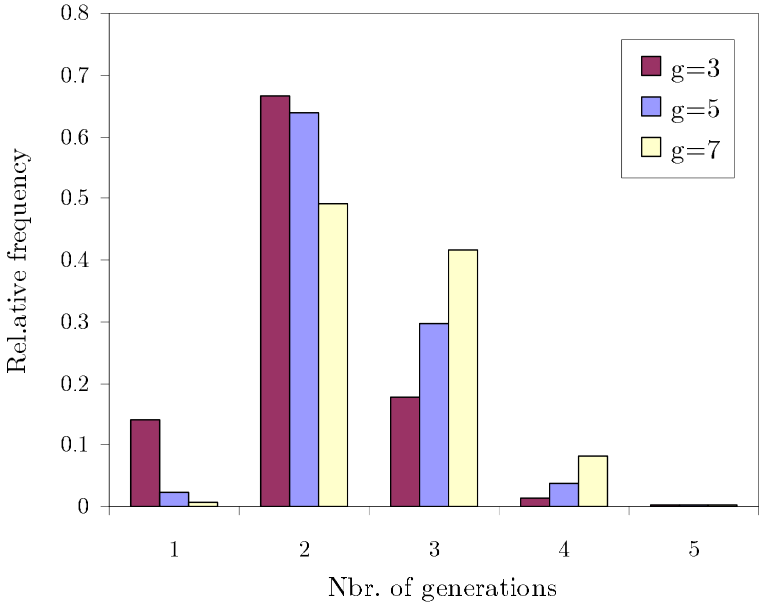


Figure 2.49: Histogram of the number of generations for different number of solutions per generation in ML-CAMS.

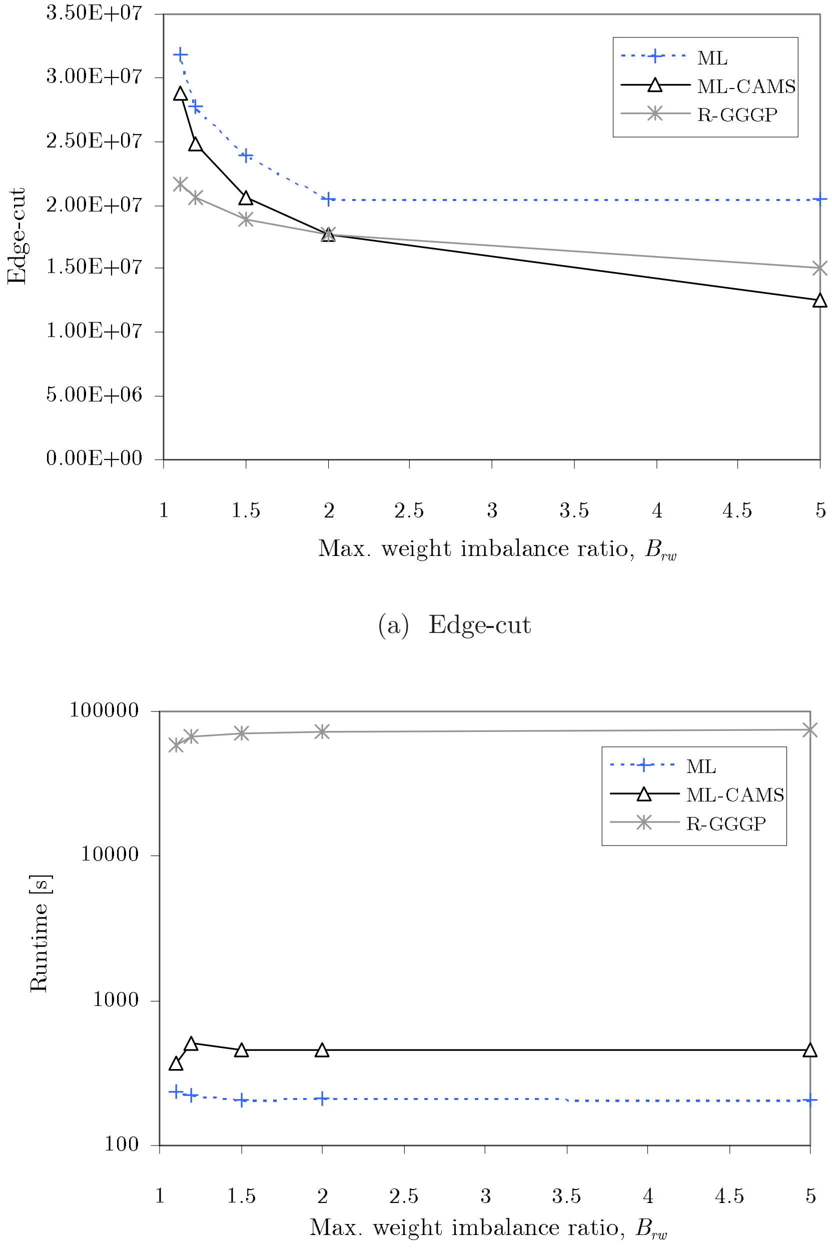
be freely modified and must therefore be set before the optimisation process is launched. Excessively tight constraints might restrict the optimisation process unnecessarily, which would possibly result in a degraded solution quality. Likewise, excessively loose constraints might cause that unacceptable solutions were configured in the network.

The main constraints are the weight constraints, the connectedness constraint and the site constraint. A priori, it can be envisaged that, although all the constraints help to define the set of feasible solutions, not all of them are equally restrictive. While weight constraints must necessarily be fulfilled, the connectedness and site constraints only aim to improve the visual appearance of solutions and, hence, need not always be ensured. Likewise, Table 2.4 shows that the average number of GPRS TSLs per PCU (i.e., 68.2) is far below the PCU capacity limit (i.e., 256). Hence, it is the imbalance among PCUs, and not the PCU capacity, what restricts the optimisation process. At this point, it is important to understand that the sensitivity to these constraints is method-dependent, i.e., not all methods are equally sensitive to the variation of constraints. Hence, the sensitivity analysis should cover both the best quality and the most efficient method. Hereafter, the analysis is restricted to ML, ML-CAMS and R-GGGP.

The first experiment evaluates the sensitivity of methods to the weight imbalance constraint. Figure 2.50 (a)-(b) present the variation of the main performance indicators when the maximum weight imbalance ratio, B_{rw}, is modified from 1.1 to 5. As expected, Figure 2.50 (a) shows that the total edge-cut decreases when the allowed imbalance is increased. From the figure, it is also deduced that relaxing the imbalance constraint beyond the value of 2 (which has been considered so far) hardly gives any edge-cut improvement for ML. By contrast, the same constraint relief is used by ML-CAMS to decrease the total edge-cut by 29%. Also remarkable is the fact that the solution achieved after 1000 R-GGGP attempts is worse than the ML-CAMS solution, although its runtime is two orders of magnitude larger. The origin of this effect can be found in the enlargement of the feasible solution space after relaxing the imbalance constraint. As a result, the naive MS approach (i.e., R-GGGP) needs more attempts to find the best solution when compared to ML-CAMS, which speeds up the convergence to the best solution. On the other hand, all methods encounter difficulties in finding an optimum solution for imbalance ratios lower than 1.5. For instance, the edge-cut of ML-CAMS increases by 62% when the imbalance ratio changes from 2 to 1.1. Nonetheless, it can be observed that ML-CAMS experiences gentle degradation with the reinforcement of the imbalance constraint. Under these conditions, the naive MS approach gives a better solution at the expense of a higher runtime. This result was expected, since the increased diversity in the search given by random trials becomes more valuable as the optimisation problem gets more complicated. Figure 2.50 (b) presents the influence of the imbalance constraint on runtime. It is observed that both MS methods experience an increase of runtime as the constraints are relaxed. In R-GGGP, this effect is due to a larger number of vertex moves during refinement, as the balanced initial partition built by GGGP must be modified significantly to exploit the allowed imbalance between subdomains. In ML-CAMS, the enlargement of the solution space also leads to an increase of the difference (i.e., distance) between the best solutions. Such an event delays the convergence to the best solutions, as fewer similarities are found among solutions. For instance, while 1000 R-GGGP attempts take only 1.2% more runtime when B_{rw} changes from 2 to 5, ML-CAMS needs 7.2% extra time. In contrast, ML is faster when the imbalance constraint is relaxed. Part of this reduction is due to a reduced number of coarsening steps. By allowing larger vertices in the coarsening process, the matching process is more flexible and the size reduction ratio of the last coarsening steps is increased.

The analysis now focuses on the connectedness constraint. All methods covered so far share the connected refinement algorithm, where the re-allocation of vertices is restricted to keep subdomains connected. Since this constraint is not strictly needed (even if it leads to more consistent solutions), it is interesting to evaluate the impact of eliminating this constraint. The comparison can now be extended to consider both heuristic and exact methods. Note that, in the previous experiments, exact methods were deliberately neglected, since they cannot deal with the connectedness constraint. Therefore, a fair comparison against heuristic methods is not possible, unless the connectedness constraint is relaxed for the latter methods.

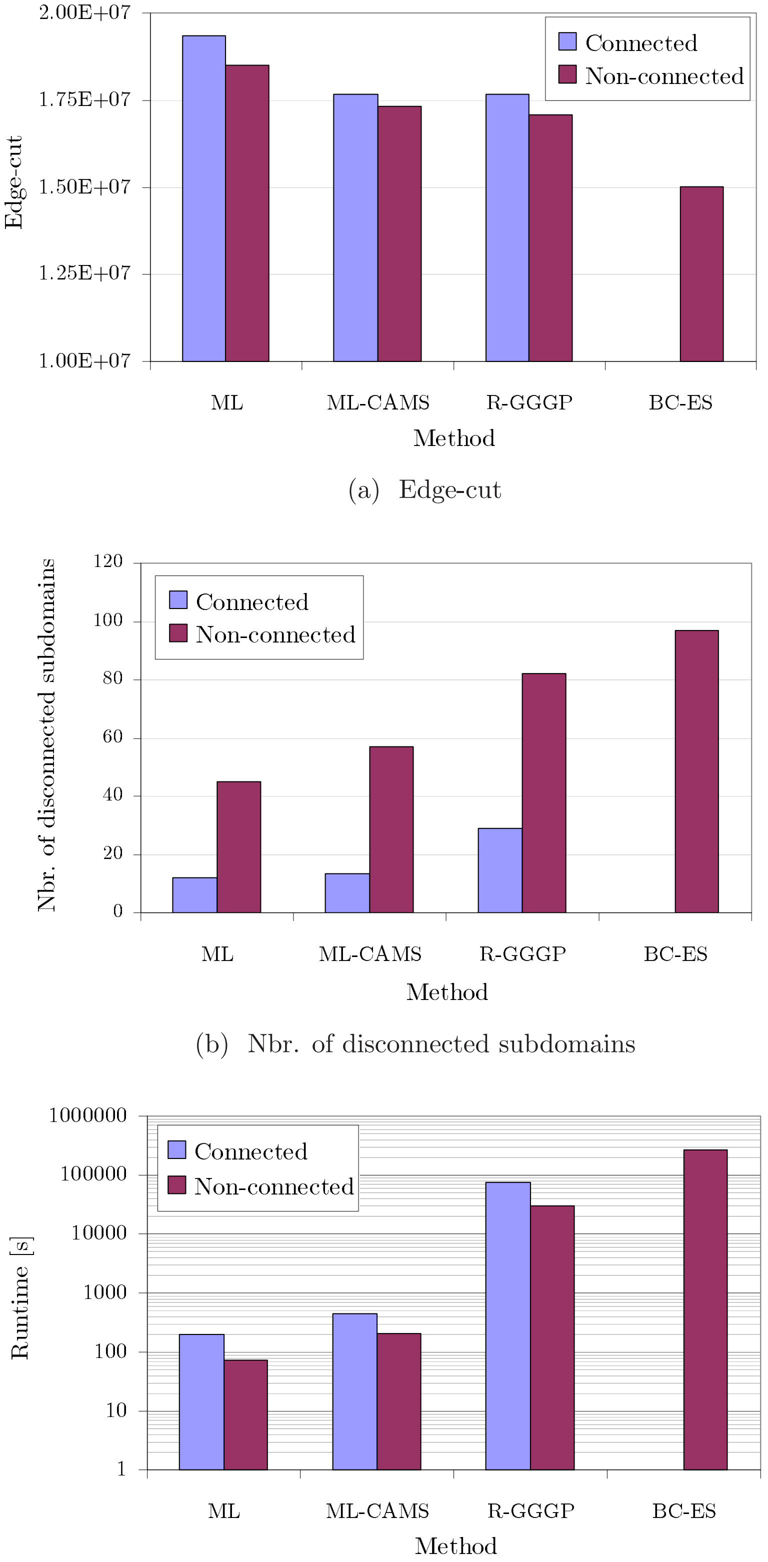
Figure 2.51 (a)-(c) present the performance of different methods with and without the connectedness constraint in the refinement algorithm. The results of BC-ES are also included in the figures for benchmarking purposes. Figure 2.51 (a) shows the influence of the constraint on the total edge-cut performance. As expected, the elimination of this constraint leads to a non-negligible edge-cut reduction in all methods. Specifically, ML-CAMS and R-GGGP achieve a 2.1% and 3.5% reduction in the total edge-cut when the connectedness constraint is eliminated. Nevertheless, the edge-cut of the best heuristic (i.e., R-GGGP) is still far from that of BC-ES. In particular, the ML-CAMS solution has 15% more edge-cut than the BC solution. This result gives evidence that there is still some space for improvement in the heuristic methods. Obviously, the lack of a connectedness constraint entails an increase of the number of disconnected subdomains. This increase is almost three-fold for any of the methods, as observed in Figure 2.51 (b). It is also remarkable that BC-ES gives the largest number of disconnected subdomains. Finally, Figure 2.51 (c) shows the change in runtime induced by the elimination of the connectedness constraint. A logarithmic scale is used on the runtime axis in order to show the performance of all methods together. From the figure, it can be deduced that the computational load can be significantly reduced by suppressing the connectedness requirement. This reduction is more than two-fold for any of the heuristic methods (note the logarithmic scale). From this result, it can be inferred that the computational load required by the connectedness checks in the connected refinement algorithm is more than half of the load of the entire method.



(a) Edge-cut

(b) Runtime

Figure 2.50: Sensitivity to the variation of the weight imbalance constraint.



(a) Edge-cut

(b) Nbr. of disconnected subdomains

(c) Runtime

Figure 2.51: Sensitivity to the elimination of the connectedness constraint.

Heuristic method	IO	ML-CAMS	ML-CAMS	R-GGGP	R-GGGP
Assignment granularity	Cell	Cell	Site	Cell	Site
Coarsening algorithm	-	SHEM	SM-SHEM	-	SM
Sum of edge weights [$\cdot 10^6$]	243.7				
Total edge-cut [$\cdot 10^6$]	82.3	17.7	22.5	17.7	21.8
Edge-cut ratio [%]	33.8	7.3	9.2	7.3	8.9
Average weight imbalance ratio	3.55	1.91	1.91	1.82	1.9
Total nbr. of disconnected subd.	192	13.4	16.0	29	30
Runtime [s]	-	455	355	74110	21321

Table 2.11: Performance of solutions with cell and site resolution.

The following experiments quantify the edge-cut impairment caused by forcing that co-sited cells are in the same PCU. For this purpose, graphs are first coarsened by SM and then partitioned. Table 2.11 compares the performance of solutions built by several methods with cell and site resolution. From the table, it can be concluded that solutions with site resolution perform competitively against solutions with cell resolution presented so far. For ML-CAMS, restricting co-sited cells to be in the same PCU only causes an absolute increase of 1.9% in the edge-cut ratio in exchange for a 28% reduction of runtime. While the former effect is caused by the lack of freedom in the assignment process, the latter is derived from the suppression of the refinement process in the last uncoarsening step. For R-GGGP, the edge-cut impairment is slightly less (i.e., 1.6%), which suggests that the optimisation surface becomes more irregular when the site granularity is enforced. This would explain that the method based on the detection of similarities (i.e., ML-CAMS) performs worse than the method that samples the entire solution space (i.e., R-GGGP). At the same time, the runtime reduction in R-GGGP is more pronounced (i.e., three-fold reduction), since the suppressed refinement process over the original graph takes most of the computational load of R-GGGP.

Although solutions with site resolution perform competitively, experiments show that it is not wise to use this strategy to reduce the computational load. On the contrary, the use of SHEM together with the suppression of the last refinement step is a better alternative to reduce runtime. Table 2.12 backs up this statement by comparing the performance of ML-CAMS with SM-SHEM and SHEM coarsening, with no refinement in the last uncoarsening step. While the former method correspond to the standard ML-CAMS algorithm with site resolution, the latter will be referred to as ML-CAMS (1^{st} *level*). From the table, it is clear that the SHEM variant is better both in terms of solution quality and runtime.

The assessment process have hitherto only considered objective metrics. Thus, nothing has been stated about the ease of management of solutions, which is a subjective metric. A CPAP solution is easier to check if it is geographically consistent, i.e., if the service areas of PCUs are continuous and do not overlap with each other. Obviously, the connectedness and site constraints play a key role on the visual appearance of solutions. Thus, enforcing these conditions leads to PCU plans that look simpler on a map. Although this effect could be inferred from the smaller number of disconnected subdomains, the following analysis shows the effect of these constraints graphically. For simplicity, the analysis is restricted to ML-CAMS over a test case, although similar results are observed for other methods and problem instances. To allow comparison, the test case is the trial BSC, already presented in Section 2.4.1. Figure 2.52 shows the geographical layout of different solutions. Figure 2.52 (a) illustrates the

Heuristic method	ML-CAMS	ML-CAMS(1^{st}-level)
Assignment granularity	Site	1^{st}-level
Matching technique	SM-SHEM	SHEM
Sum of edge weights [·1e6]	243.7	
Total edge-cut [·1e6]	22.5	18.6
Edge-cut ratio [%]	9.2	7.6
Average weight imbalance ratio	1.91	1.90
Total nbr. of disconnected subd.	16.0	18.0
Runtime [s]	355	296

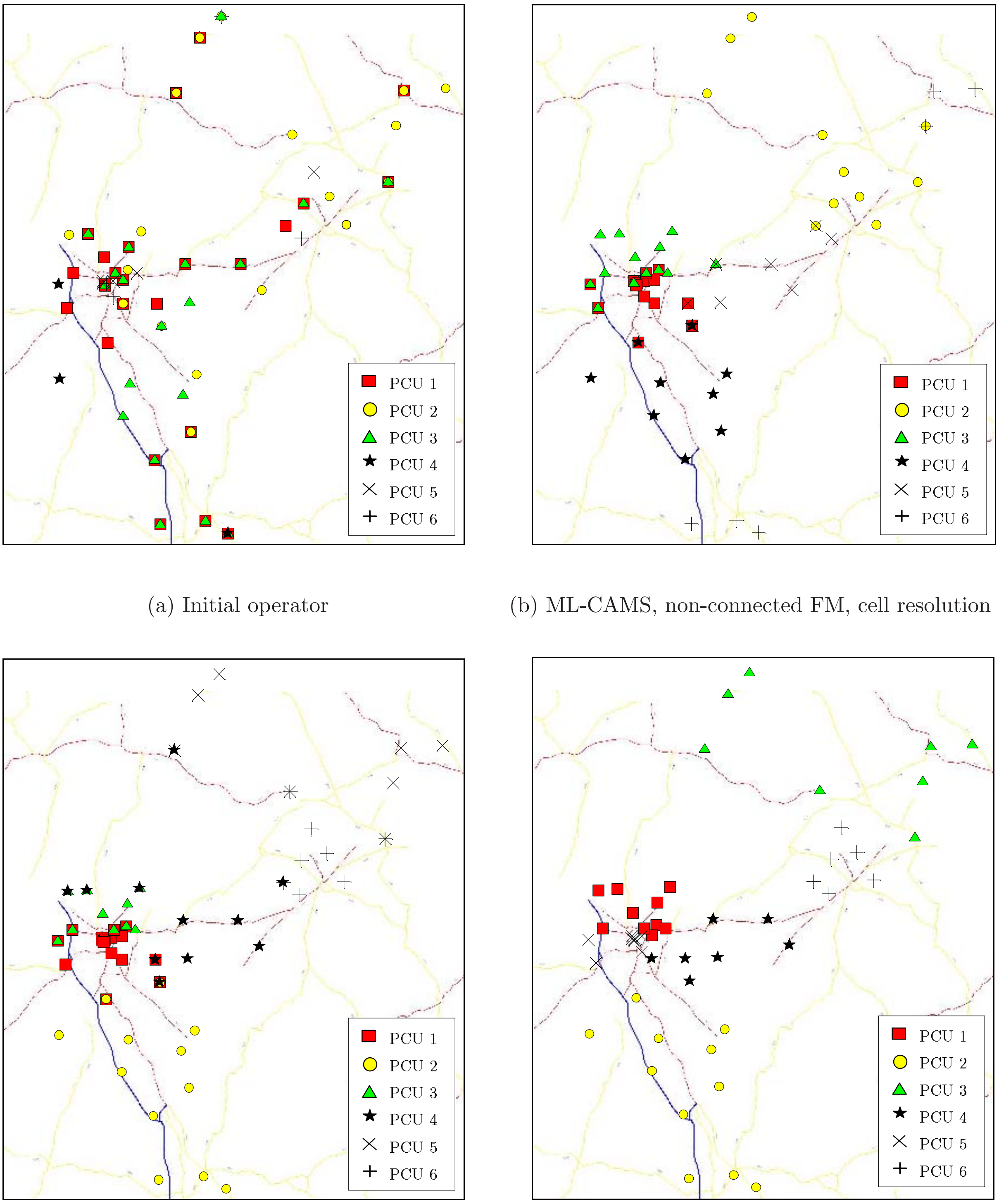
Table 2.12: Performance of site-level and first-level solutions.

initial operator solution (i.e., IO), while Figure (b)-(d) show ML-CAMS solutions with different constraints. Figure 2.52 (b) shows the ML-CAMS solution with non-connected refinement and cell resolution. It is observed that, although the solution looks much simpler than the operator's solution, one disconnected subdomain still exists. More specifically, PCU 6, denoted by a '+' symbol, covers cells on the bottom and upper-right of the map. In contrast, Figure 2.52 (c) shows that connected refinement avoids disconnected subdomains, which makes that PCUs cover a continuous area. However, there exists overlapping between areas of different PCUs due to the assignment of co-sited BTSs to different PCUs. Obviously, the solution would be easier to check if the service areas of PCUs did not overlap. This can be solved by enforcing the site constraint, as depicted in Figure 2.52 (d).

2.6 Conclusions

This chapter has dealt with the problem of assigning cells to PCUs in GERAN. The influence of this assignment process on the performance of the data transmission has been discussed. The problem has been formulated as a GPP. In this formulation, the network area under optimisation is modelled by a graph, whose vertices and edges are the cells and adjacencies of the network, respectively. In this graph, the problem of clustering vertices to minimise the relationship between vertices in different clusters models the assignment of cells to the existing PCUs. Three different mathematical models have been presented, based on the traditional ILP formulation of the GPP. Based on the previous formulation, two methods have been proposed to achieve exact or heuristic solutions. The first approach uses the traditional BC method to solve an enhanced version of the conventional ILP model of the GPP. This method achieves the optimal solution at the expense of an increased computational load. However, it does not guarantee that subdomains in the final solution are connected. The second approach combines the ML connected refinement algorithm with adaptive MS techniques. The resulting method finds high-quality solutions quickly, which in most cases also fulfill the connectedness constraint.

To prove the relevance of the problem, a field trial has been conducted over a limited geographical area. Based on drive tests, the trial has proved that the service break for inter-PCU CRSs is much larger than for intra-PCU CRSs. Concretely, the average service break for the inter-PCU case is more than twice the value for the intra-PCU case. This observation gives clear evidence that the number of inter-PCU CRSs must be minimised to improve the performance of data transmission. Likewise, the trial has shown that the existing network configuration built



(a) Initial operator

(b) ML-CAMS, non-connected FM, cell resolution

(c) ML-CAMS, connected FM, cell resolution

(d) ML-CAMS, connected FM, site resolution

Figure 2.52: Map view of solutions built by several methods in a live BSC.

manually by the operator is far from optimal. Thus, trial results have shown that even a simple optimisation algorithm can produce a significant performance enhancement.

As the trial only covered a limited area, a comprehensive analysis has been performed over an extensive collection of graphs constructed from data of a live GERAN. The analysis scenario corresponds to the geographical area of 61 BSCs. The set of 61 problem instances is deemed to be large enough to provide representative results. In the absence of PS mobility statistics, HO statistics related to CS services have been used to construct the graphs. Assuming that user mobility in both modes is similar, the errors in such an analysis should be relatively small. During the analysis, the proposed methods have been compared with other classical methods. Analysis results confirm that the solution currently configured in the network can be significantly improved by any of the proposed method. This improvement mainly affects the inter-PCU CRS ratio and the load imbalance between PCUs.

The analysis of exact methods has shown the benefits of the improved ILP model of the problem. The BC method obtains solutions with less intra-PCU CRS ratio than the best heuristic method, provided that the connectedness constraint is eliminated. Although the computational load of these methods prevents them from being applied on a daily basis, they can still be used during the planning stage for benchmarking purposes. With the availability of efficient optimisation codes, these methods might well become the preferred option in the future.

Among the heuristic methods, the method that combines ML refinement with CAMS for the initial partitioning provides the best trade-off between solution quality and runtime. Concretely, this method reduces the inter-PCU CRS ratio of the existing solution by 80%, which is also the performance of the best heuristic method. In addition, this method nearly halves the weight imbalance ratio among PCUs in the current solution, while it reduces the number of disconnected subdomains by one order of magnitude. From the results, it can be concluded that the proposed method outperforms classical ML approaches in terms of solution quality at the expense of a slight increase of computing time, when compared to the standard ML algorithm. It should be pointed out that, as the execution time for the whole network is in the order of minutes, the bottleneck is deemed not to be exclusively in the execution of the algorithm itself, but also in the access to the databases to retrieve the input data. Despite the fact that ML-CAMS is random in nature, it provides extremely robust results. Likewise, the overall inter-PCU CRS ratio gracefully degrades with the reinforcement of any of the constraints of the problem. In particular, an absolute increase of only 1.9% is observed in the previous indicator when all cells in the same site are forced to be assigned to the same PCU. Likewise, the same ratio suffers an absolute increase of 4.6% when the maximum load imbalance ratio among PCUs changes from 2 to 1.1. In addition, the need for connected subdomains only entails an absolute increase of 0.2% in the referred ratio. Finally, the combination of ML-CAMS with connected refinement and site resolution provides solutions that are much easier to check on a map, which are always preferred by network operators.

In the light of these results, it can be concluded that the proposed ML-CAMS method is a worthy candidate for the re-planning of the cell-to-PCU assignment, which must be performed by GERAN operators as part of their daily routine.

Chapter 3

Optimisation of Handover Parameters for Congestion Relief in GERAN

This chapter deals with the problem of tuning HO parameters to solve localised congestion problems in GERAN. After a brief description, the problem is formulated as an optimisation problem. A heuristic method is then proposed to share traffic between adjacent cells. Field trial results are subsequently presented to show how a simple algorithm can improve the performance of a live network. Finally, a comprehensive performance analysis of more sophisticated methods is carried out based on simulations.

3.1 Introduction

In recent years, the success of mobile services has led to an exponential increase in the number of users in cellular networks. As a consequence, traffic engineering has become one of the main areas of interest for operators of these networks. For CS services, the availability of well-known traffic models allows for a proper dimensioning of network resources during the design stage. However, as network evolves during the operational stage, the matching between traffic demand and network resources becomes more difficult. Hence, continuous adaptation of network resources is needed to cope with the dynamic nature of traffic demand. This adaptation requires a re-planning process that ends up with the re-allocation or addition of network resources. To reduce operational and capital expenditures, the frequency with which this action is carried out is kept to a minimum. In the meantime, traffic management remains the only solution to maintain network quality. Formally, *traffic management* is defined as the set of mechanisms and policies that allow the network to provide adequate QoS to the end-user with the existing infrastructure. While management mechanisms are integrated in RRM algorithms, management policies are normally applied through the optimisation of parameters in these algorithms. This is the reason why traffic management is one of the main areas where parameter optimisation has been successfully applied in mobile networks.

The management of CS-traffic in GERAN mainly deals with the selection of the BTS to which every MS is attached. This goal is achieved through three RRM procedures: admission control, congestion control and load balancing. *Admission control* decides whether or not to accept new connections depending on the availability of TSLs and statistical information on connection quality and interference levels. *Congestion control* is in charge of detecting and recovering from overload situations in the network. Finally, *load balancing* aims at redistributing the traffic demand across cells in the network in order to prevent congestion problems.

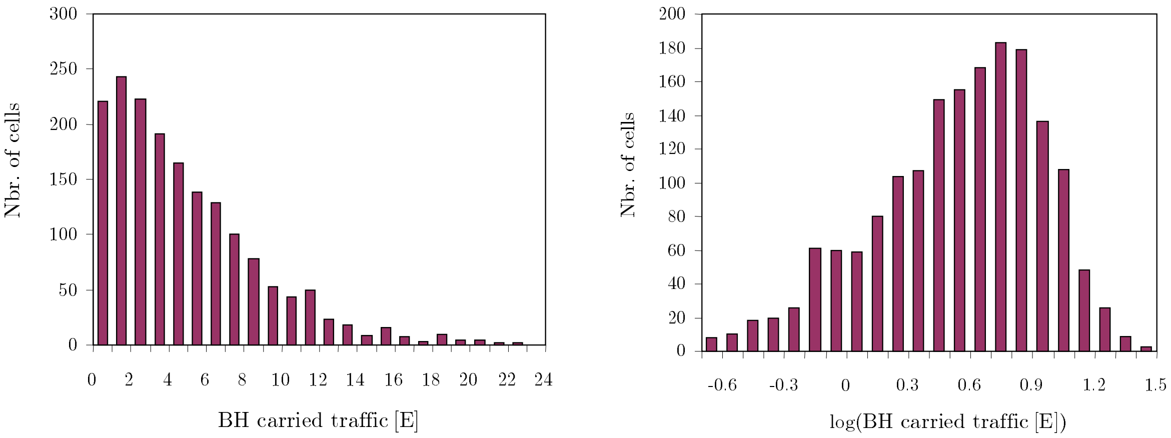
The traffic load in a congested cell can be reduced by sharing traffic between adjacent cells. This can be effectively performed by sending users in the cell border to adjacent cells that overlap with the congested cell. This technique is referred to as *load sharing through adaptive cell resizing*. In GERAN, the modification of HO boundaries is the best means to adjust the cell service area. To attain such resizing effect, tuning of HO margins is widely used. Based on this technique, several methods have been proposed in the literature [6][7][8][9]. All these methods aim to minimise the number of blocked calls in the network by adapting HO margins between adjacent cells, mainly differing in the information that drives the tuning process. Although these methods have been known for quite some time, several issues have limited their use in practice. The main drawback of displacing the HO boundary is the deterioration of call quality due to the fact that MSs are not served by the best BTS in terms of radio signal-level. As no comprehensive analysis of the limits of this technique has been carried out so far, operators usually avoid changing HO parameters to keep the network on the safe side.

In this work, the optimisation of HO parameters in GERAN is analysed from the perspective of re-planning procedures. As the main goal is to solve permanent (i.e., non-transient) local congestion problems, the adjustment of parameters is performed based on statistical (i.e., non-instantaneous) traffic indicators. By optimising the network over a longer time horizon, the stability of the network is improved. As a result, the overall call blocking ratio is minimised and more traffic is carried by the network. The preliminary analysis shows the limitations of tuning HO margins for traffic sharing purposes. To deal with such limitations, a method is then proposed to equalise network traffic by tuning several HO parameters. The method consists of two algorithms that jointly adjust HO margins and signal-level constraints based on network statistics. The aim of the method is to minimise the overall call blocking in the network, while keeping the impairment of the overall connection quality within reasonable limits. The assessment of the method is based on a two-folded approach. Firstly, field trial results show the enhancement that a simple tuning algorithm produces on the performance of a limited network area. Subsequently, more sophisticated methods are evaluated in a realistic simulation environment. During the analysis, the proposed method is compared to other classical approaches to prove that it is a cost effective means to increase network capacity.

The rest of the chapter is organised as follows. Section 3.2 outlines the role of HO parameters in the definition of cell service areas in GERAN. Section 3.3 describes three tuning algorithms to solve congestion and quality problems in GERAN. Section 3.4 presents field trial results of an algorithm for tuning HO margins in a live network. Section 3.5 presents a comprehensive analysis of refined methods over a simulation model. Section 3.6 presents the main conclusions of this chapter.

3.2 Problem Formulation

This section begins with a discussion of the reason for congestion in mobile networks. The subsequent paragraphs present the basic parameters of the HO algorithm in GERAN, which can be used to control the service area of cells in the network. Finally, the current state of research and technology is discussed. The issues presented here will justify the need for the method proposed in the next section.



(a) Linear (b) Logarithmic

Figure 3.1: Traffic distribution in a cell level in a live network.

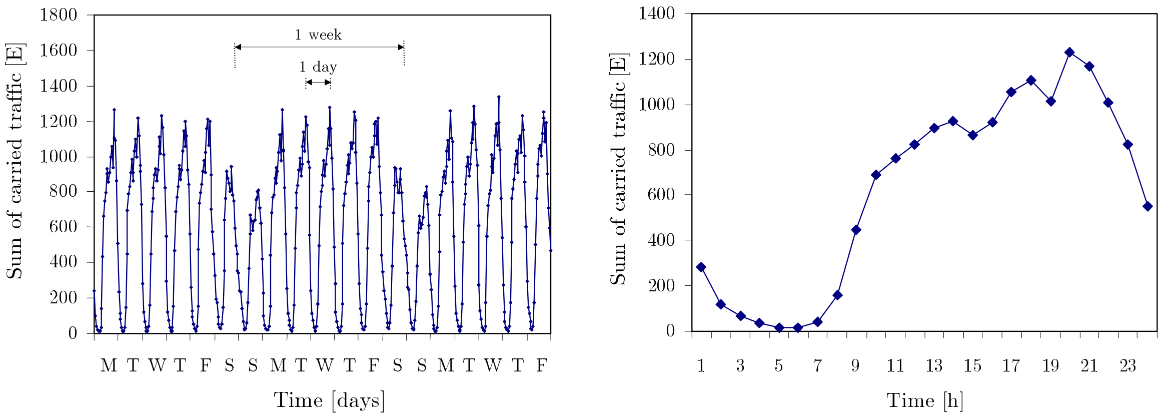
3.2.1 The Network Congestion Problem

The demand of mobile services has experienced an explosive growth in the last decades. In the past, operators coped with this growth by increasing network capacity to meet the predicted increase of traffic demand. Thus, operators used a weekly average of the daily peak traffic intensity over one hour to estimate resources for network cells [87]. In this process, it was not uncommon to anticipate the predicted traffic growth in three to six months time [88]. As a result, the network was generally over-dimensioned. However, this approach is not an effective strategy any more, as capital expenditures must be reduced due to increased competition among operators. As a consequence, traffic congestion is a common occurrence in current networks, which leads to call blocking for users and lost revenues for operators.

A major contributor to congestion problems in a cellular network is the uneven distribution of traffic in both the spatial and temporal domain. Imperfect matching between traffic demand and network resources in the spatial domain causes that some cells suffer from congestion, while surrounding cells are underutilised. At the same time, temporal fluctuations of traffic demand cause short periods of congestion followed by long periods of underutilisation. The following paragraphs present several models for cellular traffic.

From an overall network perspective, the *spatial traffic distribution* can be modelled by a log-normal distribution [89]. To prove this, Figure 3.1 (a)-(b) present the histogram of CS traffic carried during the Busy Hour (BH) in the cells of a live network. Both figures differ in the linear or logarithmic scale in the x-axis. The histogram in Figure 3.1 (b) resembles that of a normal distribution, suggesting that traffic demand in a cell level is a log-normal random variable. The previous statistical model can be extended to take into account the spatial correlation of traffic demand. Field tests have shown that traffic in urban environments is clustered around hotspots related to business activities. Thus, traffic density can be described geometrically with an exponential function centred in hotspots with a decay factor of 1-2 km [90].

The *temporal traffic distribution* is a combination of long-term and short-term trends. Long-term changes comprise yearly overall growth and seasonal fluctuations, while short-term changes include weekly, daily and hourly fluctuations. Within a week, the highest traffic demand is concentrated on working days in the middle of the week. In contrast, the daily traffic follows a



(a) 3 weeks (b) 1 day

Figure 3.2: Traffic distribution on an hourly basis in a live BSC.

multi-gaussian pattern associated to the morning, afternoon and night periods [90]. On top of these daily fluctuations are rapid fluctuations due to the randomness of the call arrival process. For clarifying purposes, Figure 3.2 presents an example of temporal traffic distribution in a BSC. Figure 3.2(a) shows the hourly profile of the carried traffic in a BSC over a 19-day period. The displayed traffic pattern shows daily and weekly periodicity on top a slow growth trend. Figure 3.2(b) expands the time axis to show the traffic distribution during the course of a day, where the multi-gaussian pattern is evident.

Although traffic periodicity is maintained across the network, it is worth noting that the maximum traffic is not simultaneously reached in all network cells [90]. This behaviour is obvious in a short time-scale (e.g., seconds) due to the randomness of the call arrival process. However, such a behaviour is also observed in a much larger time scale (e.g., hours). For instance, Figure 3.3 presents the hourly profile of the carried traffic in three different BSCs. To ease the comparison, traffic values have been normalised by the BH traffic on each BSC. From the figure, it is clear that the hourly traffic demand is not fully time correlated among BSCs, and, consequently, it is likely not to be so among cells. Thus, the BH is not the same for the three BSCs. This difference is caused by fluctuations of the spatial distribution due to massive user movement along a day. As users move from work to home at the end of the day, the bulk of traffic demand shifts from business to residential areas.

The discussion so far has been around the issue of traffic modelling, but nothing has been said about congestion modelling. As congestion is caused by an excess of traffic demand over network resources, it is expected that most of the previous models are still valid for congestion. However, it is worth noting that, unlike traffic demand, congestion is strongly influenced by the existing distribution of network resources, and it is thus affected by operator's re-planning actions.

To find a distributional model for congestion, a congestion indicator must first be defined. The *congestion rate* (CR) in the BH is usually adopted to evaluate congestion performance in a cell. This indicator represents the percentage of time in which all TSLs in the cell are busy during the BH. If the call arrival process is Poissonian, the CR coincides to the call blocking probability based on the PASTA (*Probability Averages See Time Averages*) property [91]. Under this assumption, the CR in a cell is strongly correlated to the ratio of calls blocked due to lack of resources, which is referred to as *blocking rate* (BR). The previous statement holds true,

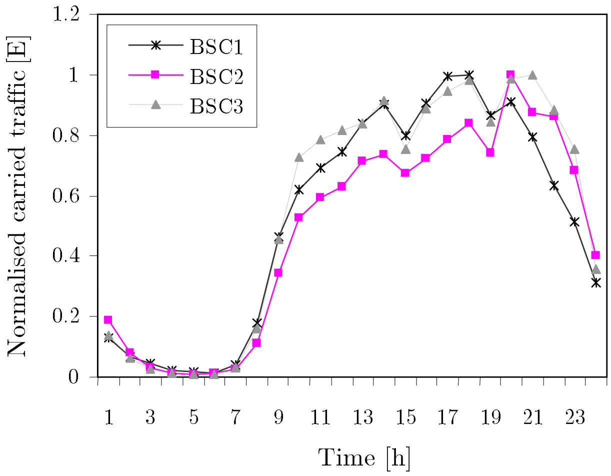


Figure 3.3: Traffic distribution on an hourly basis in three different BSCs.

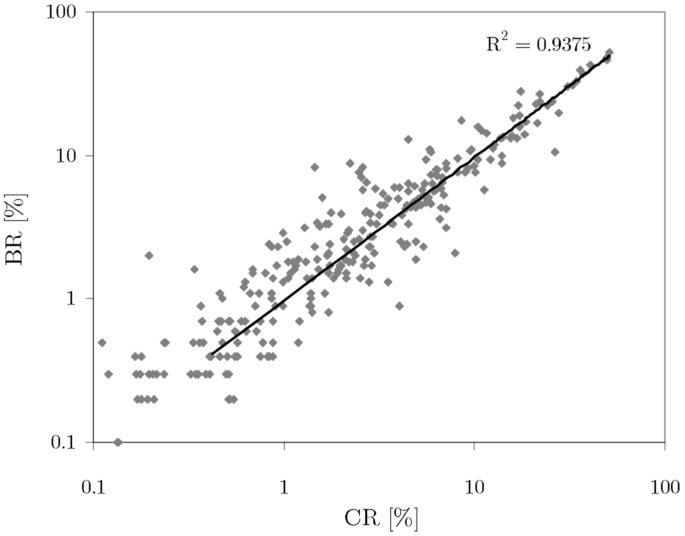


Figure 3.4: A scatter plot of congestion rate and blocking rate in cells of a live network.

regardless of the existence of PS traffic, since CS traffic has priority over PS traffic. For instance, Figure 3.4 shows a scatter plot of CR and BR in the cells of a live network. Both indicators correspond to the BH of each cell in a day. A trend line has been included together with the squared sample correlation coefficient, R^2. The high value of R^2 proves that both indicators are highly correlated. It is also observed that the correlation is weaker for lower values of both indicators, which correspond to lower values of traffic demand. This is a consequence of the granularity of congestion time measurements and the reduced statistical confidence of hourly BR measurements when the number of call attempts falls below a certain value.

Ideally, CR in a cell should never surpass a threshold that, in most cases, coincides to the desired GoS (2-5%, typically). During the re-planning stage, operators add new transceivers to those cells that steadily display unacceptable CR values. This reactive strategy ensures that, in a well-planned network, the number of cells with unacceptable congestion performance is small in relative terms. For instance, Figure 3.5 shows the ECDF of CR in the cells of a live network. From the figure, it can be deduced that, despite the fact that the overall network is

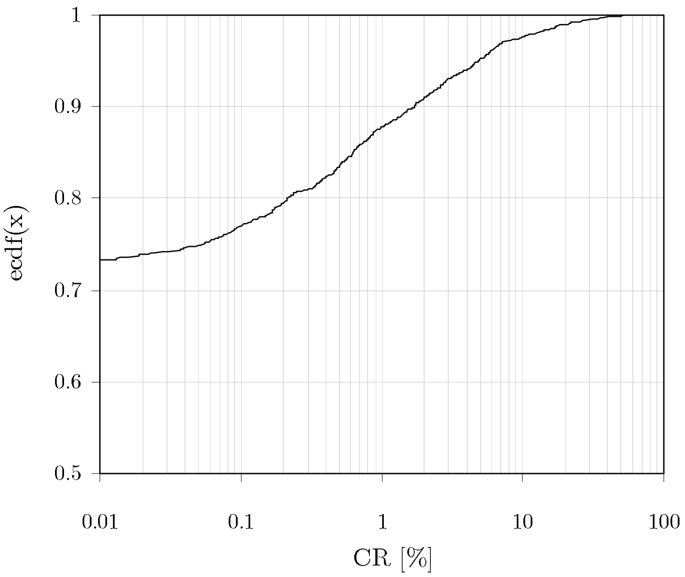


Figure 3.5: Empirical cumulative density function of CR values in a live network.

over-dimensioned, since 73% of the cells experience no congestion at all, a non-negligible share of cells (i.e., $100 \cdot (1 - 0.95)$=5%) display CR values above 5%. More critical, the latter cells proved to carry more than 10% of the network traffic. These figures give clear evidence of the imperfect matching between traffic demand and network resources.

As traffic resources are extended in the network, the spatial correlation of congestion becomes weaker and congested cells tend to appear isolated. Thus, the probability that two adjacent cells experience unacceptable CR values is small. Likewise, the probability that two adjacent cells are simultaneously congested is also small. These principles are the basis of traffic management policies that share traffic between neighbour cells, which will be described in the next section.

3.2.2 The Modification of Handover Boundaries for Congestion Relief in GERAN

In a cellular communications network, the HO mechanism is in charge of maintaining the traffic connection to an MS. Thus, a HO takes place whenever the MS moves from the coverage area of one cell to another while an active call is on-going. This process entails the establishment of a connection to the new (target) cell and the release of the connection to the old (serving) cell.[1]

There are basically two types of HOs: imperative and better-cell. An *imperative* HO is triggered whenever the BSC detects that some threshold values pertaining to the performance of a certain radio link are exceeded. The term 'imperative' refers to the fact that a HO must be performed urgently to maintain acceptable connection quality. In contrast, a *better-cell* HO is performed when a neighbour cell is better suited for the connection than the serving cell, which is evaluated at regular time intervals. While the former type of HO aims to ensure the quality in individual connections, the latter is conceived for optimising the overall network performance.

[1] This work only considers the *inter-cell handover* case, where a new connection is established due to a user moving between different cells. Therefore, the *intra-cell handover* case, where a connection is re-allocated to a different time slot in the serving cell due to interference, is neglected.

In a well-designed network, better-cell HOs tend to dominate as MSs are handed over long before imperative HO thresholds are reached. For this reason, operators often use parameters in better-cell HOs to implement their traffic management policy.

The HO procedure mainly consists of three stages: measurement processing, HO triggering and target-cell evaluation. In the first stage, the BSC averages measurement samples of radio signal power and quality for each connection. The subsequent HO triggering process depends on the type of HO. While imperative HOs are triggered by crossing a threshold, better-cell HOs are triggered based on a timer. Once a HO has been triggered, the target cell evaluation process selects the most suitable cell in terms of network performance. Both triggering and target-cell cell evaluation are based on pre-processed measurements and several specific parameters.

The discussion now focuses on the most important better-cell HO in GERAN: the *power budget* (PBGT) HO [92]. The PBGT HO belongs to the class of relative signal-strength HOs [93]. A PBGT HO is performed whenever the averaged signal-level from a neighbour BTS exceeds the one received from the serving BTS by a certain margin, provided a certain minimum signal-level is ensured. Such a mechanism ensures that, under normal conditions, any MS is served by the BTS that provides minimum pathloss. Thus, the transmitted power in the network is minimised.

As stated previously, PBGT HOs are triggered periodically. Once the PBGT HO timer of a connection has expired, the target-cell evaluation consists of two steps. First, the algorithm determines which of the neighbour cells will be evaluated. This is performed by comparing the averaged signal level received from all neighbour cells against some threshold. The candidate cells are then ranked by priority, based on their occupancy and hierarchical level in the network. The best candidate cell is finally selected among those with the highest priority. This process can be expressed in the form of equations. In the following discussion, it is assumed that all neighbour cells have the same priority (i.e., no prioritisation is performed based on load or hierarchical level). Variables in capital letters are random variables associated to radio link measurements, while variables in lower-case are parameters in the HO algorithm, which can be optimised. Likewise, signal-levels units are dBm, while power budget and margin units are dB.

The list of candidate cells is built by evaluating (3.1) for all neighbour cells. $\overline{RXLEV\_NCELL_j}$ is the averaged received signal level from neighbour cell j. $RxLevMinCell_{i \to j}$ is a threshold parameter, defined on a per-adjacency basis, that specifies the minimum signal level that must be received from cell j to be a valid HO target when connected to cell i. The last term, $\mathrm{Max}(0, msTxPwrMax_j - P)$, ensures that an MS has enough transmit power to enter the new cell. This is performed by comparing the maximum permitted UL transmission power in cell j, $msTxPwrMax_j$, with the maximum transmission power capability of the mobile handset, P. In an urban environment, where $msTxPwrMax_j$ tends to be small, this term is often equal to zero.

$$\overline{RXLEV\_NCELL_j} \geq RxLevMinCell_{i \to j} + \mathrm{Max}(0, msTxPwrMax_j - P) \tag{3.1}$$

The most suitable cell is chosen by evaluating the PBGT of the candidate cells as reflected in (3.2). To calculate the PBGT of a cell j, three factors are taken into account: (a) the average received level from the candidate and serving cell, $\overline{RXLEV\_NCELL_j}$ and $\overline{RXLEV\_DL}$, respectively; (b) the maximum transmitted power in the serving and candidate cell, $msTxPwrMax_i$ and $msTxPwrMax_j$, respectively; and (c) the current status of the DL power control in the serving cell, given by the actual and maximum BTS transmitted power, $BTSTXPWR$ and $btsTxPwrMax_i$, respectively. The PBGT of each candidate cell is compared against a threshold value, $HoMarginPBGT_{i\rightarrow j}$, and the cell that exceeds this threshold by a larger amount is finally chosen. If no cell satisfies (3.2), the MS remains in the current serving cell.

$$\begin{aligned} PBGT_{i\rightarrow j} = & \overline{RXLEV\_NCELL}_j - \overline{RXLEV\_DL} \\ & + \mathrm{Min}(msTxPwrMax_i, P) - \mathrm{Min}(msTxPwrMax_j, P) \\ & + BTSTXPWR - btsTxPwrMax_i \qquad \geq HoMarginPBGT_{i\rightarrow j} \end{aligned} \tag{3.2}$$

The role of the $HoMarginPBGT_{i\rightarrow j}$ parameter is more easily understood when $msTxPwrMax_i$=$msTxPwrMax_j$ and the DL power control is disabled (i.e., $BTSTXPWR$ = $btsTxPwrMax_i$). In these conditions, $HoMarginPBGT_{i\rightarrow j}$ is the minimum difference between the received signal levels from the source and target cell to perform a PBGT HO (and hence the name margin). Likewise, it is observed that $HoMarginPBGT$ is defined on a per-adjacency basis, i.e., for any pair of cells i and j, $HoMarginPBGT_{i\rightarrow j}$ can be different from $HoMarginPBGT_{j\rightarrow i}$.

Parameters in the previous algorithm ensure the quality and stability of the HO mechanism. On the one hand, the signal-level constraints, $RxLevMinCell_{i\rightarrow j}$, are used to discard neighbour cells that do not provide adequate radio signal level. On the other hand, the margins, $HoMarginPBGT_{i\rightarrow j}$, avoid repetitive HOs for slow-moving users due to the presence of obstacles in the line of sight. The latter fading mechanism, referred to as *slow fading*, can be modelled by a log-normal random variable to be added to the path loss with mean 0 and standard deviation, σ_{sf}, dependent on the propagation environment (i.e., 6 dB in rural areas, 8 dB in urban areas) [94]. Thus, margins are normally set to values higher than σ_{sf} to avoid unnecessary HOs [95].

Setting a positive margin causes a delay of the HO event. This effect is observed in Figure 3.6, where the HO of an MS moving from BTS i to BTS j is analysed. The discontinuous lines represent the signal-level distributions from both BTSs over distance, while the solid line represents the signal level experienced by the connection over distance. In the figure, it is observed that the HO event is delayed until the signal level from the target cell j is $HoMarginPBGT_{i\rightarrow j}$ dBs higher than that of the serving cell i. This delay can be significant in a macro cellular environment, where the slope of the signal-level distribution with distance tends to be smaller [96].

It is worth noting that, although the analysis has been restricted so far to a pair of cells, the same principle holds for all neighbour cells of the original source cell. Likewise, although the analysis has only covered the HO in one direction of the adjacency (i.e., from cell i to cell j), the same principle is equally valid in the opposite direction (i.e., from cell j to cell i). The combination of positive margin values in both directions provides a hysteresis region close to the cell boundary, since the HO in both directions is delayed from the point where the signal levels of the serving and target cells cross. It is worth noting that this hysteresis region prevents repetitive HOs due to slow fading as long as the sum of margins is larger than $2 \cdot \sigma_{sf}$ [95]. Thus, positive margin values are not strictly needed in both directions of the adjacency as long as the sum of margins satisfies the previous condition, i.e., the margin in one direction of the adjacency can be negative if it is compensated for by proper adjustment of the margin in the opposite direction of the adjacency. This is the basis of the method that modifies HO boundaries discussed in the next section.

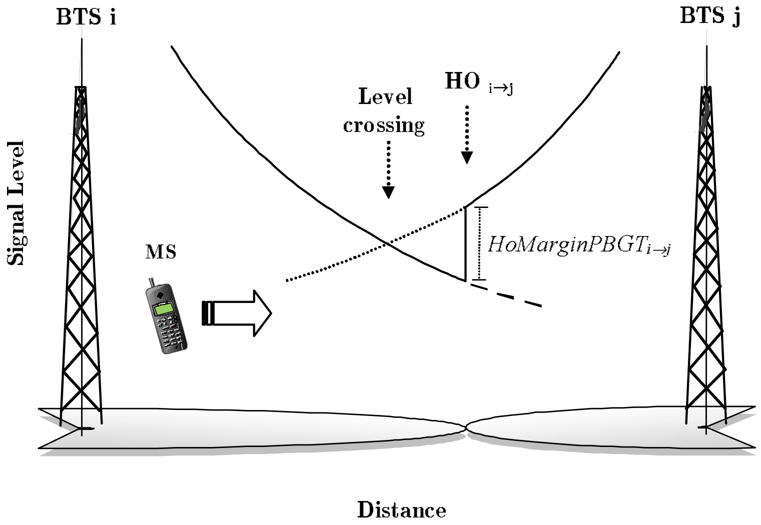


Figure 3.6: The handover margin.

From the previous analysis, it is clear that the tuning of HO margins is an effective means to change the service area of cells in a cellular network. This effect can be used to equalise the network load by displacing users between neighbour cells. However, such a strategy has several drawbacks. Figure 3.7 shows the impact of changing HO margins on the service area of a pair of adjacent cells. The figure represents signal-level distributions from both BTSs to show how the user connection signal-level is influenced by the displacement of the HO event. In the initial configuration, the $HoMarginPBGT_{i \to j}$ is set to X dB. This setting causes that the HO event is performed only when the signal level from target cell j is X dB higher than that of serving cell i. Subsequently, the $HoMarginPBGT_{i \to j}$ is raised from X dB to (X+ΔX) dB. As a consequence, the HO event is delayed further and the service area of cell i is enlarged in the direction of cell j. Unfortunately, the gain from the tuning process comes at the expense of reduced call quality and increased co-channel interference level throughout the system, as calls are carried in cells other than their nominal (i.e., optimal in terms of radio conditions) cells. This effect is clearly observed in the decrease of the minimum connection signal level experienced by the user, as the mobile drifts further into the new cell.

From Figure 3.7, it is deduced that reducing the HO margin of an outgoing adjacency leads to the opposite effect. As the HO event is brought forward, the size of the serving cell is reduced and that of the target cell is increased. Initially, this modification of the HO boundary does not affect negatively to connection quality, but, on the contrary, it leads to an increase of the minimum connection signal level. However, once this margin takes negative values, (3.2) clearly indicates that a HO is possible to a worse cell in radio signal terms, which might lead to connection-quality problems. Therefore, negative HO margins should be avoided whenever possible.

Although Figure 3.7 only presents the case of a moving user, changing HO boundaries might also affect stationary users. For the latter users, the tuning process would have an influence on the entire connection, unlike moving users, for which the effect is reduced to the vicinity of HO points. From the figures, it can also be deduced that the effectiveness of the referred strategy relies heavily on the overlapping between adjacent cells. For these reasons, it is expected that the impact of these methods is higher for macro cellular urban environments, where cell overlapping is large, user mobility is low and traffic demand tends to be localised.

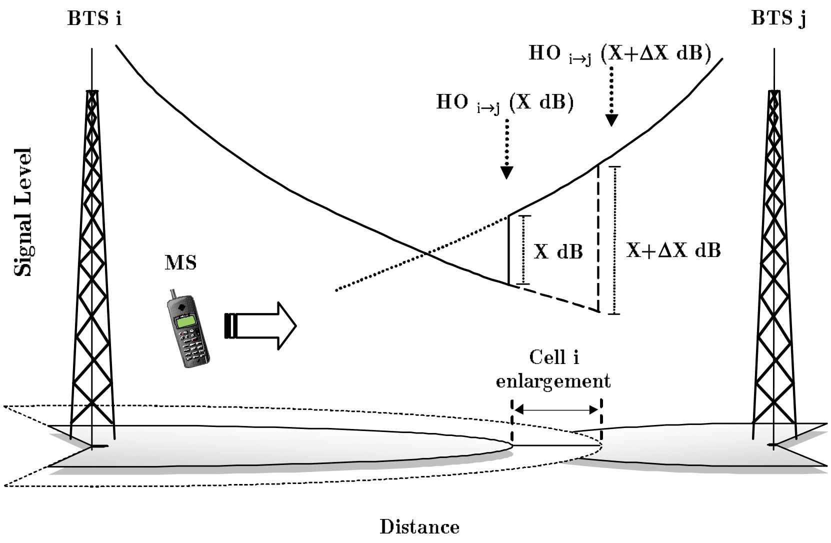


Figure 3.7: Displacement of handover boundary by changing handover margins.

3.2.3 The Tuning of Handover Parameters as an Optimisation Problem

The tuning of HO parameters can be formulated as an optimisation problem, where the main goal is to improve network performance by finding the best HO parameter settings. The following description covers the main elements of the formulation: the performance criteria, the optimised parameters and their relationship. Such a description will be used to classify the underlying optimisation problem.

Performance criteria

When optimising HO margins, several performance criteria must be taken into account. The main goal is to maximise the traffic carried by the network, since it is the main source of operator revenues. At the same time, the impairment of the overall connection quality in the network must be minimised. Finally, the increase of the signalling load associated to an increase of the number of HOs in the network must also be kept within reasonable limits.

Assuming that the overall traffic demand does not change when the network is optimised, the first objective coincides with the minimisation of the number of calls blocked due to lack of traffic resources. Thus, the carried traffic in the network is maximised when the call blocking probability is minimised. In this work, an estimation of the blocking probability is performed by the overall blocking rate in the network, $\overline{BR}$, defined as

$$\overline{BR} = \frac{\sum_{i=1}^{N} N_{b_i}}{\sum_{i}^{N} N_i} = \frac{\sum_{i=1}^{N} N_i \cdot BR_i}{\sum_{i}^{N} N_i} \,, \tag{3.3}$$

where N is the number of cells, N_i and N_{b_i} are the number of offered and blocked call attempts in cell i, and BR_i is the blocking rate in cell i. From (3.3), it is deduced that those cells with a larger number of attempts dominate the overall blocking figure and will thus be the main focus of the traffic sharing process.

The second objective aims at minimising the total connection time where a bad connection quality is experienced. To build an overall network quality indicator, the term 'bad connection quality' must first be defined. In cellular networks, fluctuations of propagation and interference conditions might cause that instantaneous C/I values fall below some desired value. Under these circumstances, the bit error rate (BER) increases significantly and so does the frame error probability (FEP) (i.e., the probability that a frame is in error after decoding) [5]. The period in which the connection experiences unacceptable quality is referred to as a *service outage*. In this work, the outage condition is defined in terms of frame error rate (FER). More specifically, a maximum FER of 5.4% is defined as the minimum acceptable call quality. Unlike traffic demand, the total connection time might change after the optimisation process, as more calls are accepted in the network. To build an indicator that is independent of carried traffic, the number of bad-quality measurements is normalised by the raw number of measurements. Such a figure is referred to as the *outage rate*, as it represents the overall ratio of time where a bad connection quality is experienced. The overall outage rate, $\overline{OR}$, is calculated as

$$\overline{OR} = \frac{\sum_{i=1}^{N} N_{mri}|_{FER \geq FER_{max}}}{\sum_{i=1}^{N} N_{mri}} = \frac{\sum_{i=1}^{N} N_{mr_i} \cdot OR_i}{\sum_{i=1}^{N} N_{mr_i}}\ , \tag{3.4}$$

where N_{mri} and $N_{mri}|_{FER \geq FER_{max}}$ are the total number of measurements and the number of bad-quality measurements in cell i, respectively, OR_i is the outage rate in cell i, and FER_{max} is the FER outage condition (i.e., 5.4%). It should be pointed out that current GSM networks do not provide FER measurements. Thus, operators normally deal with BER values, which are directly applicable to received signal-quality (RXQUAL) measurements [5]. As main drawback, the target RXQUAL value depends on the gain of the decoding process, which, in turn, depends on the frequency hopping scheme implemented in the network. While RXQUAL values from 5 to 7 lead to unacceptable FER values in non-hopping TRXs, only RXQUAL values of 6 and 7 do the same for hopping TRXs [5]. This justifies the choice of FER estimations to evaluate the overall network quality whenever possible.

The last objective aims to reduce the network signalling load by eliminating unnecessary HOs. Despite being a secondary objective, this criterion ensures the stability of the HO process. Instabilities can arise from large deviations of HO parameters from their standard values. As stated previously, a large displacement of PBGT HO boundaries might lead to unacceptable connection quality in the enlarged part of the cell service area. This would cause displaced MSs to be sent back to the original cell by the imperative *quality-reason* HO (QUAL HO), as the latter mechanism has priority over PGBT HO. The interaction between opposing mechanisms might lead to repetitive HOs, which would increase network signalling load for no reason. Obviously, this situation must be avoided. As the focus is not on the absolute signalling load, but on the stability of the HO process, an increase of the signaling load is permitted, provided it is due to a traffic increase. Hence, the tuning process should minimise the average number of HOs per call and not the raw number of HOs in the network.

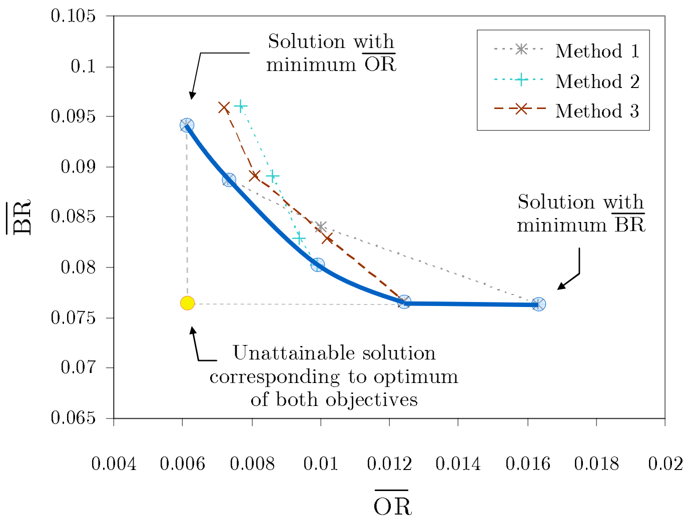


Figure 3.8: An example of the Pareto optimality principle.

From the previous analysis, it is obvious that some of the previous objectives are conflicting. As it is highly unlikely that the same parameter setting results in the best performance for all the objectives, some trade-off between objectives is needed to ensure acceptable performance. Therefore, it is natural to look at the tuning problem as a multi-objective optimisation problem. To evaluate the quality of a solution to a multi-objective optimisation problem, the Pareto optimality principle is used [97]. A solution is said to be *Pareto-optimal* (or *non-dominated*) if there is no other solution that is better in satisfying all of the objectives simultaneously. The set of Pareto-optimal solutions is commonly referred to as the *Pareto front*.

In a problem with two objectives, Pareto optimality can easily be visualised by a scatter plot of solutions, where each performance criteria is drawn on a separate axis. Figure 3.8 illustrates how this type of figure can be used to compare network configurations. Figure 3.8 shows $\overline{BR}$ and $\overline{OR}$ for the network parameter settings obtained by three tuning methods. Each point in the graph represents the performance of a network parameter setting. For easy identification, the different settings generated by the same method are joint by a discontinuous line. From these lines, it is evident that both objectives are conflicting, i.e., the reduction of $\overline{BR}$ is only achieved by an increase of $\overline{OR}$. A trade-off between $\overline{BR}$ and $\overline{OR}$ is therefore needed, regardless of the method employed. Consequently, there is no solution that achieves the minimum value of both performance criteria, which would correspond to the solution shown in the lower-left corner of the figure. However, the existence of different trade-off curves gives evidence that not all the tuning methods behave the same. More important, the crossing of curves indicates that no method outperforms the others both in terms of $\overline{BR}$ and $\overline{OR}$. On the contrary, each method provides some solutions that do not have any solution that show lower $\overline{BR}$ and $\overline{OR}$ simultaneously (i.e., points to the left-down of the figure). This set of non-dominated solutions, which is highlighted by a thick line, is the Pareto front. Conversely, all the solutions above the Pareto front are dominated solutions.

From the previous example, it is clear that a multi-objective optimisation problem has multiple optimal solutions. Hence, the full Pareto-optimal set (or, at least, a wide sample of it) must be evaluated to find the best method. To ease the analysis, the main objectives can be aggregated into a unique cost (or penalty) figure, while the secondary objectives can be handled as constraints. This approach is used in this work to assess the value of the different solutions in the Pareto front. Concretely, the overall blocking and outage rates are included in the objective function, while the average number of HOs per call is handled as an optimisation constraint.

Optimised Parameters

The main decision variables that can be adjusted to modify the service area of cells in a cellular network are the HO margins and HO signal-level constraints. In current networks, both parameters are defined on a per-adjacency basis. Given that the number of adjacencies in an entire operator network is in the order of a million, it is evident that the tuning problem is a *large-scale multi-variate* optimisation problem.

In most manufacturer equipment, HO parameters can only take integer values within certain limits. More specifically, $HoMarginPBGT_{i\to j} \in [-24, 24]$ dB, $RxLevMinCell_{i\to j} \in [0, 63]$, and $HoMarginPBGT_{i\to j}, RxLevMinCell_{i\to j} \in \mathbb{Z}$. In addition to equipment constraints, operators often impose stricter restrictions to the variation of these parameters to prevent large deterioration of the overall connection quality. Therefore, the optimisation problem can be modelled as a *constrained integer* optimisation problem.

To reduce problem size, the tuning problem can be re-formulated so that the number of optimised parameters is minimised. As stated previously, HO margins must be jointly modified in both directions of an adjacency to maintain the hysteresis region. This constraint is satisfied if

$$HoMarginPBGT_{i\to j} + HoMarginPBGT_{j\to i} = 2\sigma_{sf}\ . \tag{3.5}$$

Such an equality constraint can be used to eliminate one of the variables from the problem formulation by substituting

$$HoMarginPBGT_{j\to i} = 2\sigma_{sf} - HoMarginPBGT_{i\to j}\ . \tag{3.6}$$

Thus, the number of parameters to be optimised is halved.

Although the previous approach is valid, the problem can be re-formulated to give a more intuitive idea of the tuning process, while still maintaining the same size. From the operator's perspective, it is often easier to evaluate the new margin values in terms of the original ones. Thus, new values are better understood when expressed relative to the values initially configured in the network. Equations (3.7)-(3.8) show how the optimisation process can be expressed in terms of the deviation from the initial parameter settings, $HoMarginPBGT^{(0)}_{i\to j}$, deployed network wide.

$$HoMarginPBGT_{i\to j} = HoMarginPBGT^{(0)}_{i\to j} + \Delta HoMarginPBGT_{i\to j} \tag{3.7}$$

$$HoMarginPBGT_{j\to i} = HoMarginPBGT^{(0)}_{j\to i} + \Delta HoMarginPBGT_{j\to i} \tag{3.8}$$

From (3.5), it is clear that any shift of margin in one direction of the adjacency is accompanied by the inverse action in the opposite direction, and thus

$$\Delta HoMarginPBGT_{i \to j} = -\Delta HoMarginPBGT_{j \to i} \,. \tag{3.9}$$

From (3.9), Equations (3.7)-(3.8) can be re-formulated as

$$HoMarginPBGT_{i \to j} = HoMarginPBGT^{(0)}_{i \to j} + \Delta HoMarginPBGT_{i \to j} \,, \tag{3.10}$$

$$\begin{aligned} HoMarginPBGT_{j \to i} &= HoMarginPBGT^{(0)}_{j \to i} + \Delta HoMarginPBGT_{j \to i} \\ &= HoMarginPBGT^{(0)}_{j \to i} - \Delta HoMarginPBGT_{i \to j} \,. \end{aligned} \tag{3.11}$$

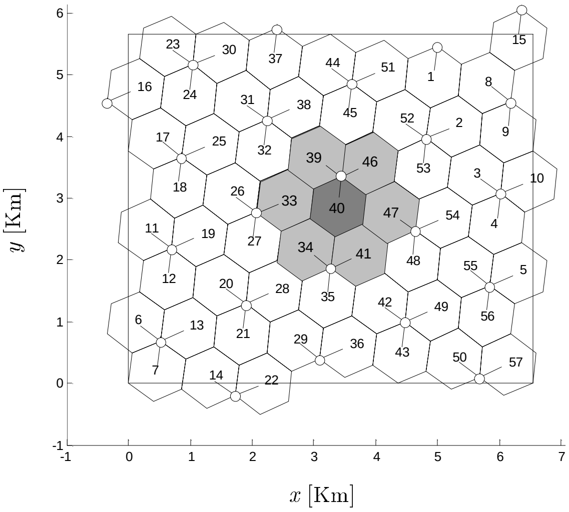
In the previous equations, $\Delta HoMarginPBGT_{i \to j}$ can be interpreted as a biasing term used with opposite sign in the evaluation of PBGT in both directions of the adjacency. These equations show that the tuning problem in an adjacency can be viewed as an *univariate* optimisation problem that aims to find the best setting for the displacement of margin in one direction of the adjacency, $\Delta HoMarginPBGT_{i \to j}$. The size of the problem is also halved, since the values for the two margins can be derived from the value of a single parameter. In doing so, however, care must be exercised to ensure that bounds for the original parameters are not exceeded.

Sensitivity analysis

The sensitivity of network performance to parameter changes can be evaluated by means of simulation models. This sort of analysis aims to identify the class of optimisation problem that is behind the tuning process. Thus, it is also possible to check whether the problem is analytically tractable or not. The following experiments aim to show some key aspects of the tuning of HO margins. These experiments are performed over a network-level simulator of a GSM network. Figure 3.9 (a) depicts the basic scenario consisting of 57 tri-sectorised cells. To aid the interpretation of results, traffic demand and resources are evenly distributed in the scenario. Likewise, the quality HO mechanism is disabled, which is equivalent to neglecting connection quality problems. Figure 3.9 (b) presents the default values for some relevant simulation parameters.

The main goal of the experiments is to check the sensitivity of cell traffic to variations of HO margins. For this purpose, PBGT HO margins in the scenario are initially configured to the value of σ_{sf} (i.e., $HoMarginPBGT^{(0)}_{i \to j} = 8$ dB). Then, PBGT HO margins of a source cell (highlighted in dark grey in Figure 3.9 (a)) are varied to cells in the first ring of adjacencies (in light grey).

Figure 3.10 (a) shows the ratio of carried traffic in source and adjacent cells, A_c, with respect to the one obtained with the original settings, $A_c^{(0)}$, for different margin deviations, $\Delta HoMarginPBGT_{i \to j}$. From the figure, it is clear that an increase (decrease) of traffic in the source cell can be achieved at the expense of a decrease (increase) of traffic in the adjacent cells. Concretely, the traffic in the source cell can be doubled by increasing PBGT HO margins to neighbour cells. Likewise, a five-fold reduction in cell traffic can be achieved by displacing PBGT HO margins in the opposite direction. These results confirm that PBGT HO margins have a strong impact on cell traffic distribution. A closer analysis reveals that most of the traffic sharing capability is achieved when the displacement is larger than $HoMarginPBGT^{(0)}_{i \to j}$ (i.e., 8 dB). This is just the confirmation that setting negative HO margins in any direction of the adjacencies boosts the potential of the method.



(a) Scenario

Path-Loss model	Okumura-Hata
Slow-fading std. (σ_{sf})	8 dB
Mobility model	Random direction
User speed (v_{ms})	3 km/h
Call arrival model	Poisson
Call length model	Exponential
Mean Call Duration (MCD)	80 s
Avg. traffic load (L)	25%
Time resolution	0.48 s
Simulated network time	48000 s

(b) Default simulation parameters

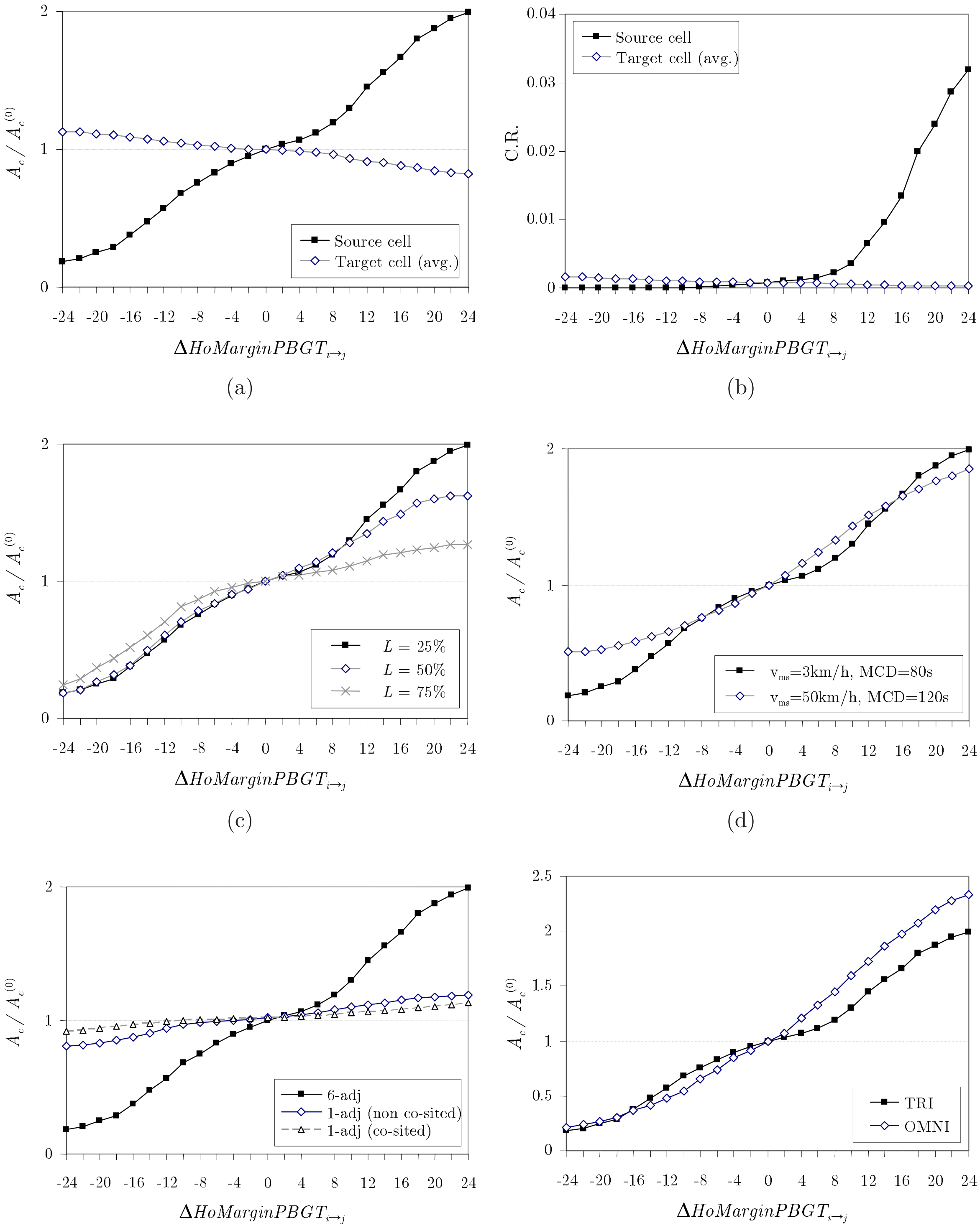
Figure 3.9: System-level simulator.

Figure 3.10 (b) depicts the CR values in source and adjacent cells. While the effect of the tuning process is hardly noticeable in the adjacent cells, it is strongly evident in the source cell. For the source cell, the relationship between CR and margin deviation is non-linear, but still convex. Although this relationship seems simple, it proves extremely dependent on network conditions, as will be shown by the next experiments.

Figure 3.10 (c) presents the influence of network load on the tuning process. For this purpose, the average traffic load in the scenario, L, is varied from 25 to 75% and the traffic carried in the source cell is evaluated for different margin deviations. From the figure, it is evident that, for the same margin settings, the traffic on the source cell heavily depends on network load. As traffic increases in the scenario, the source cell has less resources to accommodate new traffic. Thus, the sensitivity to positive margin deviations is reduced, as the tuning process reaches its saturation point earlier. The saturation effect is not so pronounced for negative margin deviations, as the displaced traffic is shared among several cells. This behaviour is beneficial as congested cells (and not empty cells) are the ones that appear isolated in a live network.

Figure 3.10 (d) presents the influence of user mobility on the tuning process. To reflect opposite mobility conditions, two combinations of user speed and mean call duration are tested: (v_{ms}=3 km/h, MCD=80 s) and (v_{ms}=50 km/h, MCD=120 s). From the figure, it is clear that the tuning effect is smoother in environments with a higher mobility. In contrast, the hysteresis region has a significant impact in low mobility environments. This observation is easily justified for the case of stationary users. Provided that propagation conditions for these users do not change with time, once a stationary user is connected to a serving cell, it will not move to an adjacent cell until margins take negative values. Hence, the need for negative margin values to displace users is more evident in urban environments, where user mobility is reduced.

The next experiment evaluates the influence of the antenna configuration. More specifically, the experiment intends to show that not all adjacencies are equally affected by the tuning process. For this purpose, the HO margins to the six adjacent cells are modified one at a time. For symmetry reasons, there are only two cases of adjacency to be considered, depending on whether the adjacent cell shares site with the original cell or not. These two types of adjacency are referred to as *co-sited* and *non co-sited* adjacencies. Figure 3.10 (e) shows the effect on



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.10: Sensitivity to the variation of PBGT handover margins over a test case.

carried traffic from the tuning process of both types of adjacencies. The overall result of the six adjacencies is also superimposed for comparison purposes (denoted by *6-adj*). From the figure, it is clear that not all adjacencies are equally important. In particular, adjacencies to co-sited cells show a smaller sensitivity to margin deviations. This result is mainly due to the reduced overlapping area caused by the antenna radiation pattern. From Figure 3.10 (e), it might wrongly be inferred that the overall result of the tuning process can be derived from the sum of effects in every single adjacency when it is the only one regulated. On the contrary, the sensitivity to margin deviations in one adjacency is dependent on the current parameter settings in other adjacencies. A closer analysis (not shown here) reveals that the 6-adj curve is not the result of adding twice the 1-adj (co-sited) curve plus four times the 1-adj (non co-sited) curve. This behaviour avoids the separation of the multivariate optimisation problem into multiple univariate problems.

To complete the analysis, every tri-sectorised site is substituted by three omni-directional antennas located in the centre of the corresponding cells. Figure 3.10 (f) shows the comparison of results achieved by the tuning process with both antenna configurations. The most remarkable result is the increased effectiveness for positive margin displacements in omni-directional cells. This result is mainly due to an increased cell overlapping, especially of those cells that were under same site in the old tri-sectorised configuration.

The previous experimental analysis has shown several important properties of the tuning of PBGT HO margins. On the one hand, it has proved that there exists a non-linear relationship between margin deviations and tele-traffic performance indicators. Likewise, it has shown that, in principle, the tuning problem is not separable in an adjacency level, but it has to be considered as a whole. From these two results, the tuning problem can be classified as a *large-scale non-linear multi-variate* optimisation problem. On the other hand, the analysis has shown that the result of the tuning process depends on many factors that are difficult to model. As these factors greatly vary from cell to cell, it can be concluded that the analytical approach is not feasible. Nonetheless, it has been proved that tuning PBGT HO margins on a per-adjacency basis is a powerful technique to re-distribute network traffic when strong cell overlapping exists. The effectiveness of this technique is significantly increased by setting negative margin values, especially in low mobility environments.

3.2.4 Current State of Solution Techniques

Congestion phenomena in mobile networks have received considerable attention both in the academic and commercial literature. Thus, several approaches have been proposed to deal with the source of the problem: the uneven distribution of mobile traffic demand in the spatial and temporal domain. This section outlines the state of research and practice related to the problem. While the former provides a brief survey of methods proposed in the literature to relief network congestion, the latter focuses on methods and tools currently in use in live networks.

State of Research

Load management is a central issue in large distributed systems consisting of many interconnected elements. Thus, much of the existing literature on this topic has been proposed in the context of distributed computing systems and fixed telecommunication networks.

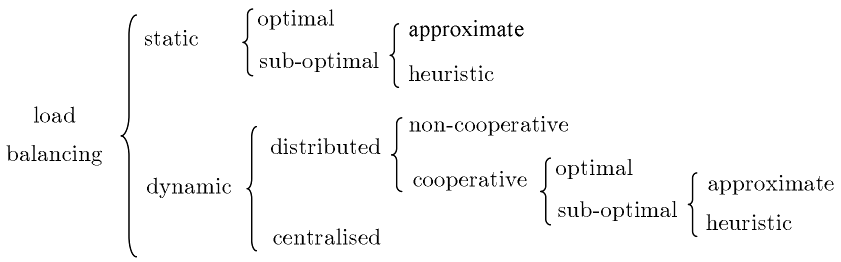


Figure 3.11: A taxonomy of load balancing strategies [101].

In distributed computing systems, jobs do not have stringent time requirements and can be freely diverted to processing elements. Thus, emphasis has been on maintaining the workload evenly distributed among processing elements so as to minimise the mean response times of a job [98]. On the one hand, *load sharing* policies ensure that no processor is idle while adjacent processors are overloaded. On the other hand, *load balancing* policies strive to equalise the load throughout the network to maximise efficiency. Figure 3.11 reflects a broad classification of load balancing strategies in this application domain. *Static* strategies are generally based on information about the system's average behaviour rather than its actual current state, while *dynamic* strategies react to the current state when making transfer decisions. A strategy is *centralised* if the responsability for scheduling resides on a single node, while this decision is physically distributed among network elements in a *distributed* strategy. In the latter case, a strategy is *cooperative* when each element carries out its own portion of the scheduling task, but works toward a global system goal. Finally, *adaptive* strategies change parameters in the load assignment algorithm dynamically according to the current state of the system. It is worth noting that, although it is often assumed that dynamic load balancing outperforms other methods, this is not always true. On the one hand, dynamic load balancing strategies put much higher resource requirements on the system than static load sharing [99]. On the other hand, static strategies can be formulated as classical optimisation problems, for which exact solution techniques exist [100].

Unlike computing systems, service requests in telecommunication networks have strict real-time requirements [102]. Thus, priority has normally been given to keeping system throughput and response times at an appropriate level in all conditions. In this context, *load control* policies regulate the admission of new service requests to ensure low response times for admitted requests. At the same time, dynamic load balancing techniques deal with network delays.

Unfortunately, most algorithms on distributed computing systems and fixed telecommunication networks assume that, albeit delayed, the load can be reallocated in any node of the network. Obviously, this assumption does not hold in the cellular environment, where users can only be served by a small subset of cells providing adequate coverage. Hence, the application of load balancing algorithms from other fields is not immediate. Nonetheless, the different techniques can still be classified following the same taxonomy as *on-line* (i.e., dynamic) and *off-line* (i.e., static) approaches.

Rapid fluctuations in traffic demand are usually dealt with by processes that react instantly to overload situations. These real-time processes are often referred to as *congestion relief* algorithms. After detecting an overload situation, the algorithm executes a series of actions to bring the system out of congestion and avoid system instability. These actions aim to reduce temporarily the traffic load in an overloaded cell, which is normally achieved by either increasing traffic resources or reducing traffic demand. Provided that no free TRX is available in the cell, the number of traffic channels can be momentarily increased by degrading connection quality of active users. At the same time, the traffic demand can be reduced by sending users to surrounding cells that are lightly loaded. If congestion still persists, new call attempts must be blocked (or queued) and on-going connections must be terminated. Obviously, the latter situation should be avoided.

In GERAN, the number of traffic channels in a cell can be increased by means of half-rate coding [103]. *Dynamic half-rate coding* makes use of half-rate coding to accommodate up to twice the number of users with the same hardware resources, but at the expense of a slight call-quality reduction. Although it might seem that permanent application of HR coding might provide an overall capacity enhancement, this is not actually the case [10]. Hence, HR does only provide a capacity benefit from its dynamic allocation as a blocking relief strategy. Unfortunately, the number of TRXs on which this feature can be used is normally limited as operators have to pay for the use of HR codecs on a per-TRX basis [31].

Once the capacity of a cell becomes exhausted, new incoming users have to be sent to surrounding cells to avoid call blocking. *Directed-retry* (DR) [104] directs calls during the set-up phase from the congested cell to a neighbour cell that also provides coverage in the area where the call attempt is made. Although this technique relieves the negative effect of congestion, it does not solve the source of the problem. Thus, very shortly after call establishment, a HO is attempted from the new (i.e., non-optimal) cell to the old (i.e., best-serving) cell. Hence, although this feature reduces call blocking, it only provides little permanent capacity increase.

To make the most of cell overlapping, the re-assignment of users to other cells can be triggered before full congestion is reached. Such a strategy is known as *dynamic load sharing*, for which there exists several implementations. In its simplest form, this strategy can be viewed as an enhancement of DR, where the relief action is triggered earlier [105]. User re-allocation, referred to as a *traffic-reason* HO, might affect both new and on-going connections. Alternatively, user re-allocation can be performed by dynamically changing size and shape of the service area of cells in the network [8][106]. This effect can be achieved by modifying PBGT HO margins dynamically based on current load. The most advanced solutions ensure system stability by monitoring the load of target cells and preventing users from returning back to the original cell as soon as a new channel is available [9].

The above-mentioned reactive methods deal with the randomness of the call arrival and termination processes. However, they are often ineffective to solve persistent congestion problems caused by spatial concentration of traffic demand. Similar to other real-time procedures, these techniques are prone to instability. Thus, operators often use conservative internal parameter settings to avoid instabilities, which greatly reduces the potential of these methods. Hence, local congestion is normally counteracted in the long term by re-planning strategies, such as the extension of the number of TRXs or cell splitting. In the short term, off-line adaptation of service cell area remains the only solution to reduce traffic load in those cells that cannot be upgraded quickly or simply do not justify the addition of resources. The latter case can be the situation where congestion takes place due to seasonal traffic (e.g., in a holiday resort). In this case, even though the lack of traffic resources might persist for some time, no re-planning action is taken by the operator due to effort and expenses.

Several techniques have been reported in the literature to modify the service area of a cell in GERAN. A first group of techniques modify physical BTS parameters, such as the transmitted power [107] or the antenna radiation pattern [108] [109]. As these techniques involve maintenance actions, they are seldom used in practice. Alternative, a second group of methods aims to modify logical RRM parameters, which is far easier. In particular, the tuning of access and HO parameters stands out from all other techniques due to its simplicity and effectiveness. A deeper analysis shows that, although the tuning of CRS parameters is claimed as a promising technique [110], the effect of this technique is restricted to the beginning of user connection. Thus, this adaptation technique suffers from the same problem as DR, i.e., displaced users are handed over back to best-serving cell shortly after connection establishment. For this reason, the modification of HO margins is normally suggested as the means to modify the cell operational area [6][7][8][9].

Proactive (i.e., predictive) approaches have been suggested [6][7] for the tuning of HO margins to deal with persistent congestion problems. In these off-line approaches, a model of the system under optimisation is built based on network statistics. The key process is the modelling of the spatial traffic distribution from mobile measurements in the past. For this purpose, either signal-level measurements sent periodically by the MS [7] or mobile positioning information sent by the MS at call set-up and HO [6] can be used. Over this model, the optimal PBGT HO margins can be found by solving a classical optimisation problem. Thus, the tuning process is driven by a global optimisation criterion (e.g., the total carried traffic in the network), unlike real-time methods, where simple local balancing rules are considered (e.g., minimise the load difference between neighbour cells).

As stated previously, the HO margin tuning problem can be classified as a *large-scale constrained integer multi-variate non-linear multi-objective* optimisation problem. To solve this challenging problem, several simplifications can be performed. First, key performance indicators are aggregated into a scalar objective function by a weighted sum of terms. Likewise, the problem is converted into a continuous optimisation problem by neglecting the integrality constraint. The solution thus obtained could be later rounded off to the nearest integer value in all variables to satisfy the integrality constraint. The impact of the latter action on network performance depends on the sensitivity of the system to parameter changes in the vicinity of the optimum value. As the integrality constraints offer fine granularity of parameter values and the optimisation surface is smooth, the resulting performance impairment should remain relatively small. Following a similar approach, parameter bounds are often neglected and later taken into account by truncating infeasible values to achieve feasibility. Provided that parameter limits are sufficiently large, the performance degradation should remain small. Finally, the size of the problem can be reduced if the number of optimised parameters is minimised. In a real network, not all adjacencies are equally important, but most of the HO traffic is carried by a small number of adjacencies. By restricting the tuning process to the most relevant adjacencies, the optimisation problem is simplified. Nonetheless, there still remain thousands of parameters to be optimised. As a result of all these simplifications, the easiest problem to be solved is a *large-scale unconstrained continuous multi-variate non-linear scalar optimisation problem.*

A wide spectrum of methods exists for unconstrained continuous non-linear optimisation (for a survey, the reader is referred to [70]). A first group of methods, known as *trajectory-based* methods, aim to find *local* minima by iteratively replacing some initial solution by a neighbour one. At each iteration, the new solution is defined by a local search operator, whose behaviour can be either deterministic or random. In contrast, *population-based* methods use of a set of initial solutions that are refined simultaneously. Both the addition of randomness and the handling of several solutions in parallel help to escape from local minima.

Although the previous methods provide excellent results when applied over analytical models, not all of them are equally suitable for their implementation in a live network. Unfortunately, the construction of a precise network model requires tools to collect and analyse mobile measurements that are not currently available for operators. Hence, in most cases, the optimisation algorithm must interact directly with the network. In a real operating network, limited testing can be performed to avoid potential performance degradation. Consequently, only subtle parameter changes are permitted. Such a constraint makes deterministic trajectory-based methods the only viable solution. These methods can be broadly classified depending on whether derivative information is used or not. While *direct search methods* use only function values, and are thus most suitable for problems that are very non-linear or have a number of discontinuities, *indirect search methods* use first- and second-order derivative information, when they are easily calculated, to improve convergence behaviour. Unfortunately, none of the previous approaches is practical for the problem considered here. On the one hand, network performance indicators are subject to random fluctuations, which makes the interpretation of results derived from subtle parameter changes complicated. Under these conditions, numerical computation of derivatives is troublesome. On the other hand, although direct search methods, such as the *Nelder-Mead* method [111], work moderately well for stochastic problems, the number of steps to reach the optimum can be arbitrarily large, even for low-dimensional convex problems. It is worth noting that, even if a precise simulation-based model was available, most of the previous problems would still remain. In a simulation model, the number of iterations would be limited by the computational load of simulations. In the best case, only a few tens of attempts could be made, which is obviously not enough for the large number of parameters to be optimised. As a result, the tuning problem is commonly solved by a simple heuristic approach.

State of Technology

All reactive methods discussed earlier are currently supported by most manufacturer equipment. Nevertheless, as operators have to pay extra fees for these features, few of them are implemented in a live environment. DR is currently commonplace due to its simplicity and effectiveness. In contrast, dynamic load sharing is rarely found due to its implementation complexity and difficulty to find proper settings to achieve stable behaviour. For this reason, off-line parameter optimisation often remains as a last resort to relief congestion in those cells that suffer from unacceptable call blocking after DR. As the analysis tool required in proactive methods is seldom available, operators use naive rules to solve the tuning problem in a live situation.

For simplicity, operators set most parameters to safe standard values that are deployed network wide. These standard settings are provided by manufacturers without considering the peculiarities of each operator's network. Subsequent parameter tuning has to be performed manually after a complex analysis task, which must be carried out on a per-cell basis. For this reason, parameter optimisation is only performed locally for those cells that experience severe quality or congestion problems.

In live networks, HO margins are usually set to large positive values (e.g., 6-8 dBs) to avoid unnecessary HOs in the presence of shadowing. Subsequent parameter modification enables operators to re-shape the operational area of cells to cope with traffic hotspots with existing resources. To restrict undesirable effects, operators commonly set conservative limits to the parameter tuning process, which in turn reduce the overlapping area at the disposal for traffic sharing purposes. In most cases, only positive margin values are set, since the PBGT equation thus ensures that any HO is always to a better cell in terms of signal-level. This practice severely limits the benefits of the tuning process, especially in low mobility environments.

For HO signal-level constraints, values slightly above MS receiver sensitivity are normally set to avoid unnecessary discard of candidate cells. By enlarging the set of candidate cells, a larger macro-diversity gain from HO is obtained in severe shadowing conditions (e.g., in a coverage hole). For instance, HO signal-level constraints in macro-cells fall within the interval [-100, -95] dBm (i.e., $RxLevMinCell \in [10, 15]$). Such a strategy leaves the door open to HOs to cells that do not provide adequate connection quality due to interference reasons. Nonetheless, these loose settings normally give acceptable results when combined with positive margins, since the latter ensure that HOs only take place to better cells. In these conditions, it is margins and not signal-level constraints what ensures adequate connection quality after a HO event. However, this is not the case when HO margins take negative values as a result of the tuning process. In this situation, a safety mechanism must be provided to ensure that no HO is performed to cells that would offer unacceptable connection quality. The previous goal can be achieved by carefully optimising HO signal-level constraints. However, it is worth noting that the minimum signal-level to provide adequate connection quality can greatly vary from cell to cell, as it depends on the actual interference and propagation channel conditions. This explains why fine tuning of signal-level constraints is hardly ever done.

The advent of the first automatic network optimisation tools has boosted the development of off-line tuning algorithms based on the NMS. These applications relieve operators from tedious tasks related to the collection and analysis of data. At the same time, these tools take charge of checking and implementing the suggested parameter changes in the network. As a result, operators can develop tuning methods that would have been unthinkable in the manual approach. Analysis tasks can now involve configuration and statistical data from several tables in the NMS databases. More important, the process can now be repeated for all cells in the network without much effort. In addition, the frequency with which parameters can be modified is increased, which can be used to cope with user mobility trends within a day. While network parameters were modified at most once a week in the past, parameters can now be modified several times a day. However, it should be pointed out that the shorter the data collection period is, the more erratic the tuning process becomes.

3.3 Method of Tuning Handover Parameters

This section describes a set of heuristic methods to adjust several HO parameters in GERAN. Firstly, traffic sharing is justified as a means of enhancing network performance under spatial concentration of traffic demand. The basic traffic balancing rule, which is the core of the tuning method, is then presented. This rule aims to equalise network congestion by adjusting PBGT HO margins on a per-adjacency basis. A second method is described to adapt HO signal-level constraints to uneven interference conditions in the network. Subsequently, both methods are combined into a single method that tunes HO margins and signal-level constraints simultaneously. Finally, the convergence properties of these heuristic approaches are discussed.

3.3.1 Traffic Sharing on a Cell Basis

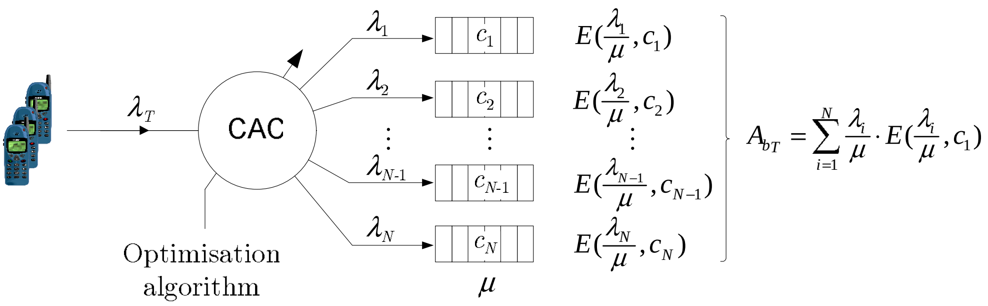
In the previous section, it was concluded that the HO margin tuning problem was a challenging problem both from the theoretical and practical perspectives. In particular, the large set of margin parameters and the absence of a system model prevents computationally-expensive methods from being applied. Consequently, simple heuristic approaches must be adopted, even if optimal performance is not guaranteed.

The simplest approach to face the problem is to break it down into simpler problems of the same type. This can be performed by considering the tuning problem on a per-adjacency basis. Thus, the multi-variate optimisation problem is converted into a number of univariate optimisation problems that can be solved independently. In this approach, the parameter optimised is the deviation of PBGT HO margins from the original settings in every single adjacency, $\Delta HoMarginPBGT_{i \to j}$. Formally, such an approach can be seen as a non-cooperative multi-agent strategy [112] to solve the tuning problem. Distributed problem solving guarantees problem scalability, which is crucial in large networks. Unfortunately, it is difficult to isolate the influence of a single adjacency on the overall network performance, as the performance criterion in a cell (e.g., CR) depends on margins of all its adjacencies. As a result, optimising adjacencies independently might not lead to the global optimum.

In spite of these shortcomings, a simple balancing rule defined on a per-adjacency basis might help to enhance network performance. It is expected that being fair to all cells in the network is beneficial from both the user and system perspectives. Therefore, equalising congestion across the network should lead to better solutions, especially in the presence of severe localised congestion problems. The rest of this section is devoted to justifying the previous assumption. For this purpose, the traffic sharing problem is modelled as a non-linear optimisation problem. Over a simplified model, the convexity of the objective function is shown and the optimality conditions are derived analytically. From this analysis, it is shown that, although balancing congestion on a per-adjacency basis does not lead to the optimal solution, the performance difference is small.

Naive model

The first network model considered in the analysis is shown in Figure 3.12. The network consists of a set of N BTSs that serve call requests from users. These BTSs are heterogeneous in terms of capacity (i.e., each BTS has a different number of channels). The user demand is modelled by a unique flow of calls that can be freely distributed among BTSs (i.e., full overlapping between all cells is assumed). The call arrival process is a time-invariant Poisson process with overall rate λ_T. The service time is a random variable exponentially distributed with intensity $\mu = 1/MCD$, where MCD is the mean call duration. Thus, the total offered traffic in the network (i.e., the carried traffic with unlimited network capacity) is $A_T = \lambda_T / \mu = \lambda_T \cdot MCD$. The assignment of a call to a BTS is performed during call set-up by the call access control (CAC) algorithm and maintained throughout the call. Consequently, the channel holding time, T_h, coincides to the call duration, and the service rate per channel is identical in all network cells, i.e., $\mu_i = \mu = 1/MCD$. Service on each BTS follows an Erlang's loss model (i.e., a call attempt is lost if all channels in the cell are busy). Under these assumptions, the call blocking probability in a cell is given by the Erlang's B formula



λ_T
CAC
Optimisation algorithm
λ_1
λ_2
λ_{N-1}
λ_N
c_1
c_2
c_{N-1}
c_N
μ
$E(\frac{\lambda_1}{\mu}, c_1)$
$E(\frac{\lambda_2}{\mu}, c_2)$
$E(\frac{\lambda_{N-1}}{\mu}, c_{N-1})$
$E(\frac{\lambda_N}{\mu}, c_N)$
$A_{bT} = \sum_{i=1}^{N} \frac{\lambda_i}{\mu} \cdot E(\frac{\lambda_i}{\mu}, c_1)$

Figure 3.12: A naive model of the traffic sharing problem.

$$E(A_i, c_i) = \frac{\dfrac{{A_i}^{c_i}}{c_i!}}{\displaystyle\sum_{j=1}^{c_i} \frac{{A_i}^j}{j!}} , \tag{3.12}$$

where A_i and c_i are the offered traffic and the number of channels in cell i, respectively. The total blocked traffic in the network is the sum of blocked traffic in each cell, which can be calculated as

$$A_{bT} = \sum_{i=1}^{N} A_{bi} = \sum_{i=1}^{N} A_i \, E(A_i, c_i) \, . \tag{3.13}$$

As this work deals with off-line tuning methods, the decision of transferring a call to a BTS does not depend on the current state of the system, but is static in nature. Hence, the goal of the traffic sharing problem is to find the best partitioning of the traffic demand among BTSs so that the total blocked traffic is minimised. The underlying optimisation problem can be formulated as

$$\text{Minimise} \quad \sum_{i=1}^{N} A_i \, E(A_i, c_i) \tag{3.14}$$

$$\text{subject to} \quad \sum_{i=1}^{N} A_i = A_T \, , \tag{3.15}$$

$$A_i \geq 0 \qquad \forall \, i = 1 : N \, . \tag{3.16}$$

(i.e., minimise the total blocked traffic, given that the total offered traffic in the network is A_T and all traffic values are non-negative). In Appendix B.1, it is shown that the optimal solution to this problem must satisfy that

$$E(A_i, c_i) + A_i \frac{\partial E(A_i, c_i)}{\partial A_i} = E(A_j, c_j) + A_j \frac{\partial E(A_j, c_j)}{\partial A_j} \qquad \forall \, i, j = 1 : N. \tag{3.17}$$

The previous equation proves the optimality of the solution achieved by the traffic sharing process. In particular, (3.17) shows that the best performance is obtained when the *incremental blocking probability*, $E(A_i, c_i) + A_i \cdot \frac{\partial E(A_i,c_i)}{\partial A_i}$, is the same for all cells. This conclusion seems contrary to the common practice of equalising network congestion problems. Thus, balancing the blocking probability, $E(A_i, c_i)$, would not lead to the optimal solution, unless the second term in both sides of the equality showed the same dependence on the blocking probability. To discard the latter, (3.17) is developed further by noting that [113]

$$\frac{\partial E(A_i, c_i)}{\partial A_i} = E(A_i, c_i) \left[\frac{c_i}{A_i} - 1 + E(A_i, c_i) \right] . \tag{3.18}$$

Thus, (3.17) is converted into

$$E(A_i, c_i) + A_i E(A_i, c_i) \left[\frac{c_i}{A_i} - 1 + E(A_i, c_i) \right] = \\ E(A_j, c_j) + A_j E(A_j, c_j) \left[\frac{c_j}{A_j} - 1 + E(A_j, c_j) \right] , \tag{3.19}$$

which can be re-written as

$$E(A_i, c_i) \left[1 + c_i - A_i(1 - E(A_i, c_i))\right] = E(A_j, c_j) \left[1 + c_j - A_j(1 - E(A_j, c_j))\right] . \tag{3.20}$$

By noting that $\{c_i - A_i(1 - E(A_i, c_i))\}$ is the average number of free channels in a cell with offered traffic A_i and c_i channels, $N_{fc}(A_i, c_i)$, (3.20) is re-written as

$$E(A_i, c_i) \left[1 + N_{fc}(A_i, c_i)\right] = E(A_j, c_j) \left[1 + N_{fc}(A_j, c_j)\right] . \tag{3.21}$$

As it is well known that the average number of free channels (or, conversely, the average number of busy channels) is not the same for two cells with the same blocking probability but different number of channels, it can be deduced that forcing $E(A_i, c_i) = E(A_j, c_j)$ does not ensure that $N_{fc}(A_i, c_i) = N_{fc}(A_j, c_j)$. Hence, balancing the blocking probability is not the same as balancing the incremental blocking probability, and, consequently, the former action does not ensure optimal performance. In homogeneous networks, all cells have the same number of channels (i.e., $c_i = c_j$). For symmetry reasons, (3.17) have a trivial solution $A_i = A_j$. In these conditions, equalising any traffic indicator leads to the optimal solution. Nevertheless, this is not true for heterogenous networks, where the number of channels can vary from cell to cell. Nonetheless, the following numerical example will show that, although the solutions achieved by different balancing strategies differ significantly, the performance difference is relatively small.

The example considers a network of 3 BTSs with a different number of TRXs. The cell capacity vector is **C**=[22 6 6] TSLs and the offered traffic-to-capacity ratio is 2/3 (i.e., $A_T = (22 + 6 + 6) \cdot \frac{2}{3} = 22.7E$). Four traffic sharing strategies are tested, which aim to equalise the average traffic load (i.e., $L_i = A_i[1 - E(A_i, c_i)]/c_i$), blocking probability (i.e., $E(A_i, c_i)$), blocked traffic (i.e., $A_i\, E(A_i, c_i)$) and incremental blocking probability (i.e., $E(A_i, c_i) + A_i \cdot \frac{\partial E(A_i,c_i)}{\partial A_i}$) in the cells, respectively.

Balancing Criterion	L_i	$E(A_i, c_i)$	$A_i \cdot E(A_i, c_i)$	$E(A_i, c_i) + A_i \frac{\partial E(A_i,c_i)}{\partial A_i}$
A_T [E]			22.7	
$\mathbf{A}$ [E]	[13.9 4.38 4.38]	[16.9 2.89 2.89]	[15.2 3.71 3.71]	[16.2 3.24 3.24]
$\mathbf{L}$	[0.62 0.62 0.62]	[0.73 0.46 0.46]	[0.68 0.56 0.56]	[0.71 0.50 0.50]
$\mathbf{E}$ [%]	[1.16 14.5 14.5]	[4.61 4.61 4.61]	[2.36 9.69 9.69]	[3.55 6.60 6.60]
$\mathbf{A} \cdot \mathbf{E}$ [E]	[0.16 0.64 0.64]	[0.78 0.13 0.13]	[0.36 0.36 0.36]	[0.57 0.21 0.21]
$\mathbf{E}(\mathbf{A}, \mathbf{C}) + \mathbf{A} \cdot \nabla \mathbf{E}(\mathbf{A}, \mathbf{C})$	[0.11 0.47 0.47]	[0.32 0.20 0.20]	[0.19 0.35 0.35]	[0.26 0.26 0.26]
A_{bT} [E]	1.437	1.045	1.079	1.002

Table 3.1: Results of different balancing strategies in the example.

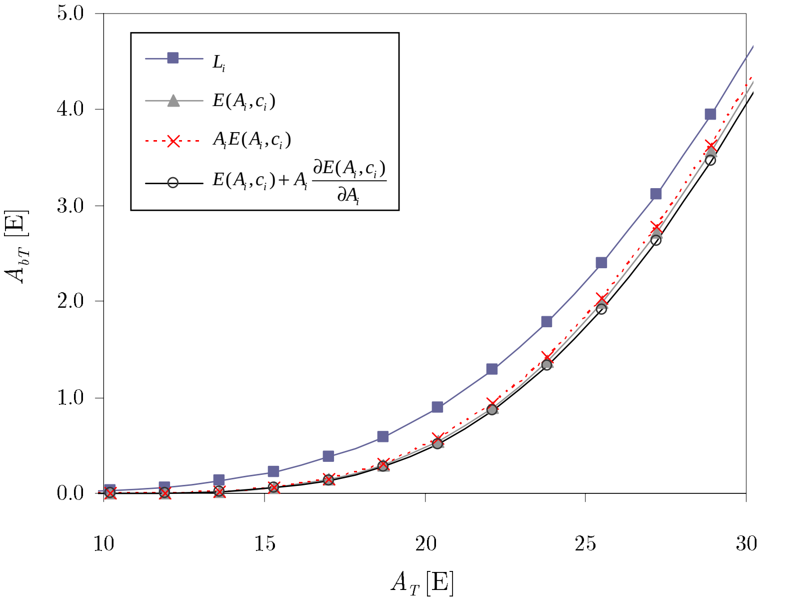


Figure 3.13: Total blocked traffic for different balancing strategies in the example.

Table 3.1 presents the results of the different balancing strategies in separate columns. Each row in the table presents the value of a tele-traffic indicator by showing a vector with the values of the three cells. Obviously, the second and third components of every vector have the same value, as the corresponding cells have the same TSL capacity. Likewise, all cells have the same value of the indicator equalised in each strategy, which is highlighted in grey. From the table, it is clear that the best performance in terms of total blocked traffic (last row) is achieved by equalising the incremental blocking probability (5^{th} column). Nonetheless, it is observed that large imbalances of the latter indicator still give adequate blocking performance. For instance, equalising the blocking probability (3^{rd} column) causes that the incremental blocking probability (7^{th} row) in cells 2 and 3 is 50% larger than in cell 1, but the total blocked traffic only increases by a 4.5%. In contrast, a 44% increase of blocked traffic is obtained by equalising the average traffic load (2^{nd} column), which is performed in most real-time algorithms.

Figure 3.13 confirms that the previous conclusions are valid, regardless of the total traffic. This figure shows the increase of total blocked traffic with total offered traffic for all strategies. From the figure, it is clear that balancing the blocking probability achieves almost the same performance as balancing the incremental blocking probability. In contrast, balancing the average traffic load leads to solutions that perform much worse in terms of total blocked traffic

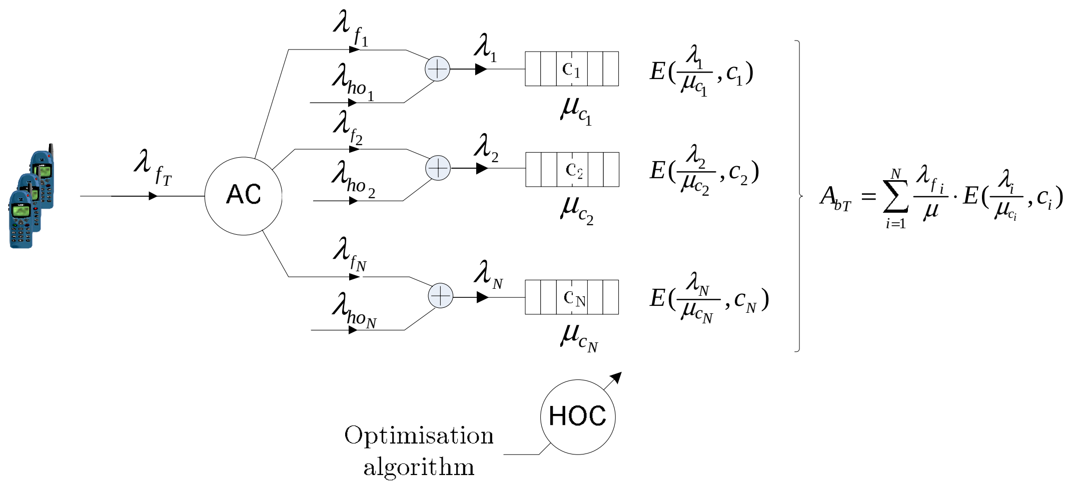


Figure 3.14: A refined model of the traffic sharing problem.

(and, consequently, total carried traffic). Finally, it should be pointed out that, although the actual performance difference depends on the difference in cell capacity, the given example is a realistic case.

Refined model

The previous naive model assumes that users can be freely assigned to BTSs in the network. Likewise, the assignment of calls to BTSs is performed during the call set-up process and not modified later, as it is assumed that all cells can provide adequate coverage during the entire call. Thus, the desired balancing effect only relies on the CAC procedure. Such a model, albeit intuitive, is unable to capture two key issues in the cellular environment: the user mobility and the limited cell coverage.

As a call progresses, the user might leave the service area of the initial cell and enter that of a surrounding cell. The HO process ensures that a user is always connected to the best serving cell. These HO decisions might cause that balancing actions taken by CAC become ineffective, as HO decisions prevail over other mechanisms. Thus, users displaced during call set-up would be handed over back to best-serving cell shortly after connection establishment. To avoid this situation, the service area of a cell should be controlled by tuning HO (and not access) parameters.

The refined model in Figure 3.14 considers the existence of HOs by modelling a call as a series of connections to several BTSs [114]. The total flow of connection requests in a cell is assumed to be a Poisson process of rate $\lambda = \lambda_f + \lambda_{ho}$, where λ_f and λ_{ho} are the arrival rates of new call and HO requests, respectively. The channel holding time in a cell, whether from a new call or a HO, is a random variable that can be modelled by a negative exponential distribution of parameter $\mu_c = 1/MHT$, where MHT is the mean holding time [114].

It is worth noting that the channel holding time does not only depend on user mobility, but is also affected by the cell service area defined by HO parameter settings. Therefore, traffic sharing can now be performed by tuning the handover control (HOC) process. The aim of the tuning process is to minimise the blocking of new calls, which is assumed to be the only source of lost traffic. This goal is achieved by simultaneously controlling λ_{ho} and μ_c on each cell to achieve the values of $A_i = \lambda_i/\mu_{ci} = (\lambda_{f\,i} + \lambda_{hoi})/\mu_{ci}$ that minimise

$$A_{bT} = \sum_{i=1}^{N} A_{bi} = \sum_{i=1}^{N} \frac{\lambda_{f\,i}}{\mu} E(A_i, c_i) = \sum_{i=1}^{N} \frac{\lambda_{f\,i}}{\mu} E(\frac{\lambda_i}{\mu_{c_i}}, c_i) \tag{3.22}$$

(i.e., the total blocked traffic due to the rejection of new calls of average duration $1/\mu$).

Hitherto, it has been assumed that users can be freely assigned to network cells. In a live network, a call can only be assigned to cells providing adequate coverage where the call is originated. This fact limits the maximum offered traffic that can be assigned to every network cell. A lower bound on cell traffic is associated to calls in the area where the cell is the only one providing adequate coverage. An upper bound corresponds to calls performed in the entire cell coverage area. These bounds might limit the capability of the traffic sharing process if low cell overlapping is present in congested areas, causing that the optimal solution to the unconstrained problem could not be reached. Therefore, the new traffic sharing problem can be formulated as the constrained optimisation problem

$$\text{Minimise} \quad \sum_{i=1}^{N} \frac{\lambda_{f\,i}}{\mu} E(A_i, c_i) \quad \text{or} \quad \sum_{i=1}^{N} \lambda_{f\,i} E(A_i, c_i) \tag{3.23}$$

$$\text{subject to} \quad \sum_{i=1}^{N} A_i = A_T \;, \tag{3.24}$$

$$A_{lbi} \leq A_i \leq A_{ubi} \quad \forall\, i = 1 : N, \tag{3.25}$$

where A_{lbi} and A_{ubi} are lower and upper bounds that arise from spatial concentration of traffic demand. In Appendix B.2, it is shown that the optimal solution is the one that satisfies that

$$\lambda_{f\,i} \frac{\partial E(A_i, c_i)}{\partial A_i} = \lambda_{f\,j} \frac{\partial E(A_j, c_j)}{\partial A_j} \tag{3.26}$$

for all cells i, j where constraint (3.25) is inactive[2],

$$\lambda_{f\,u} \frac{\partial E(A_u, c_u)}{\partial A_u}\bigg|_{A_u = A_{ub}} \leq \lambda_{f\,i} \frac{\partial E(A_i, c_i)}{\partial A_i} \leq \lambda_{f\,l} \frac{\partial E(A_l, c_l)}{\partial A_l}\bigg|_{A_l = A_{lb}} \tag{3.27}$$

for all cells l and u where constraint (3.25) is active due to the lower and upper bound, respectively, and

$$\sum_{i=1}^{N} A_i = A_T. \tag{3.28}$$

Using (3.18), (3.26) can be re-written as

[2] An inequality constraint is *inactive* (or *not binding*) when the equality does not hold, and *active* otherwise.

$$\lambda_{f_i} E(A_i, c_i) \left[\frac{c_i - A_i(1 - E(A_i, c_i))}{A_i} \right] = \lambda_{f_j} E(A_j, c_j) \left[\frac{c_j - A_j(1 - E(A_j, c_j))}{A_j} \right], \tag{3.29}$$

which can be simplified into

$$\lambda_{f_i} E(A_i, c_i) \frac{N_{fc}(A_i, c_i)}{A_i} = \lambda_{f_j} E(A_j, c_j) \frac{N_{fc}(A_i, c_i)}{A_j} \ . \tag{3.30}$$

From (3.26), it can be deduced that the optimal solution is reached when $\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i}$ is the same for all cells. Likewise, (3.27) suggests that, in those cells where one of the traffic bounds is reached during traffic sharing, traffic has to be fixed to the limit value and the traffic excess (or defect) re-distributed among the remaining cells. This fact justifies that sharing the traffic between adjacent cells leads to the optimal solution even in the presence of traffic constraints.

Solution technique

The non-linear equation system in (3.26) and (3.28) can be solved as a least-square optimisation problem by minimising the norm of the residual

$$||r|| = \sum_{i=1}^{N} \sum_{j=1}^{N} \left[\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} - \lambda_{f_j} \frac{\partial E(A_j, c_j)}{\partial A_j} \right]^2 + \left[\sum_{i=1}^{N} A_i - A_T \right]^2 \tag{3.31}$$

through an *iterative descent method* [115]. In these methods, a sequence of solutions $A^{(0)}, A^{(1)}, \cdots, A^{(n)}$ is computed by the formula

$$A^{(n+1)} = A^{(n)} + \beta^{(n)} \cdot g^{(n)}, \tag{3.32}$$

where $g^{(n)}$ is a vector indicating some direction of decrease of the objective function (i.e., $||r||$) and $\beta^{(n)}$ is an iterative parameter that indicates the step-length in the direction of $g^{(n)}$. A natural choice of $g^{(n)}$ is that opposite to the gradient of the objective function in the current solution, while $\beta^{(n)}$ is selected such that the maximum of the objective function in the selected direction is reached. These choices correspond to the *steepest descent* method. Unfortunately, the gradient of the objective function in a large-scale problem is difficult to estimate, especially if performance indicators are subject to some uncertainty due to statistical variation. Thus, heuristic approaches remain the only viable solution to find directions of decrease.

It is worth noting that (3.31) considers all possible pairs of cells $(i,\ j)$. This implies that one equation like (3.26) has been included for each pair of cells, which might seem unnecessary as $(N - 1)$ pairs that covered the N cells would retain the same information. However, this formulation naturally introduces diffusive load-balancing algorithms covered in the next section.

3.3.2 Tuning of Handover Margins

From the previous analysis, it can be concluded that a cellular network performs best when blocking problems are distributed across cells. Consequently, reducing large blocking differences by re-distributing traffic between neighbour cells should lead to the best solutions. This effect can be achieved by a local balancing rule that equalise some congestion indicator on a per-adjacency basis. Under normal conditions, repeated application of this local balancing rule should make the traffic distribution converge to the optimal solution, even if the traffic is spatially localised. Such an approach of re-allocating traffic is referred to as a *diffusive* method, which can be seen as a means to reach optimality by minimising the residual norm in (3.31).

Tuning algorithm

To minimise (3.31), the local balancing rule in the adjacency (i,j) would suggest that the offered traffic transferred from cell i to cell j in iteration n was

$$\delta A_{i\to j}^{(n)} = \beta^{(n)} \cdot \left[\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} - \lambda_{f_j} \frac{\partial E(A_j, c_j)}{\partial A_j} \right]^{(n)}, \tag{3.33}$$

where $\beta^{(n)}$ is the *diffusion parameter* that controls the magnitude of changes in each iteration. The new offered traffic value in cell i can be computed by aggregating the traffic flow to all its neighbour cells, $V(i)$, as

$$A_i^{(n+1)} = A_i^{(n)} - \sum_{j\in V(i)} \delta A_{i\to j}^{(n)} = A_i^{(n)} + \sum_{j\in V(i)} \delta A_{j\to i}^{(n)}, \tag{3.34}$$

where $V(i)$ is the set of neighbours of cell i, and superscripts (n) and $(n+1)$ denote the current and next iteration. To make the tuning process more intuitive, (3.33) is modified so as to equalise the total blocked traffic between neighbour cells as

$$\delta A_{i\to j}^{(n)} = \beta^{(n)} \cdot \left[\frac{\lambda_{f_i} E(A_i, c_i) - \lambda_{f_j} E(A_j, c_j)}{\lambda_{f_i} + \lambda_{f_j}} \right]^{(n)}, \tag{3.35}$$

where λ_{f_i} and $E(A_i, c_i)$ can be extracted from counters in the NMS.

In the previous formulas, it has been assumed that the tuning algorithm has direct control over the offered traffic values per cell, A_i, which have been the decision variables so far. Unfortunately, the tuning algorithm can only influence the offered traffic through PBGT HO margins. Therefore, the algorithm must achieve the balancing goal by modifying these parameters progressively. In each iteration, the new margin values in both direction of an adjacency are calculated by computing a single increment

$$\delta HoMarginPBGT_{i\to j}^{(n+1)} = -\beta^{(n)} \cdot \left[\frac{\lambda_{f_i} E(A_i, c_i) - \lambda_{f_j} E(A_j, c_j)}{\lambda_{f_i} + \lambda_{f_j}} \right]^{(n)}, \tag{3.36}$$

where $\delta HoMarginPBGT_{i\to j}^{(n+1)}$ is the change from the previous margin settings. The overall negative sign indicates that the margin value from i to j must decrease when call blocking in cell i is larger than in cell j. From (3.36), it can be deduced that the change in the opposite direction of the adjacency, $\delta HoMarginPBGT_{j\to i}^{(n+1)}$, has the same magnitude, but opposite sign, to maintain cell overlapping. Hence,

$$HoMarginPBGT_{i\to j}^{(n+1)} = HoMarginPBGT_{i\to j}^{(n)} + \delta HoMarginPBGT_{i\to j}^{(n)} \, , \tag{3.37}$$

$$HoMarginPBGT_{j\to i}^{(n+1)} = HoMarginPBGT_{j\to i}^{(n)} - \delta HoMarginPBGT_{i\to j}^{(n)} \, . \tag{3.38}$$

As the above-described method applies the discrete control rule in (3.36) to modify $HoMarginPBGT_{i\to j}$ based on statistical measurements of $\lambda_{f\,i}, \lambda_{f\,j}, E(A_i, c_i)$ and $E(A_j, c_j)$, this method will be referred to as *Slow HO Margin Control* (SHMC).

Influence of RRM Features

In the previous discussion, it has been assumed that a call is blocked when all traffic channels are busy in the cell where the call attempt is made. However, in a live network, these incoming calls can be re-directed to neighbour cells with spare capacity by means of DR. This mechanism has been neglected in the balancing rule so far. Thus, $E(A_i, c_i)$ and $E(A_j, c_j)$ in (3.36) represent BR before DR (i.e., probability that a call is initially blocked, regardless of whether it is later re-directed or not). Hence, it might be argued that, to reduce the number of calls actually blocked, the balancing rule should consider instead BR after DR (i.e., probability that a call is finally blocked). While the former approach aims at equalising congestion problems in the network, and could be considered as a proactive approach, the aim of the latter would be to balance pure blocking problems, which is rather a reactive approach. As will be shown later, it is beneficial to equalise congestion problems in the network, even if real call blocking does not exist. By doing so, the number of calls that are re-directed is minimised, which can enhance network quality. For this reason, the tuning algorithm in this work is based on BR before DR, even if the final assessment figure for blocking performance is BR after DR.

3.3.3 Tuning of Handover Signal-Level Constraints

Under normal conditions, the PBGT HO ensures that an MS is connected to the cell that provides the minimum pathloss (or, equivalently, the maximum signal-level). Once HO margins are set to negative values for traffic sharing purposes, an MS might be assigned to a cell providing worse signal level than the serving cell. In this situation, it is critical to set HO signal-level constraints properly to avoid bad connection quality. For this purpose, the following method tunes these parameters on a per-cell basis based on statistical network measurements. For clarity, the required network statistics are described first and the tuning algorithm is then presented.

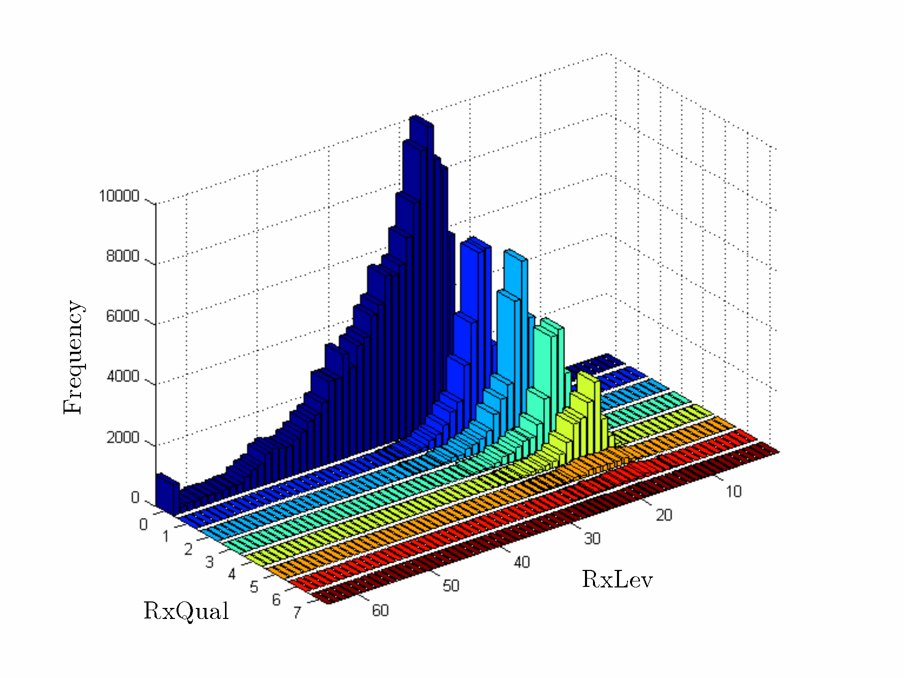


Figure 3.15: An example of *RxLev-RxQual* statistics.

RXLEV-RXQUAL Statistics

The proposed method relies on measurements performed by MS and BTS. In GERAN, measurement reports (MRs) are sent from MS and BTS to the BSC every 0.48s [116]. This piece of information comprises signal level (i.e., RXLEV) and signal quality (i.e., RXQUAL) measurements on both uplink (UL) and downlink (DL) for active connections. The collection of these measurements on a cell basis provides a means to obtain the relationship between the received signal level and the perceived connection quality in any particular cell. Such a relationship can be used to identify interference and propagation peculiarities of each cell.

Some vendors provide these measurements in the form of tables, which are referred to as *RxLev-RxQual Statistics* [117]. An example of such statistics is depicted in Figure 3.15. This illustration shows a 2-D histogram of (RXLEV, RXQUAL) samples extracted from a network simulator. Specifically, the displayed statistics belong to the DL of the beacon TRX of a cell. The x-axis represents RXLEV values ranging from 0 (bad) to 63 (good), the y-axis represents RXQUAL values ranging from 0 (good) to 7 (bad), and the z-axis represents number of samples. At first glance, it is evident that not all RXLEV values appear with the same frequency. On the contrary, most MRs display RXLEV values in the range 15-30, which correspond to MSs near the cell boundary. Likewise, it is observed that several RXQUAL values are possible for a certain RXLEV value. As interference level is not constant in time and space, the relationship between signal-level and connection quality in a cell is not deterministic but probabilistic (i.e., for a certain RXLEV, different RXQUAL values are possible, depending on current interference and propagation conditions). Nonetheless, it is evident that a higher RXLEV leads to a higher probability of acceptable RXQUAL. Thus, measurements with high RXLEV values permanently display RXQUAL 0 (i.e., the best connection quality), while those with low RXLEV values might occasionally experience RXQUAL values up to 7 (i.e., the worst connection quality).

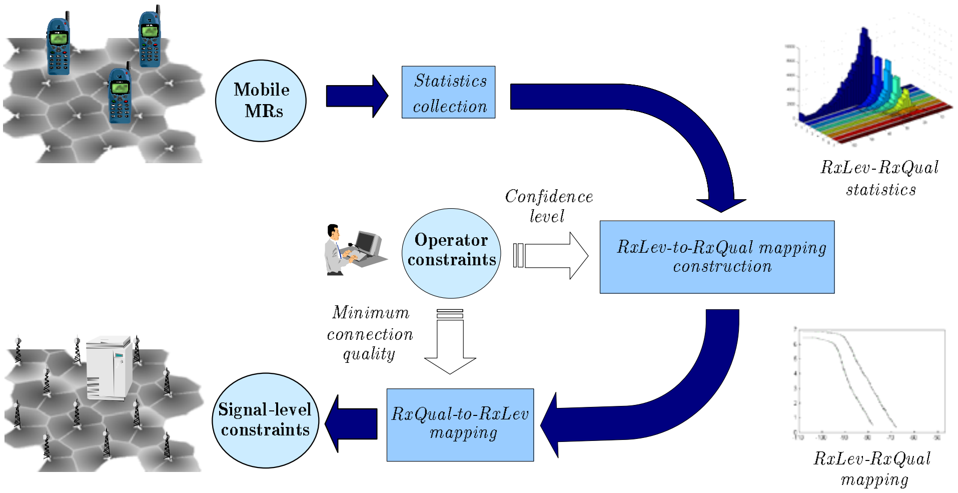


Figure 3.16: The tuning of HO signal-level constraints.

Tuning Algorithm

Figure 3.16 outlines the basics of the tuning algorithm. The process begins with the collection of MRs to build the *RxLev-RxQual* statistics. As stated previously, these statistics show the histogram of the jointly distributed random variables RXLEV and RXQUAL on a per-cell basis. From such information, a one-to-one correspondence between received signal level and predicted connection quality can be derived for each cell. This mapping function, which is the core of the algorithm, is referred to as *RxLev-RxQual mapping* function. By means of this function, the algorithm computes the minimum signal level that ensures the minimum connection quality defined by the operator. Finally, the HO signal-level constraint is set to the resulting value.

To find the function that relates RXLEV and RXQUAL in a cell, the stochastic nature of this relationship must be addressed. For this purpose, the algorithm builds the intermediate functions in (3.39)-(3.41) by treating RXLEV and RXQUAL as random variables. Firstly, the probability of every (RXLEV, RXQUAL) pair is estimated. Thus, the joint PDF of RXLEV and RXQUAL is built by normalising the frequency of every (RXLEV, RXQUAL) pair to the total number of samples. Then, the probability of experiencing a signal quality in the connection, given that a certain signal level is received, is determined. Thus, the PDF of RXQUAL conditioned to RXLEV is obtained by normalising to the number of MRs with signal level equal to RXLEV. Finally, the probability of experiencing a signal quality equal or better than RXQUAL for every signal level is determined. Thus, the CDF of RXQUAL conditioned to RXLEV is extracted by the cumulative sum in the quality axis in every signal-level band.

$$
\begin{aligned}
pdf_{RxLev,RxQual} &= P(\mathrm{RXLEV} = RxLev, \mathrm{RXQUAL} = RxQual) && (3.39)\\
pdf_{RxQual/RxLev} &= P(\mathrm{RXQUAL} = RxQual \mid \mathrm{RXLEV} = RxLev) && (3.40)\\
cdf_{RxQual/RxLev} &= P(\mathrm{RXQUAL} \leq RxQual \mid \mathrm{RXLEV} = RxLev) && (3.41)
\end{aligned}
$$

Figure 3.17 illustrates an example of the $cdf_{RxQual/RxLev}$ surface. Such a surface represents the probability of experiencing a signal quality better than *RxQual*, provided that a certain signal level *RxLev* is received. If a minimum signal quality is fixed as a requirement, a vertical

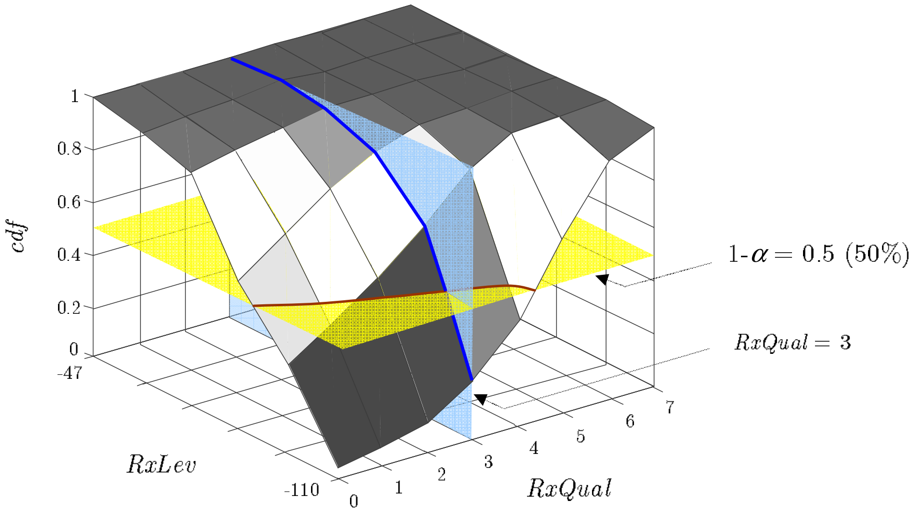


Figure 3.17: An example of $cdf_{RxQual/RxLev}$ surface.

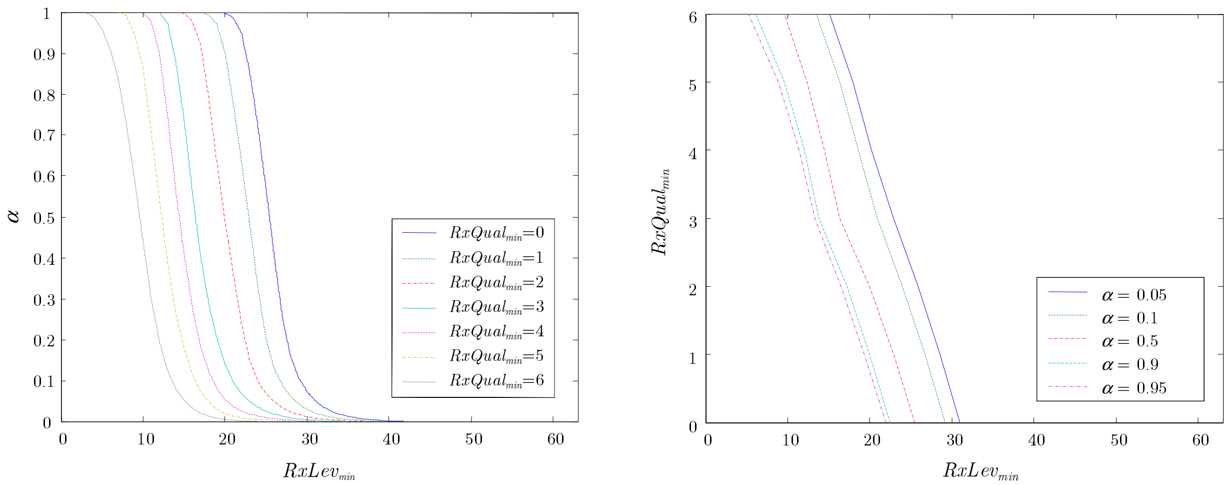
plane parallel to the XZ plane is defined (e.g., $RxQual$=3 in the figure). The intersection line between the previous plane and the surface under study represents the probability of reaching at least the predefined signal quality for every signal level. This probability can be viewed as the confidence of ensuring the connection quality target. Thus, a higher probability means that a higher ratio of samples will satisfy the predefined quality target. On the other hand, whenever a confidence level is desired, a horizontal plane parallel to the XY plane is defined (e.g., $cdf_{RxQual/RxLev} = 0.5$ in the figure). In this case, the intersection line between the surface and the plane represents the minimum signal quality that is ensured for every signal-level with that confidence level. For convenience, the confidence level is hereafter denoted by its reciprocal quantity, the significance level, α.

From the $cdf_{RxQual/RxLev}$ surface, it is straightforward to find the minimum RXLEV value that ensures that the probability that signal quality is not acceptable is below a certain threshold. Such a minimum value, $RxLev_{min}$, estimated on a per-cell basis, is defined as

$$RxLev_{min} = \min \{ x \mid P(\text{RXQUAL} > RxQual_{min} \mid \text{RXLEV} = x) \leq \alpha \} \tag{3.42}$$

(i.e., minimum RXLEV that ensures that the probability of RXQUAL being worse than $RxQual_{min}$ is less than α). In (3.42), $RxQual_{min}$ and α are internal parameters that define the outage condition in terms of RXQUAL and the target outage probability, respectively.

It is thus clear that the tuning process is controlled by the latter parameters. By fixing one of these, a mapping between the remaining parameter and $RxLev_{min}$ is defined. Such mappings will be referred to as *RxLev-Significance* and *RxLev-RxQual* curves, examples of which are given in Figure 3.18. In Figure 3.18 (a), the minimum signal quality, $RxQual_{min}$, has been fixed to values from 0 to 6. Thus, seven different $RxLev\text{-}Significance$ curves are obtained. For a certain value of $RxQual_{min}$, it is observed that the lower (i.e., the more restrictive) α, the higher the $RxLev_{min}$. Likewise, raising $RxQual_{min}$ leads to a decrease in $RxLev_{min}$, which is evident from the fact that curves shift to the left with increasing (i.e., worse) $RxQual_{min}$. Finally, it is worth noting that the slope of the curves is rather steep, i.e., changes of 4-5 dBs in signal level might cause that the outage probability varies from 0.9 to 0.1. This effect is due to the rapid



(a) *RxLev-Significance* (b) *RxLev-RxQual*

Figure 3.18: The relationship between signal level, signal quality and significance level.

decrease of BER with increasing C/I. In Figure 3.18 (b), it is the target outage probability, α, that has been fixed to values from 0.05 to 0.95. As a result, a family of *RxLev-RxQual* curves is obtained. In every curve, it is observed that the lower (i.e., better) $RxQual_{min}$, the higher the $RxLev_{min}$. Unlike *RxLev-Significance* curves, *RxLev-RxQual* curves show a moderate slope, i.e., a deviation of 15-20 dB in signal level is needed to cover the whole $RxQual_{min}$ range. From Figure 3.18, it is evident that choosing different values of $RxQual_{min}$ and α leads to different mapping curves. Such constraints may be used to tailor the behaviour of the algorithm. In this work, $RxQual_{min}$=4 and 5 for non-hopping and hopping TRXs, respectively, to provide similar FER values after decoding [5]. However, the target outage probability, α, must still be defined by the operator. As the above-described method builds a model of the system to be optimised, it will be referred to as *Optimisation of Signal-Level Constraints* (OSLC).

Influence of RRM Features

As stated previously, *RxLev-RxQual* statistics reflect the interference and propagation conditions in a cell. Thus, any network feature deployed to reduce interference has an impact on these statistics that might affect the behaviour of the algorithm. The following paragraphs describe how the algorithm deals with these issues.

The first issue that must be addressed is the link and TRX dependence. As network features are not always enabled in all TRXs of a cell, interference levels may differ from TRX to TRX. For the same reason, DL interference conditions do not necessarily coincide to those of the UL. For monitoring purposes, *RxLev-RxQual* statistics in a cell are usually broken down on a per-TRX and per-link basis. However, if the above-described algorithm was applied to every single case independently, it might end up suggesting different signal level-constraints for each case. Obviously, such a result is not valid, as it contradicts the way the parameter is defined. The signal-level constraint is a parameter that is defined on a per-cell basis and must therefore be shared by all connections (i.e., TRXs and links) in a cell. Hence, the algorithm must consider all the cases simultaneously to derive a single value for the parameter.

To avoid the link dependence, the optimisation procedure is restricted to the DL, as it is assumed that the DL is the most restrictive link in GERAN. This assumption is valid in urban scenarios, where interference is the main limitation and diversity techniques are normally implemented in the UL [10]. Obviously, traffic sharing is more likely to take place on these scenarios and, therefore, it is in these scenarios where it is expected that optimising signal-level constraints has a larger impact.

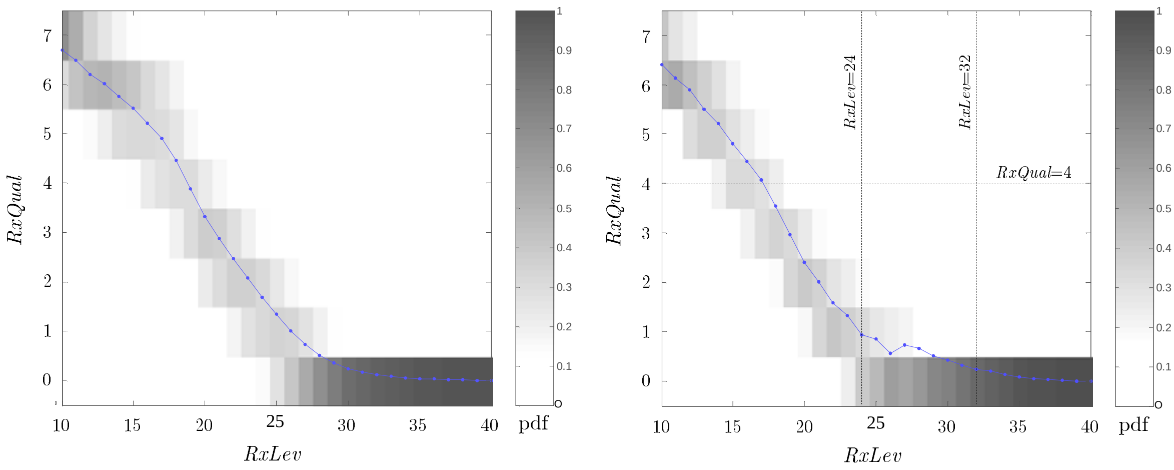
The TRX dependence can be eliminated by aggregating measurements in all TRXs as if they came from a single TRX. This action can be performed because the tuning process only has to ensure that a certain share of measurements show acceptable quality, regardless of the TRX they belong to. At this point, it is worth noting that the decision of accepting a user in a cell is based on the signal level received from the TRX where the BroadCast CHannel (BCCH) is located (commonly referred to as BCCH-TRX), regardless of the TRX the connection is finally assigned to. By aggregating all TRXs, it is implicitly assumed that all TRXs in a cell have the same maximum transmitted power. Thus, the received signal level from the BCCH-TRX is a good estimation of the signal level from any TRX in the cell. If this is not the case, an offset term has to be included to compensate for differences in transmitted power between TRXs.

A closer analysis is required to explain how the algorithm circumvents the effect of the power control (POC) feature. In GERAN, the POC algorithm aims to maintain the signal level and the signal quality of each connection among certain values. This behaviour has a significant impact on the way the received signal level and perceived signal quality are related. To evaluate the impact of POC on a Traffic CHannel (TCH) TRX, network simulations were performed in a homogeneous scenario. In the example, the POC algorithm was configured to maintain RXLEV between 24 and 32 and RXQUAL below 4. Figure 3.19 (a)-(b) represents the $pdf_{RxQual/RxLev}$ before and after enabling POC by a colormap. Figure 3.19 (a) shows that, when POC is disabled, the median value of the RXQUAL distribution steadily decreases (i.e., enhances) with RXLEV. In contrast, Figure 3.19 (b) shows that, after enabling POC, the same is not true in the region of the RXLEV-RXQUAL plane defined by POC parameters. For an easy identification, this region has been enclosed by dashed lines. In this region, a high RXLEV might be due to a high (i.e., bad) RXQUAL, while a low RXLEV is an indicator of low (i.e., good) RXQUAL. It is then clear that RXLEV depends on RXQUAL when POC is active, whereas RXQUAL depends on RXLEV when POC is inactive (i.e., outside POC region). From this observation, it is easily understood that the influence of POC over the OLSC algorithm can be avoided if the signal-quality target and the final value of the signal-level constraint are in the region where POC is inactive (i.e., where TRXs are transmitting at full power).

Practical Issues

Although the above-described method is a powerful tool to adapt to network irregularities by optimising signal-level constraints on a cell basis, some issues are still open.

RXLEV values on *RxLev-RxQual* statistics are discretised in the BSC to reduce the amount of information in network databases. Thus, only a limited number of RXLEV intervals are available in vendor equipment (typ. 6). To maintain resolution in the region of interest, the intervals can be unevenly distributed on the signal-level axis. Thus, the impact of discretisation can be minimised by a proper definition of interval limits. In a well designed network, the final value of the signal-level constraints is expected to vary not more than 10-12 dBs from cell to cell. Consequently, inaccuracies due to a lack of resolution in RXLEV should remain relatively small.



(a) No POC

(b) POC

Figure 3.19: The influence of power control in signal-level and signal-quality relationship.

As small changes on the signal-level constraints have a limited effect on network performance, it is assumed hereafter that inaccuracies from discretisation can be neglected. Hence, the rest of the analysis does not consider the discretisation of RXLEV.

Even if the algorithm is able to calculate signal-level constraints from network measurements, the value of the target outage probability, α, must still be selected. A restrictive value might lead to excessive signal-level constraints, reducing the number of cells in the HO candidate list. Under these conditions, both macro-diversity and trunking gain would be unnecessarily reduced. On the other hand, a loose value might prove ineffective to avoid HOs to a bad cell. This selection problem is addressed by the fuzzy-logic method presented next.

3.3.4 Fuzzy Tuning of Handover Margins and Signal-Level Constraints

The above-described algorithms for tuning HO margins and signal-level constraints can be combined into a single method. For convenience, such a method in which both algorithms run simultaneously (but independently) is referred to as *Slow HO Margin Control* and *Optimisation of Signal-Level Constraints* (SHMC+OSLC). In this method, PBGT HO margins are modified on a per-adjacency basis to equalise blocked traffic between neighbour cells and HO signal-level constraints are adjusted on a per-adjacency basis based on interference in the target cell.

The effectiveness of the previous approach can be improved if both algorithms are synchronised. The method proposed here aims to solve some of the issues of the previous algorithms, which are summarised as follows:

1) In (3.36), the HO margin increment depends only on the difference of blocked traffic between neighbour cells, regardless of the current margin values in the adjacency. However, the analysis presented in Figure 3.10 showed that network sensitivity largely increases when HO margins become negative. In these situations, it may be necessary to reduce the magnitude of changes to ensure stability. This can be achieved by adjusting the diffusion parameter, β, which can be interpreted as a gain-scheduling mechanism.

2) To set the new HO margin values, only blocking performance has been taken into account so far. Hence, interference issues have been neglected. However, the neighbours of a serving cell can potentially be interfered by the latter when cells share frequencies. Although this situation is avoided in the BCCH-TRX by careful planning, it is not the case for the rest of TRXs, for which tight frequency reuses are normally used. This is not an issue whenever HO margins are positive. From (3.2), it can be deduced that the PBGT value gives and estimation of the minimum C/I after HO if frequency collision occurs between source and target cell. Thus, the margin value is the C/I value that would be experienced when both cells are transmitting at full power. Hence, positive margin settings ensure positive C/I values after HO, even in the case of frequency collision. Obviously, this is not the case for negative margin values. In this case, it might happen that an MS sent to a neighbour cell experienced bad quality due to interference from the old (i.e., best serving) cell. Hence, it is clear that the constraints on margins should be tighter for potentially interfered neighbours. Thus, non-interfered neighbours should be favoured during traffic sharing, as their margins can be adjusted further. This can be achieved by setting the minimum PBGT HO margin on a per-adjacency basis. The aim of this constraint is to ensure that acceptable connection quality is experienced even in the presence of frequency collision. Thus, the tuning process must ensure that the C/I is above some threshold defined as the outage condition, $(C/I)_{min}$. For this purpose, the probability that a serving cell i interferes to a neighbour cell j, $p_{c_{i \to j}}$, is calculated as [118]

$$p_{c_{i \to j}} = \begin{cases} 0 & \text{if } N_{f_{i,j}} = 0, \\ \frac{a_f}{N_{f_{i,j}}} \frac{A_{c_i}}{N_{ts_i}} & \text{if } N_{f_{i,j}} > 0, \end{cases} \tag{3.43}$$

where a_f is the service activity factor, $N_{f_{i,j}}$ is the number of frequencies shared by cells i and j, A_{ci} is the carried traffic in the serving cell and N_{tsi} is the number of TSLs for traffic purposes in the serving cell. For a non-hopping TRX, $N_{f_{i,j}}$ is 0 for non-interfered cells and 1 for interfered cells. For a hopping TRX, $N_{f_{i,j}}$ is the number of frequencies in the frequency hopping list. In case of several TRXs in the serving cell, a weighted sum is used to account for the different probability of interfering collision across TRXs as

$$p_{c_{i \to j}} = \frac{\sum\limits_{k=1}^{N_{trx}} A_{c_{i,k}} p_{c_{i \to j,k}}}{\sum\limits_{k=1}^{N_{trx}} A_{c_{i,k}}} , \tag{3.44}$$

where the subscript k denotes the TRX index and N_{trx} is the number of TRXs in the serving cell. At this point, it is worth noting that the tuning process is not interested in the instantaneous C/I values, but only in the average. Thus, the average C/I after considering the gain of frequency hopping, $\overline{\frac{C}{I}}$, can be computed as [118]

$$\overline{\frac{C}{I}} = \left(\frac{C}{I}\right)_{\text{collided TSL}} - 10 \log p_{c_{i \to j}} = \mathit{HoMarginPBGT}_{i \to j} - 10 \log p_{c_{i \to j}} , \tag{3.45}$$

where all terms are in dBs. From (3.45), it follows that

$$HoMarginPBGT_{i \to j} = \frac{\overline{C}}{I} + 10 \log p_{c_{i \to j}} \geq \left(\frac{C}{I}\right)_{min} + 10 \log p_{c_{i \to j}} \ . \quad (3.46)$$

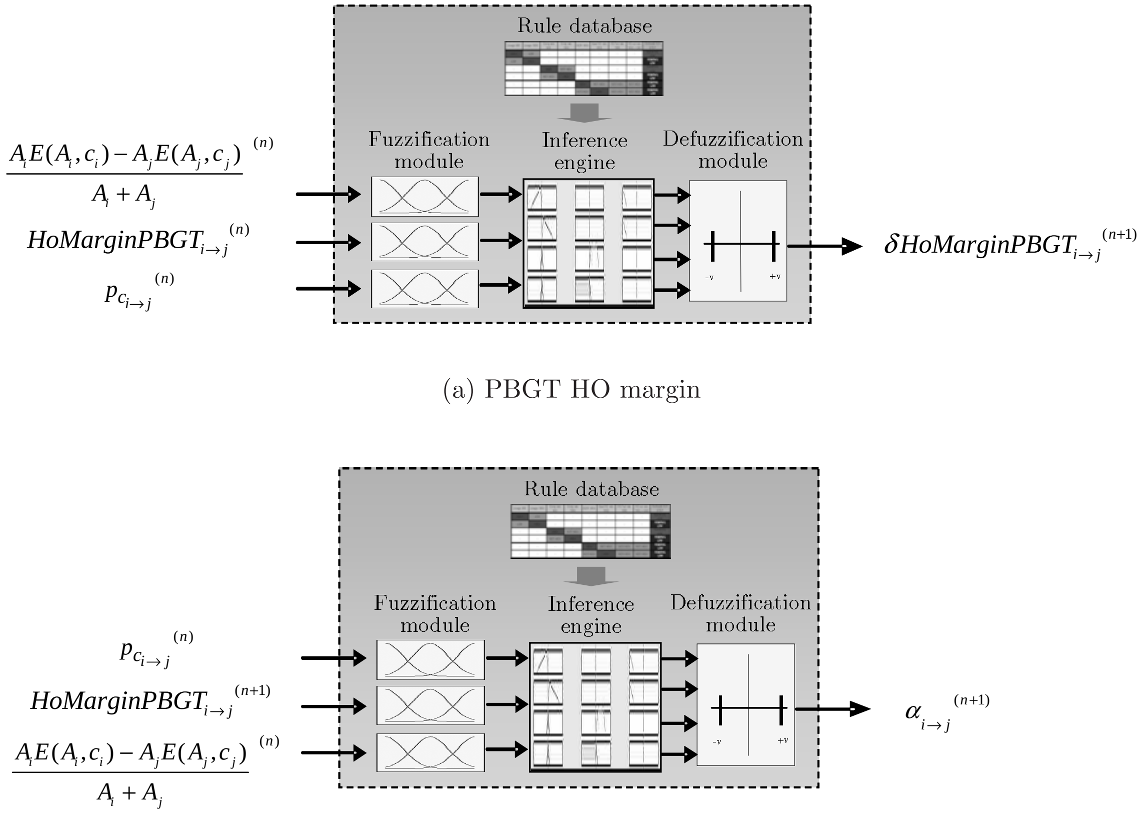
The latter equation shows that the minimum PBGT HO margin value is defined on a per-adjacency basis from $p_{c_{i \to j}}$. Hereafter, $(C/I)_{min}$ is set to 9dB.

3) In the previous approach, equilibrium is reached when the blocked traffic is the same in both serving and neighbour cell. A further refinement considers the adaptation beyond the balance of blocked traffic. This adaptation process intends to favour adjacencies with positive margins at the expense of the ones with negative margins, once the equilibrium of blocked traffic is reached. Thus, cells continue to re-shape searching for a better balance among neighbour cells. To achieve this goal, a slow-return mechanism towards the initial settings (i.e., positive HO margin values) is implemented.

4) For simplicity, previous approaches have not fully exploited the fact that HO signal-level constraints can be defined on an adjacency basis. So far, these constraints have been, at most, adjusted based on interference in the target cell. Thus, in OSLC, all adjacencies with the same target cell have the same value for the HO signal-level constraint. Alternatively, the target outage probability in OLSC, α, can be adjusted on a per-adjacency basis based on the current state of HO margins, blocking difference and interfering collision probability in the adjacency. Thus, $\alpha_{i \to j}$ (and, hence, $RxLevMinCell_{i \to j}$) should be more restrictive when $HoMarginPBGT_{i \to j} < 0$, the target cell j is interfered by the source cell i, and the source cell i is not heavily congested. By adapting $\alpha_{i \to j}$ instead of $RxLevMinCell_{i \to j}$, the interference in the target cell can still be taken into account by OLSC.

For efficiency, the proposed method is designed as a fuzzy controller. *Fuzzy inference systems* (FISs) [119] are especially suitable for decision making under approximate information. This attribute stems from its ability to deal with imprecise, flexible or uncertain information. While other approaches strive to define parameters accurately, fuzzy sets provide a more adequate representation of experts' understanding in linguistic (i.e., imprecise) terms. Likewise, fuzzy sets can be used to represent the degree of satisfaction of soft (i.e., flexible) constraints. This flexibility can be used to improve the overall optimisation objective when a trade-off among the constraints and objectives is allowed. In addition, fuzzy sets ease the handling of data expressed by intervals to deal with uncertainties that can be rigorously bounded.

The structure of the FIS for tuning HO parameters is shown in Figure 3.20. The FIS consists of two modules: one devoted to the optimisation of PBGT HO margins and the other to HO signal-level constraints. The module in Figure 3.20 (a) computes the margin increment in an adjacency, $\delta HoMarginPBGT_{i \to j}^{(n+1)}$, from the difference of blocked traffic, the current margin value and the interfering collision probability in the adjacency. The module in Figure 3.20 (b) computes the target outage probability, $\alpha_{i \to j}^{(n+1)}$, from the same indicators. This parameter is then used to derive the HO signal-level constraint by applying OSLC to *RxLev-RxQual* statistics in the target cell of the adjacency. Although it might seem that both modules share the same inputs, and could thus be implemented as a single multiple-output FIS, it is worth noting that $\alpha_{i \to j}^{(n+1)}$ depends on the new margin value, $HoMarginPBGT_{i \to j}^{(n+1)}$, whereas the margin increment depends on the old value, $HoMarginPBGT_{i \to j}^{(n)}$. Thus, the output of the former module is used to compute an input to the latter module. It is also worth noting that, in the margin case, only


Rule database
Fuzzification module
Inference engine
Defuzzification module
$\frac{A_i E(A_i,c_i) - A_j E(A_j,c_j)}{A_i + A_j}^{(n)}$
$HoMarginPBGT_{i\to j}^{(n)}$
$p_{c_{i\to j}}^{(n)}$
-v
+v
$\delta HoMarginPBGT_{i\to j}^{(n+1)}$
Rule database
Fuzzification module
Inference engine
Defuzzification module
$p_{c_{i\to j}}^{(n)}$
$HoMarginPBGT_{i\to j}^{(n+1)}$
$\frac{A_i E(A_i,c_i) - A_j E(A_j,c_j)}{A_i + A_j}^{(n)}$
-v
+v
$\alpha_{i\to j}^{(n+1)}$

(a) PBGT HO margin

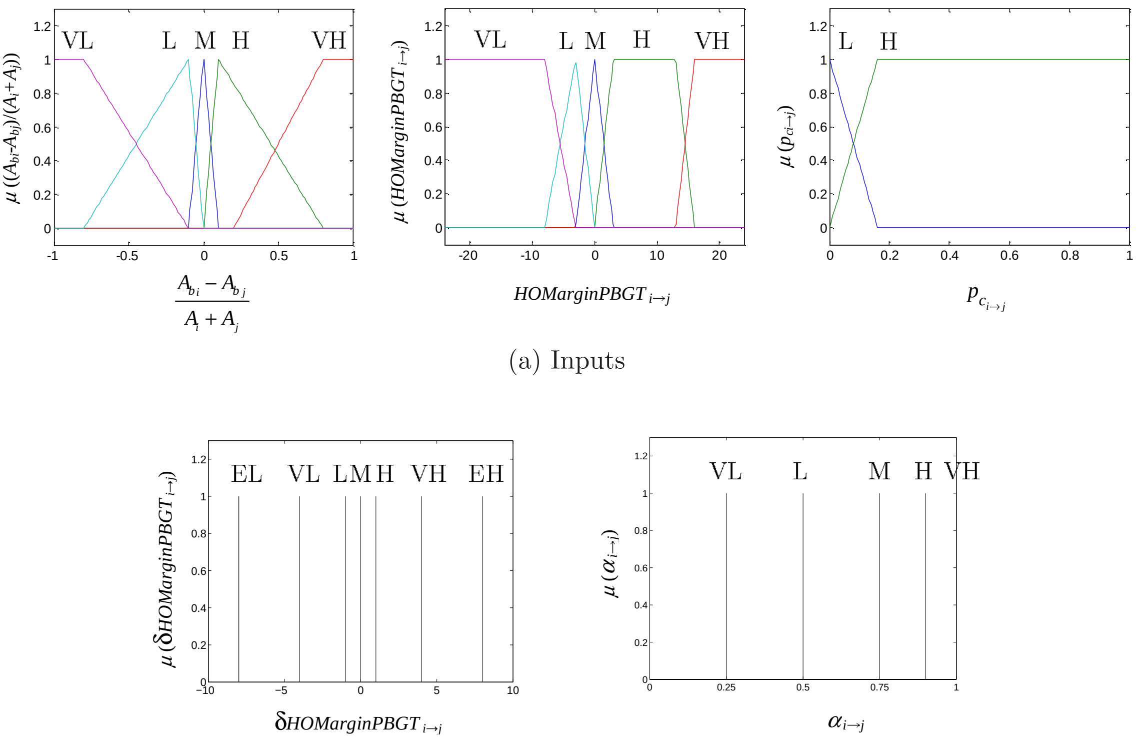
(b) Target outage probability

Figure 3.20: The FIS structure for the regulated parameters.

one direction of the adjacency has to be evaluated. Specifically, the algorithm calculates the margins in the direction of the adjacency where the source cell has a higher blocking. Then, the margins in the opposite direction are set to maintain the hysteresis region.

For simplicity, both FIS modules are implemented based on the *Takagi-Sugeno* approach [120]. The FIS consists of three stages: fuzzification, inference and defuzzification. In the *fuzzification* stage, each (crisp) value of the input variables is mapped into a set of fuzzy (or linguistic) variables. This mapping is made by a membership function, $\mu_{mn}(x_m)$, which defines the degree with which each value of the input variable m, x_m, is associated to the fuzzy variable n. Figure 3.21 depicts the membership functions used to describe inputs and outputs in linguistic terms. As observed in the figure, each input and output is classified in terms of linguistic variables, which range from extremely low (EL) to extremely high (EH). For simplicity, the selected input membership functions are trapezoidal, triangular or constant. The overlap between input membership functions is the key to crisp input values being associated to several linguistic variables simultaneously. In contrast, the output membership functions are constant functions. This type of FIS is referred to as *zero-order Sugeno FIS*.

In the *inference* stage, a set of *if-then* rules defines the mapping of the input to the output in linguistic terms. A single fuzzy rule has the form 'if x is A, then y is B', where A and B are linguistic variables to classify inputs and outputs, respectively. The first part of the rule (i.e., x is A) is called the *antecedent*, while the second part (i.e., y is B) is called the *consequent*. In contrast to classical expert systems, where only one rule is fired at a time, several rules can be fired simultaneously in a fuzzy inference engine. The firing of each rule depends on the degree



(a) Inputs

(b) Outputs

EL: Extremely Low, VL: Very Low, L: Low, M: Medium, H: High, VH: Very High, EH: Extremely High

Figure 3.21: The FIS membership functions.

in which its antecedents are satisfied (referred to as *truth value* of the rule). Several rules can thus be fired with different strengths. In the case of multiple antecedents, a single truth value must be computed for the whole antecedent. In this work, the *algebraic product operator* is used to join several antecedents. Thus, the rules take the form ’if x is A and y is B, then z is C’. The strength of rule l, $s_l(\mathbf{X})$, is then calculated by the *algebraic product* [121] operation

$$s_l(\mathbf{X}) = \prod \mu_{mn}(x_m) \quad \forall\ (m,n) \text{ in the antecedent of rule } l, \tag{3.47}$$

where $\mathbf{X}$ is the input vector to the FIS and $\mu_{mn}(x_m)$ is the truth value of any part of the antecedent. Table 3.2 summarises the set of rules that describe the tuning process. For instance, rule 1 reads as: ”if blocking difference is very large and the current margin is very large, then the HO margin step is extremely large”. Briefly, the margin increment, $\delta HoMarginPBGT_{i\to j}$, and the target outage probability, $\alpha_{i\to j}$, are larger for adjacencies that display large blocking difference between source and target cell, cells do not share frequencies and current margin values are positive. A closer look on the table reveals some of the choices made in the design of the FIS. On the one hand, reducing the number of membership functions in the input variables helps to reduce the number of rules. In contrast, increasing the number of membership functions in the output variables allows finer gain-scheduling and slow-return mechanisms. Likewise, it is observed that the blocked traffic difference is only classified as M, H or VH (and not as L or VL). As the FIS only computes margins in the direction of the adjacency where the source cell has a higher blocking, the blocked traffic difference can only take non-negative values, and therefore be classified as VH, H or M, as shown in Figure 3.21 (a).

Rule no.	$\frac{A_iE_i-A_jE_j}{A_i+A_j}$	$HoMarginPBGT_{i\rightarrow j}$	$p_{c_{i\rightarrow j}}$	$\delta HoMarginPBGT_{i\rightarrow j}$
1	VH	VH	-	EL
2	VH	H	L	EL
3	VH	H	H	VL
4	VH	M|L|VL	L	VL
5	VH	M|L|VL	H	VL
6	H	$\overline{\text{M}\mid\text{L}\mid\text{VL}}$	L	VL
7	H	$\overline{\text{M}\mid\text{L}\mid\text{VL}}$	H	L
8	H	M|L|VL	L	L
9	H	M|L|VL	H	L
10	M	VH	-	L
11	M	H	-	M
12	M	M|L|VL	-	H
Rule no.	$\frac{A_iE_i-A_jE_j}{A_i+A_j}$	$HoMarginPBGT_{i\rightarrow j}$	$p_{c_{i\rightarrow j}}$	$\alpha_{i\rightarrow j}$
13	-	M|H|VH	-	VH
14	VH	L	L	H
15	$\overline{\text{VH}}$	L	L	M
16	$\overline{\text{M}}$	L	H	M
17	M	L	H	L
18	-	VL	-	VL

| : Logical OR, $\overline{(\bullet)}$: Logical NOT

Table 3.2: Set of fuzzy control rules.

In the *defuzzification* stage, the output value is obtained by aggregating all rules. The *centre-of-gravity* method [121] is applied here to compute the final value of the output, O_{fis}, as

$$O_{fis} = \frac{\sum_{l=1}^{N_r} s_l(\mathbf{X}) \cdot o_l}{\sum_{l=1}^{N_r} s_l(\mathbf{X})}, \tag{3.48}$$

where o_l is the crisp output value of rule l, $s_l(\mathbf{X})$ is the strength of rule l with input $\mathbf{X}$ and N_r is the number of rules.

The above-described method is a fuzzy extension of the previous methods, and hence the name of *Fuzzy Slow HO Margin Control-Fuzzy Optimisation of Signal-Level Constraints* (FSHMC+FOSLC).

Practical Issues

The previous methods solve congestion in a cell by modifying the service area of neighbour cells. As a result, many users are sent to a cell other than the best serving cell in the early stages of the call connection. This action leads to an increase of the number of HOs in the network, especially in a low-mobility environment. To keep this increase as small as possible, it is important to ensure that the cell where the user initiates the call is the cell to which it would be re-directed by HO. Thus, the number of HOs can be minimised by synchronising the cell service area for MSs in idle and connected mode. This can be achieved by favouring the camping of users in those cells that are the main target of HOs.

To achieve such an effect, the parameters of the CRS algorithm can be adjusted. As explained in the previous chapter, an idle MS selects the cell where a call will be initiated based on C1 and C2 criteria [116]. The C1 value is calculated on a per-cell basis as

$$\mathrm{C1} = \mathrm{RLA\_C} - RxLevAccessMin \ , \tag{3.49}$$

where RLA_C is the average received signal level from a cell and *RxLevAccessMin* is a parameter that defines the minimum signal level to have access to the network through that particular cell. Alternatively, the C2 value is calculated as

$$\mathrm{C2} = \begin{cases} \mathrm{C1} + \mathit{CellReselectOffset} - \mathit{TemporaryOffset} \cdot \mathrm{H}(PenaltyTime - t) & \\ & \text{if } PenaltyTime < 640\mathrm{s}, \\ \mathrm{C1} - \mathit{CellReselectOffset} & \text{if } PenaltyTime = 640\mathrm{s}, \end{cases} \tag{3.50}$$

where *CellReselectOffset* is a permanent offset, *TemporaryOffset* is a temporary offset that is subtracted during the period defined by *PenaltyTime*, and H(x)=1 if x>0, 0 otherwise. The *CellReselectOffset* parameter in (3.50) can be used to control the dominance area of cells during CRS to synchronise it to the one defined by HO. Cells with positive HO margins in their outgoing adjacencies should have a positive bias, as their service area defined by HO is larger. On the contrary, cells with negative HO margins in their outgoing adjacencies should have a negative bias, as their service area defined by HO is smaller. In this work, only negative bias is implemented, as positive bias might cause that calls are initiated in cells that do not provide adequate signal level. For this purpose, the third line of (3.50) must be adopted, since *CellReselectOffset* ≥ 0. With this expression, the larger the value of *CellReselectOffset*, the smaller C2, and hence the smaller the dominance area during CRS.

Unfortunately, CRS offsets are defined on a per-cell basis, whereas HO margins are defined on a per-adjacency basis. Hence, a single offset value must be computed by averaging HO margins in all adjacencies of a cell. A weighted average is used here to give priority to those adjacencies that attract most HOs in a cell. Thus, the new value of *CellReselectOffset* in cell i is calculated as

$$\mathit{CellReselectOffset}_i^{(n+1)} = \max\left(0, -\frac{\sum\limits_{j\in V(i)} \Delta HoMarginPBGT_{i\to j}^{(n+1)} \left(N_{ho}{}_{i\to j}^{(n)} + N_{ho}{}_{j\to i}^{(n)}\right)}{\sum\limits_{j\in V(i)} \left(N_{ho}{}_{i\to j}^{(n)} + N_{ho}{}_{j\to i}^{(n)}\right)}\right), \tag{3.51}$$

where $V(i)$ is the set of neighbours of cell i, $\Delta \mathit{HoMarginPBGT}_{i\to j}^{(n+1)}$ is the new margin displacement from the default settings, $\mathit{HoMarginPBGT}_{i\to j}^{(0)}$, and $N_{ho}{}_{i\to j}^{(n)}$ and $N_{ho}{}_{j\to i}^{(n)}$ are the number of HOs in each direction of the adjacency with the old settings. Basically, a cell with $\Delta \mathit{HoMarginPBGT}_{i\to j}^{(n+1)}<0$ (i.e., reduced HO dominance area) leads to $\mathit{CellReselectOffset}_i^{(n+1)}>0$

(i.e., negative bias). Conversely, a cell with $HoMarginPBGT_{i\rightarrow j}^{(n+1)}>0$ (i.e., enlarged HO dominance area) would lead to $\mathit{CellReselectOffset}_{i}^{(n+1)}<0$ (i.e., positive bias), which is substituted by 0. This method is referred to as *Adaptation of Cell Re-selection Offsets* (ACRO).

3.3.5 Convergence Analysis

One of the main weaknesses of fuzzy controllers designed from experts' knowledge is the lack of a proof of convergence. In the proposed diffusive method, equilibrium should be reached after a few steps if the magnitude of steps is kept within reasonable limits. However, although convergence is rather intuitive to prove, the convergence speed is more difficult to estimate.

A priori, the convergence rate of diffusive methods has an upper bound in the gradient descent method [70]. It is expected that the direction of changes in the diffusive method is close to the negative of the gradient. This can be intuitively shown from the fact that the bulk of the changes takes place in the adjacencies with a severe congestion imbalance, and, in these adjacencies, small changes in traffic demand greatly reduce congestion due to the increasing slope of the Erlang-B function with traffic. Hence, it is envisaged that the convergence rate of the diffusive method is not far from that of gradient descent (provided that a proper step-length is set), i.e., the deviation from the optimal performance should converge to zero as a geometric series with iterations. Thus, large imbalances should be corrected after a few iterations, leading to a fast decrease of the total blocked traffic. However, it is also expected that the method shares the main drawback of gradient methods, i.e., slow convergence rate as the solution approaches to the optimum. Consequently, the impact of the method on network performance might be negligible after a few iterations.

A deeper analysis reveals that the convergence rate depends on network topology and initial traffic distribution. Intuitively, the more cells in the network, the slower the convergence of the balancing algorithm. For multi-processor systems, it was shown in [122] that pure load-balancing diffusive algorithms converge asymptotically to a balanced state in $O(D \cdot K^2)$ time, where D is the number of network dimensions and K is the number of processors in the largest dimension. This result, albeit logical, provides a pessimistic bound for cellular networks. Thus, it is expected that the presence of constraints in the tuning process speeds up convergence. Unlike distributed computing systems, traffic demand in mobile networks cannot traverse the whole network during the balancing process. Consequently, the effect of re-allocating traffic is restricted to those cells where most of the traffic is concentrated. Likewise, bounds set by the operator on the HO margins produce the same result. Finally, the integrality constraint on the HO margins avoids unnecessary iterations to reach the optimal solution, which would have a negligible impact on network performance. Obviously, all these constraints improve convergence at the expense of degrading the quality of the final solution.

3.4 Field Trial

This section presents the results of the field trial described in [123]. The purpose of this initial test was to show the potential of tuning PBGT HO margins in a live environment. By comparing the performance of the existing network configuration with the one obtained by a simple tuning algorithm, the gain of the optimisation process could be roughly estimated. For clarity, the trial methodology is described first and the trial results are discussed later.

3.4.1 Trial Set-up

The aim of the trial was to solve localised congestion problems caused by operator tariff policy. The tariff policy of a cellular network operator has a strong impact on the tele-traffic load in the network. Offering free evening calls will inevitably lead to an increase in the level of evening traffic, since free mobile phone calls replace chargeable fixed-line calls. Due to free evening talk time, traffic tends to be generated in residential areas, where peak day-time traffic is comparatively low. As a consequence, network capacity would have to be added in these areas to cater for calls that provide no extra revenue. To maximise revenues, operators aim to handle this additional off-peak traffic demand with the existing network infrastructure. This goal can be achieved if spare capacity, which may be available in surrounding cells, is used to carry traffic from the congested cells. Thus, additional resources do not have to be deployed to cope with new capacity demands. The following description gives a brief outline of the scenario, the experiments carried out and the criteria adopted to evaluate the method during the trial.

Trial Scenario

The trial area consisted of one BSC providing seamless coverage. The geographical area under the BSC was a dense-urban area, covering both business and residential areas. The trial BSC comprised 91 cells, distributed in 45 sites, with 1800 adjacencies.

Assessment Methodology

A computer programme was created to tune PBGT HO margins. The tuning algorithm aimed to equalise the difference in call blocking rate (BR) between neighbour cells. This decision aimed to achieve a consistent blocking probability throughout the network. Thus, the change of margins between a pair of cells was proportional to the difference of call blocking rate between them. This action resulted in permanent modification of cell service area. To maintain the hysteresis, these changes were carried out on a per-adjacency basis. Thus, the PBGT HO margin from $i \rightarrow j$ was the same as from $j \rightarrow i$, but might be different from $i \rightarrow m$. In the algorithm, the hysteresis (i.e., $2 \cdot HoMarginPBGT_{i\rightarrow j}^{(0)}$) was 12 dB, the diffusion parameter, β, was 1000, and the maximum HO margin step permitted, δ, was 10 dB. To reduce the number of changes in the network, the balancing rule was only applied to those adjacencies of a cell that carry more than 4% of the total number of incoming and outgoing HOs. For the same reason, all changes below 2 dB were not implemented in the network.

The trial extended over a period of two consecutive weeks. The method was enabled during the first week (i.e., *active* period), whereas it was disabled in the following week (i.e., *inactive* period). By comparing network performance in both periods, the benefit of the method could be quantified.

During the week when the method was enabled, the method was applied twice daily to adapt to the traffic distribution during *peak* (i.e., 8-18 h) and *off-peak* (i.e., 19-23 h) periods. In this approach, it is assumed that the spatial traffic distribution does not change from one day to the next (provided that HO margins are not modified), while it does within a day (as a consequence of user displacement from business to residential areas). These daily fluctuations can be dealt with by using a differentiated HO parameter set for both periods. These two parameter sets were optimised independently by launching two tuning processes in parallel (i.e., one for each period of the day), which modified parameters on a daily basis. Thus, PBGT HO margins on each period were amended based on measurements of the same period on the previous day, i.e., the new peak (off-peak) margin values were calculated from the peak (off-peak) margin values and the peak (off-peak) BR measurements on the previous day. The analysis of data in the algorithm on a daily basis proves suitable, since the call volume in a cell remains virtually unchanged during working days within peak and off-peak period. Thus, the current state of the network can be inferred from measurements of the previous day. Obviously, the previous assumption does not hold for Monday and Saturday, as the spatial traffic distribution during working days might differ from that of the weekend. As a consequence, the tuning algorithm could only be applied from Tuesday to Friday based on measurements from Monday to Thursday.

Assessment Criteria

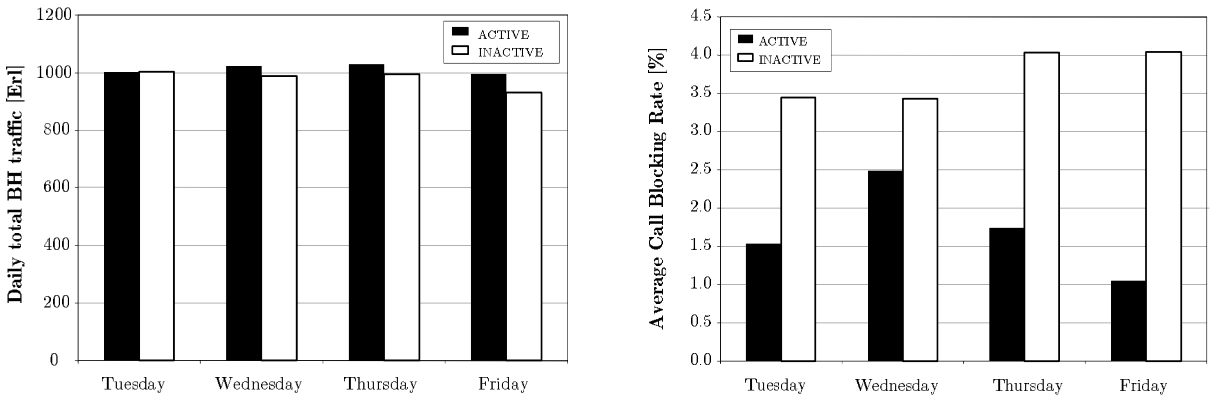
The main assessment criteria were the total carried traffic and the average call blocking rate in the area during the Busy Hour (BH). As secondary criteria, the drop call rate and the total number of HOs were also analysed. It is worth noting that the BH is defined on a per-cell basis (i.e., the BH for cells in business areas might be in the peak period, whereas the BH for cells in residential areas might be in the off-peak period). The previous performance indicators were gathered on a daily basis and the analysis was focused on the days of the week when HO parameters changed in the active period (i.e., Tuesday to Friday).

3.4.2 Trial Results

The analysis is first focused on the total carried traffic in the trial area. Figure 3.22 (a) depicts the sum of BH traffic carried by the trial cells for four consecutive days with and without the method. The total daily BH traffic averaged over the corresponding four days was 1012.1E when the method was active, while the respective traffic was 979.9E during the inactive period. This means that the carried traffic was, on average, 3.3% higher when the method was enabled. Obviously, this small difference in the total carried traffic in the network was expected, as the network had been dimensioned properly. In the figure, it is also observed that the difference between active and inactive states increased as the tuning progressed, i.e., the performance difference was negligible on Tuesday, whereas it was more pronounced on Friday.

Figure 3.22 (b) shows the daily call blocking rate averaged over the trial area for both periods. It can be noticed that the average of this rate was 1.7% when the method was active, whilst this rate raised to 3.7% (i.e., more than doubled) when the method was inactive. An increase during the inactive period was noticed, despite the fact that the carried traffic reduced by 3%. These results show conclusively that the method enables the network to carry more traffic and reduce call blocking, thus increasing the effective network capacity.

The raw number of HO attempts in the area increased by 20% when the method was active. This is mainly due to the additional PBGT HOs caused by setting negative HO margins. However, the share of HO triggering causes remained the same. This is a clear indication that quality HOs also increased with the method, which suggests that capacity improvement was



(a) Total BH traffic

(b) Average Call Blocking Ratio

Figure 3.22: Performance results from the tuning of PBGT HO margins on a daily basis.

achieved at the expense of a slight call quality impairment. This was expected due to the fact that some calls were not carried by the best-serving cell, but an adjacent cell offering spare capacity instead. Nonetheless, the drop call rate also benefited from the application of the method. When the method was in use, this rate averaged 1.6%, while it increased to 2.3% (i.e., 31% increase) when it was inactive. This effect is possibly due to a higher HO success rate as a result of congestion relief in the area.

From the trial results, it can be concluded that tuning PBGT HO margins is a powerful technique to reduce call blocking in a cellular network. However, as the tuning algorithm was extremely simple, it is not possible to estimate the maximum benefit that can be achieved by optimising these parameters. In addition, the small network area covered by the trial might not reproduce situations of severe congestion, where the method is taken to its limits. For both reasons, the analysis is extended by the simulation experiments described in the next section.

3.5 Simulation study

Once trial results have shown the potential of tuning PBGT HO margins, this section aims to prove how the limits of this technique can be extended by also tuning HO signal-level constraints. For this purpose, a set of simulations are performed over a system-level simulator. For clarity, the preliminary conditions of the simulations are described first and the results are discussed later.

3.5.1 Simulation Set-up

The following paragraphs describe the experiments carried out to assess the value of the different methods based on simulations. This description covers the simulation scenario and the assessment methodology.

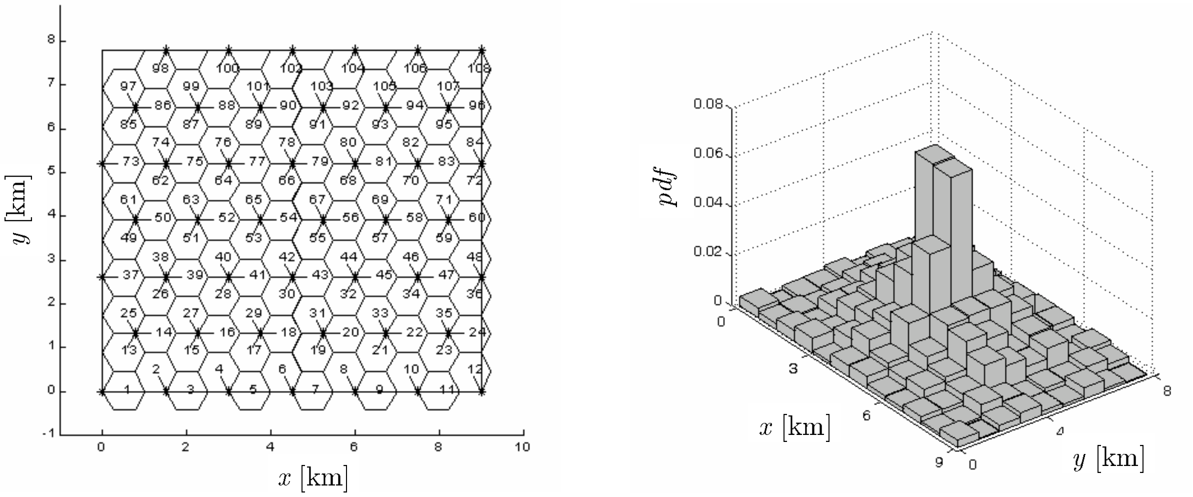
Scenario	TU3, MACRO	$r_c = 0.5$ km
Propagation model	Okumura-Hata with wrap-around	$L_b = 126.4 + 35.22 \log d$
	Minimum coupling losses	$L_{bmin} = 70$ dB
	Correlated log-normal slow fading	$\sigma_{sf} = 8$ dB, $d_{csf} = 50$ m
Mobility model	Random direction, constant speed	$v_{ms} = 3$ km/h
Service model	CS-Voice	MCD $= 1/\mu = 80$ s, $a_f = 0.5$
Spatial traffic distribution	Correlated log-normal	$\sigma_{traf} = 4$ dB
Adjacency plan	Symmetrical adjacencies	32 per cell
BTS configuration	Antenna	Tri-sectorised, HPBW = 65°
	max. EIRP	43 dBm
Features in use	Random FH, POC, DTX, DR	
HO parameter settings	Qual HO Threshold	RXQUAL = 4
	Qual HO margin	0 dB
	PBGT HO margin	[-24, 24] dB
	PBGT HO period	6 s
	RxLevMinCell	[0, 63] (≡ [-110, -47] dBm)
	DR threshold	15 (≡ -95 dBm)
Overall traffic load	36%	
Time resolution	SACCH frame (480 ms)	
Simulated network time	28 h (per optimisation epoch)	

Table 3.3: Simulation parameter settings.

Simulation Scenario

Simulations are performed on a dynamic system-level GERAN simulator, partly developed in this thesis. Table 3.3 summarises the main models and parameters in the simulation tool (for more details about the meaning of these parameters, the reader is referred to Appendix E in [10]). The simulation scenario intends to model a macro-cellular urban environment with severe congestion problems to push the proposed methods to their limits. The layout, depicted in Figure 3.23 (a), consists of 108 hexagonal cells in 36 tri-sectorized sites uniformly distributed. To reproduce a realistic case, voice traffic demand is unevenly distributed in the scenario. For this purpose, a log-normal spatial traffic distribution is implemented following the approach suggested in [124]. Figure 3.23 (b) represents the geographical distribution of traffic demand by showing the probability that a user initiates a call in each part of the scenario. From the figure, it is clear that cells in the middle of the scenario receive most of the traffic demand. This concentration of traffic demand calls for the application of traffic sharing techniques. It is worth noting that this situation can be considered as a worst-case scenario, since most of the traffic demand is originated in a small geographical area. As a consequence, limited blocking reduction can be achieved in congested cells by sharing the load with adjacent cells, as the former are also congested. The overall traffic load in the scenario is controlled by the total call arrival rate. During the simulations, the latter parameter is set so that the overall traffic load is 36%, which would result in a low blocking probability if traffic and resources were evenly distributed in the scenario (e.g., 0.007 for 8 TSLs/cell). Hence, it is the imperfect spatial match between traffic demand and deployed resources (and not the overall lack of resources) what causes the congestion problem.

For computational efficiency, a single TRX is simulated per cell. This TRX can either model a Traffic-CHannel (TCH) TRX or the BroadCast-CHannel (BCCH) TRX, depending on features in use. This decision is consistent with the way operators tend to assign frequencies



(a) Layout

(b) 2-D spatial traffic distribution

Figure 3.23: Simulation scenario.

to TRXs in a live network. To ease frequency planning, TRXs in the network are normally structured in layers that do not share frequencies. As the number of TRXs per cell varies across the network, not all layers are present in every cell, which allows for the use of a different reuse scheme for each layer (and hence the name of *Multiple Reuse Pattern*) [125]. In this strategy, the interference conditions in any TRX only depend on TRXs of the same layer in other cells. This property ensures that the interference conditions can be modelled by simulating each TRX layer separately. During the experiments, only the DL is simulated, as it proves to be the most restrictive link in GERAN.

Assessment Methodology

Six congestion-relief methods are simulated. The first two methods are classical RRM features: *Directed retry* (DR) and *Traffic-Reason HO* (TRHO). The other four methods are combinations of the self-tuning methods described in Section 3.3: the *Slow HO Margin Control* (SHMC), the previous method with adaptation of HO signal-level constraints based on the interference in the target cell (SHMC+OSLC), the fuzzy variant that jointly optimises HO margins and signal-level constraints (FSHMC+FOSLC) and the previous method with adaptation of CRS offsets (FSHMC+FOSLC+ACRO).

DR is the benchmark against which all other methods are compared, as it is the default method currently in use to cope with localised congestion problems. Thus, other methods are normally added on top of DR, since the DR feature is hardly ever disabled by operators. As already explained, DR assigns blocked calls to cells other than the best serving cell. For an adjacent cell to be considered a potential target, the signal level received from it must be above a certain threshold, defined by the *DrThreshold* parameter. In live networks, although this parameter is definable on per-adjacency basis, most operators fix it to a safe value, which is deployed network wide. In this work, the homogenous settings are slightly modified to extend the limits of DR. Thus, the *DrThreshold* parameter is set to its maximum value (i.e., 63 ≡ -47dBm) for adjacent cells that are potential co-channel interferers of the serving cell. This technique reduces the number of candidate target cells, and, consequently, more calls are finally blocked in the network. This effect is more evident with tight frequency reuses, as most adjacent cells share the same frequencies with the serving cell (and are thus potential interferers). However, this drawback is more than compensated for by the reduction of network quality impairment, which can be used to extend the limits of DR and all other algorithms that run in parallel. As a result, the *DrThreshold* parameter in the remaining adjacencies can be adjusted in a wider range without compromising network quality. Likewise, more freedom is given to other congestion-relief methods, as more room for quality impairment is available. For a similar reason, conservative settings are used for DR (i.e., *DrThreshold*=15 ≡ -95 dBm) when combined with the techniques described below.

In TRHO, PBGT HO margins are modified temporarily to relief overload situations. Whenever the load in a cell is above a certain threshold, defined by the *UpperLoadThreshold* parameter, the margins of outgoing adjacencies are reduced. For simplicity, current vendor equipment only implements two different margin values: a default value *HoMarginPBGT* and a temporary value used for congestion relief *TrHoMarginPBGT*. Thus, only abrupt changes between these two values are possible. To prevent users from returning back to the original cell, users are not allowed to make a HO for a period of time defined by the *GuardTime* parameter. To avoid additional instabilities, the load of a neighbour cell must be below another threshold, defined by the *MaxLoadOfTgtCell* parameter, to be considered as a candidate for a HO. The default parameter settings used in the simulations are *UpperLoadThreshold*=0.87, *MaxLoadOfTgtCell*=0.75, *HoMarginPBGT*=8 dB, *TrHoMarginPBGT*=0 dB and *GuardTime*=20 s.

In self-tuning methods, PBGT HO margins are adjusted to modify the service area of cells permanently. For simplicity, these algorithms aim to balance BR (and not the blocked traffic) between neighbour cells in the default configuration. The simulation of these methods consists of a series of steps (hereafter referred to as *epochs*) arising from the feedback loop between network and controller. In this work, each epoch comprises 100000 simulation steps, equivalent to 28 hours of actual network time. After each epoch, new parameter settings are configured in the network based on performance measurements of the previous period. Within each epoch, network parameters remain unchanged. The speed of the tuning process is controlled by the diffusion parameter, β. Thus, a higher β leads to larger margin steps, which speeds up the tuning process. For clarity, the value of β will be specified hereafter by the maximum step that can be achieved between two consecutive epochs, δ. Unless stated otherwise, the value of β is set such that $\delta = 2$ dB.

As will be shown later, the impact of methods on network quality depends largely on the frequency reuse scheme in use. For efficiency, the analysis is restricted to the most common schemes for tri-sectorised cells: BCCH 4/12 NH, TCH 3/9 NH, TCH 1/3 RH3, TCH 3/3 RH3 and TCH 1/1 RH9 (more details about these schemes can be found in [10]). For a fair comparison, all schemes share the same number of frequencies per layer (i.e., 9), except for the BCCH 4/12 NH case (i.e., 12). The TCH 1/3 RH3 is the default scheme used in the simulations.

Assessment Criteria

Several performance indicators must be taken into account to assess the value of a given HO scheme [126]. For simplicity, the analysis is focused on two performance indicators: the overall blocking rate, $\overline{BR}$, and the overall outage rate, $\overline{OR}$. It is worth noting that, even though the former indicator is computed from BR after DR (i.e., real blocking), the balancing rule still deals with the BR before DR to reduce the number of users re-directed.

To assess the value of a particular network configuration, a single cost figure is calculated from the two previous indicators. In this work, the penalty of a network configuration, p, is calculated by a weighted sum with non-linear terms as

$$p = \omega_{BR} \left(\frac{\overline{BR}}{\overline{BR}_t} \right)^e + \omega_{OR} \left(\frac{\overline{OR}}{\overline{OR}_t} \right)^e , \tag{3.52}$$

where $\overline{BR}$ and $\overline{OR}$ are the performance figures of the configuration, $\overline{BR}_t$ and $\overline{OR}_t$ are the performance target values, e is a constant to penalise the non-fulfillment of objectives, and ω_{br} and ω_{or} are the relative weights of the capacity and quality criteria. Hereafter, $\overline{BR}_t = 0.05$, $\overline{OR}_t = 0.01$, $e = 3$, and $\omega_{br} = \omega_{or} = 1/2$. The former two values are aligned to current operator demands [10], while having identical weights means that both targets are equally important. With these settings, a network configuration with $\overline{BR}$=0.05 and $\overline{OR}$=0.01 has p=1.

To assess the value of a method, the analysis investigates the trade-off between network capacity and network quality in the solutions achieved by the method. For a fair comparison among methods, an estimation of the whole Pareto-front is carried out for each method. This analysis is performed by showing a scatter plot of $\overline{OR}$ versus $\overline{BR}$ for the different network configurations provided by the method. For self-tuning methods, the construction of this sort of graph is rather straight-forward due to the epoch structure. As tuning progresses, $\overline{BR}$ decreases at the expense of an increase of $\overline{OR}$. The trade-off in these methods can be analysed by representing the point ($\overline{OR}$, $\overline{BR}$) for each epoch. A series of points is thus obtained, which is hereafter referred to as a *trajectory*. For RRM methods, the whole Pareto-front may be obtained by adjusting their internal parameters. In particular, the values of *DrThreshold* in DR and *TrHoMarginPBGT* in TRHO are modified to investigate the $\overline{OR}$-$\overline{BR}$ trade-off in these methods. In the absence of an automatic tuning method for these parameters, a trial-and-error approach has been followed to select the parameter range to be evaluated.

In principle, the main focus of the analysis of self-tuning methods is on the asymptotic behaviour. Therefore, those algorithms that lead to a better trade-off among $\overline{BR}$ and $\overline{OR}$ in the equilibrium state are normally preferred. However, as self-tuning algorithms gradually change parameters of the real network, not only the steady state but also the transient response is important. Thus, methods with a similar equilibrium state but a different transient response (e.g., intermediate states, convergence speed) should be clearly differentiated. For the evaluation of transient responses, an infinite-horizon discounted model is considered [127]. In such a model, the *total penalty* of a trajectory, P, is calculated as

$$P = \sum_{n=0}^{\infty} \gamma^n p_n , \tag{3.53}$$

where p_n is the penalty on epoch n. From (3.53), it is deduced that penalties in the future are given less importance according to a geometric law with discount factor γ, where $0 \leq \gamma \leq 1$. This discount factor reflects that, in live environments, early rewards are normally preferred to delayed rewards, as traffic conditions may greatly vary with time and situations of persistent congestion are solved in the long-term by other approaches. To circumvent the need for simulating an infinite number of epochs, it is assumed that equilibrium is reached after h epochs. Thus, the penalty is still calculated with an infinite horizon as

$$P = \sum_{n=0}^{\infty} \gamma^n p_n \approx \sum_{n=0}^{h-1} \gamma^n p_n + \sum_{n=h}^{\infty} \gamma^n p_h = \sum_{n=0}^{h-1} \gamma^n p_n + \frac{\gamma^h}{1-\gamma} p_h \, , \tag{3.54}$$

where p_h is the penalty in the last simulated epoch. Hereafter, h=19 and γ=0.85. A horizon of 19 epochs means that only 20 epochs (i.e., 1 initial state + 19 tuning steps) are simulated. This horizon proves to be large enough to ensure that the system has reached equilibrium. Even if this is not the case, the relatively low value of γ ensures that the relevance of epochs beyond this point is negligible.

For a fair comparison between RRM and self-tuning methods, the same performance indicator must be shared. Since in the former methods internal settings do not change from epoch to epoch, the trajectory consists of a single solution and the expected performance is the same across epochs. Under this assumption, (3.54) can be simplified to

$$P = p \sum_{n=0}^{\infty} \gamma^n = \frac{p}{1-\gamma} \, , \tag{3.55}$$

where p is the constant penalty figure. From (3.55), it can be deduced that a more intuitive indicator can be obtained if a normalising factor is introduced in (3.54). Thus, the *overall penalty* of a trajectory, P', is defined as

$$P' = (1-\gamma) P = (1-\gamma) \sum_{n=0}^{\infty} \gamma^n p_n \approx (1-\gamma) \sum_{n=0}^{h-1} \gamma^n p_n + \gamma^h p_h . \tag{3.56}$$

The latter value can be interpreted as the weighted average (rather than the weighted sum) of penalties across the horizon. From (3.56), it follows that a method that shows a consistent value of p across epochs (e.g., RRM method) has P'=p.

To estimate the increase of signalling load due to an increased number of HOs, the *overall HO ratio*, $\overline{HR}$, and the *mean holding time*, MHT, are calculated as

$$\overline{HR} = \frac{N_{hoT}}{N_{cT}} \, , \tag{3.57}$$

$$MHT = \frac{T_{hT}}{N_{hT}} \, , \tag{3.58}$$

where N_{hoT} and N_{cT} are the total number of HOs and carried calls, T_{hT} is the total connection time and N_{hT} is the total number of connections in the scenario.

3.5.2 Simulation Results

The first experiment shows the limitations of classical congestion-relief techniques over a test case. Figure 3.24 shows the performance of several methods in the scenario with TCH 1/3

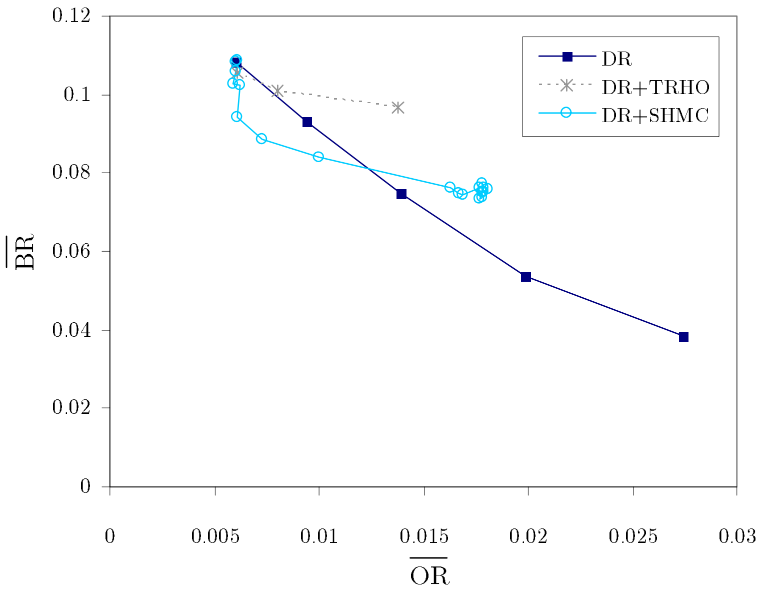
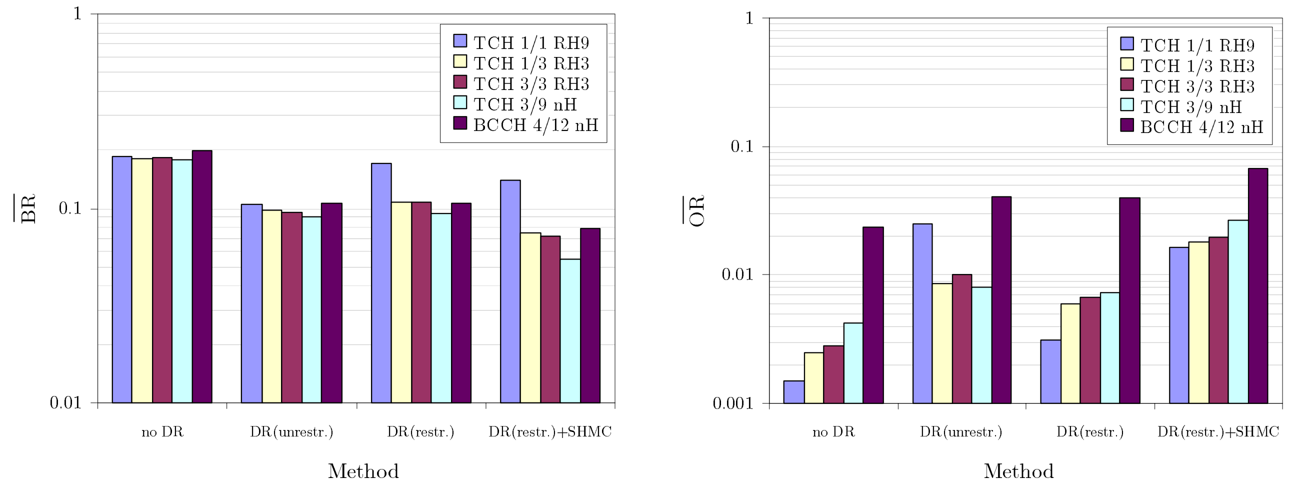


Figure 3.24: Performance of classical congestion-relief methods in the scenario.

RH3 frequency reuse. In the figure, each point represents the performance of a different network configuration, while each curve groups points achieved by the same method. For DR and DR+TRHO (i.e., RRM methods), each point represents a different configuration of internal parameter settings. In the DR curve, from left to right, each point corresponds to a value of *DrThreshold* in the set {15, 13, 11, 9, 7} ≡ {-95, -97, -99, -101, -103} dBm to non-interfered neighbour cells. In the DR+TRHO curve, each point corresponds to a value of *TrHoMarginPBGT* in the set { 0, -4, -8} dB. For DR+SHMC (i.e., self-tuning method), each point corresponds to an epoch in the tuning process. Thus, the DR+SHMC curve represents the states of the system as they are reached during the tuning process. In DR+TRHO and DR+SHMC, *DrThreshold* is set to the default value (i.e., 15 ≡ -95 dBm). Finally, it is worth noting that, while points in the DR and DR+TRHO curves (i.e., RRM methods) have no time relationship, and can be simulated separately, points in the DR+SHMC curve (i.e., self-tuning method) are part of a sequence of states reached during the tuning process, which must be simulated sequentially. In the figure, it is observed that all methods reduce $\overline{BR}$ by increasing $\overline{OR}$. Despite its simplicity, DR is an effective method to solve localised congestion problems when its parameters are adjusted. Concretely, $\overline{BR}$ in the scenario can be decreased from 10.8% to 3.9% (i.e., almost three-fold reduction) when *DrThreshold* changes from 15 to 7. However, this effect is achieved at the expense of a significant impairment of the overall connection quality, which is evident from the increase of $\overline{OR}$ from 0.6% to 2.7% (i.e., five-fold increase). By contrast, DR+TRHO is totally ineffective. Even if the addition of TRHO leads to a slight reduction of $\overline{OR}$ from 10.8% (i.e., DR only) to 10.6% (i.e., DR+TRHO with default settings), no further benefit is obtained from adjusting *TrHoMarginPBGT* in the set {0, -4, -8} dB. This result is caused by the safety mechanism that prevents users from being sent to cells with high loads. Since most congested cells in the scenario are adjacent to each other, no load balancing is triggered among these cells. Finally, it is observed that DR+SHMC performs reasonably well in its initial epochs, where the *HoMarginPBGT* parameter is still positive in all adjacencies. Thus, $\overline{BR}$ is reduced from 10.8% to 9.4% with no quality impairment. However, once HO margins become negative, severe call quality impairment is observed (i.e., $\overline{OR}$ up to 1.8%), while limited blocking relief is attained (i.e., $\overline{BR}$ reduced from 9.4% to 7.6%). From this result, it can be



(a) Blocking rate (b) Outage rate

Figure 3.25: Influence of frequency reuse scheme on the performance of classical methods.

concluded that, with the classical method to tune PBGT HO margins, there is no point in setting negative HO margin values.

The next experiment highlitghts the cause of limitations by testing the previous methods with different frequency reuse schemes. Figures 3.25 (a)-(b) show $\overline{BR}$ and $\overline{OR}$ for four methods with five reuse schemes. The four methods considered (from left to right in the figures) are no DR, DR with $DrThreshold = 15$ to all neighbour cells (DR unrestr.), DR with $DrThreshold =$ 15 and 63 to non-interfered and interfered neighbours, respectively (DR restr.), and the latter with tuning of HO margins (DR restr.+SHMC).

The analysis is first focused on DR. From the comparison of the first two sets of bars in Figure 3.25 (a), it can be deduced that $\overline{BR}$ is halved when DR is enabled (note the logarithmic scale). However, Figure 3.25 (b) shows that unrestricted DR can severely deteriorate $\overline{OR}$, regardless of the conservative *DrThreshold* value configured (i.e., 15 ≡ -95 dBm). When a call is re-directed to a cell other than its nominal cell, it might experience bad quality due to proximity to an interferer. This situation is rather common in TCH 1/1 RH9, as source and target cell in DR share the same frequencies. For the latter scheme, $\overline{OR}$ increases from 0.1% to 2.5% when DR is enabled with no restrictions. It is thus clear that restricting DR to neighbours interfered by the original cell can improve network quality. With this approach, $\overline{OR}$ in TCH 1/1 RH9 can be reduced in an order of magnitude, but at the expense of loosing most of the traffic sharing capability of DR. Such a loss is negligible for loose frequency reuses, where the number of co-channel interferers in the neighbor-cell list is small. These results prove the benefit from restricting DR to interfered neighbours. It should be pointed out that the higher $\overline{BR}$ for the BCCH-TRX under the same traffic is due to the use of 6 TSLs instead of 8, which aims to reflect the existence of signalling channels on this TRX. Likewise, the higher $\overline{OR}$ in the BCCH-TRX is due to the lack of features such as DTX, POC and FH.

The effect of SHMC can be isolated from that of DR by comparing the last two sets of bars. It is observed that SHMC can reduce $\overline{BR}$ at the expense of a non-negligible quality impairment. As in other congestion-relief methods, the impairment from SHMC is more severe in tighter frequency reuses. For instance, $\overline{OR}$ increases by two orders of magnitude when SHMC is enabled in TCH 1/1 RH9. More difficult to explain is the fact that $\overline{BR}$ depends on

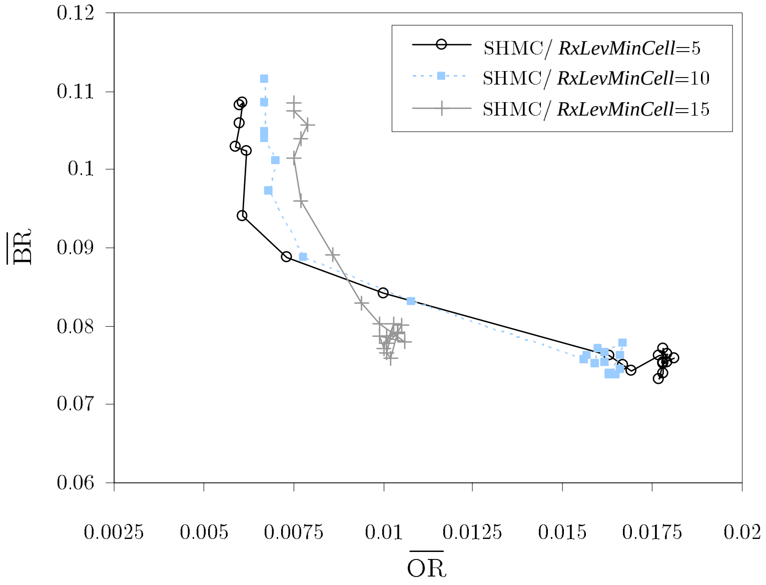


Figure 3.26: Influence of HO signal-level constraints in the performance of SHMC.

the frequency reuse scheme. Thus, $\overline{BR}$ is lower for TCH 3/9 nH, regardless of the method used. Such an unexpected behaviour is more evident for SHMC. This outcome is mainly due to the Qual HO. If, after a PBGT HO, a user experiences bad quality, a Qual HO is triggered. In the subsequent target cell evaluation, cells are ranked again based on their PBGT, but Qual HO margins are set to 0 to ensure that the MS is sent to the cell with the highest received signal-level. This fact causes that the MS is sent back to the original cell shortly after a PBGT HO occurs. Obviously, this ping-pong effect, which limits the traffic sharing capability, is less frequent in loose frequency reuses, since the reuse distance is larger. This is the reason why SHMC is more effective with loose frequency reuses (e.g., TCH 3/9 nH).

The previous results provide strong evidence that most traffic sharing techniques experience interference problems in tight frequency reuses, unless countermeasures are adopted. The rest of the analysis is restricted to the TCH 1/3 RH3 scheme, although similar results are expected with other frequency reuses. Likewise, the restricted version of DR is used.

The next experiment shows how interference problems in SHMC can be alleviated by adjusting HO signal-level constraints. For this purpose, SHMC is tested with different values of the *RxLevMinCell* parameter. Figure 3.26 shows the performance of SHMC with the latter parameter fixed to {5, 10, 15} (≡ {-105, -100, -95} dBm) for all neighbour cells in the scenario. It is observed that, in the initial stage of the tuning process (i.e., upper part of the curves), the higher (i.e., the more restrictive) *RxLevMinCell*, the higher $\overline{OR}$ (i.e., the lower overall network quality). In particular, in the first epoch, where HO margins have not changed yet, $\overline{OR}$ increases from 0.6% to 0.75% by raising *RxLevMinCell* from 5 to 15. This is mainly due to the fact that, when normal HO margin values are used, PBGT HO ensures that an MS is never sent to a worse cell. Hence, increasing HO constraints only leads to rejection of valid candidate neighbour cells, which might be used in the case of severe shadowing of the serving cell. In these conditions, unnecessary restriction of HO through tight signal-level requirements would contribute to worsen (rather than enhance) network quality. On the contrary, when HO margins become negative (i.e., lowest part of the curve), the absence of signal-level restrictions might cause severe impairment of call quality. In these cases, the enforcement of HO restrictions can

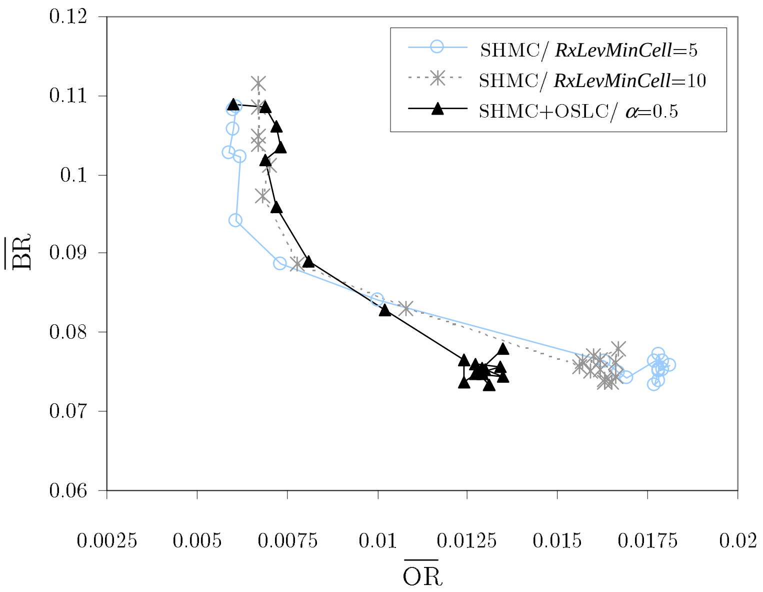


Figure 3.27: Performance of the combination of SHMC and OSLC.

prevent quality problems. For instance, $\overline{OR}$ at the end of the tuning process can be reduced from 1.8% to 1.0% (i.e., almost halved) with a negligible loss of the $\overline{BR}$ reduction capability by increasing *RxLevMinCell* from 5 to 15. From these results, it can be concluded that signal-level constraints must adapt to the current state of HO margins.

The following experiment shows the benefit of restricting HO to target cells that are more interfered when using SHMC. Figure 3.27 presents the results of combining SHMC and OSLC. In SHMC+OSLC, the *RxLevMinCell* parameter is optimised on a per-adjacency basis by applying OSLC over measurements of the target cell. The results of SHMC with homogeneous *RxLevMinCell* settings are also superimposed for comparison purposes. For a fair comparison, it is important to ensure that SHMC+OSLC ends up with a similar level of restrictions, i.e., even if *RxLevMinCell* is modified on a per-adjacency basis, the average value of *RxLevMinCell* should be maintained. This was achieved by setting the target outage probability, α, to 0.5. After the initial epoch, where *RxLevMinCell*=5 (≡ -105 dBm), OSLC modifies *RxLevMinCell* in the adjacencies, averaging 12 (≡ -98 dBm) in the scenario. This value can be inferred from the figure, as the SHMC+OSLC curve is close to SHMC with *RxLevMinCell*=10 (≡ -100 dBm). After some loss of quality in the initial epochs, the quality improvement is noticeable in the last epochs, when HO margins become negative. Concretely, $\overline{OR}$ in equilibrium can be reduced from 1.8% to 1.3% when OSLC is introduced. It is worth noting that this quality improvement is achieved with no loss of $\overline{BR}$ reduction capability. From these results, it is clear that non-homogeneous *RxLevMinCell* settings often lead to a better trade-off between network quality and capacity in SHMC.

Figure 3.28 shows the influence of the target outage probability on the performance of SHMC+OSLC. From the figure, it is evident that the algorithm is quite sensitive to this parameter. The reasoning behind this process is that the higher α, the looser the HO signal-level constraints and the lower *RxLevMinCell*. The outcome of increasing α depends on the current state of HO margins. While $\overline{OR}$ decreases with increasing α for the early epochs, the opposite is true for the last epochs. This result suggests that α should be adjusted based on the current state of HO margins.

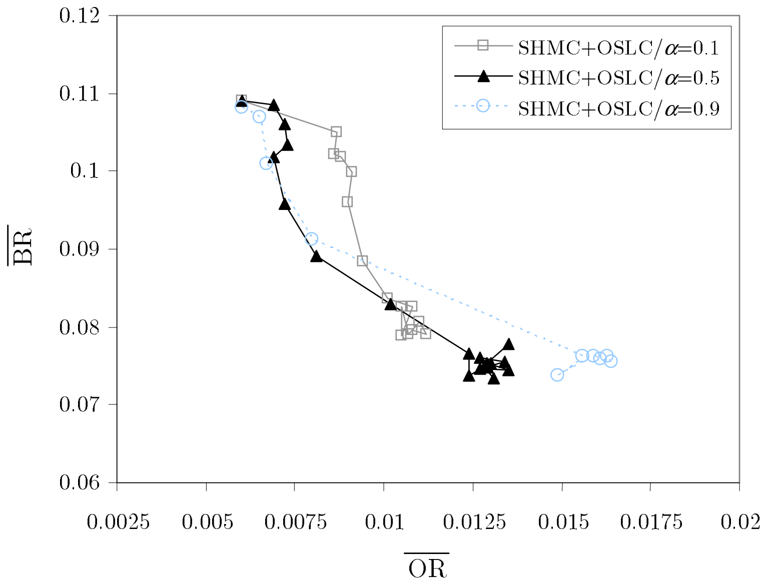


Figure 3.28: Influence of the outage probability in SHMC+OSLC.

The following experiment proves the benefit of jointly optimising HO margins and signal-level constraints by the FIS proposed. The key difference is the optimisation of *RxLevMinCell* not only based on the interference in the target cell, but also on the values of *HoMarginPBGT*. Figure 3.29 presents the results of FSHMC+FOSLC. In the example, the target outage probability in OSLC ranges from 0.4 to 1, depending on the current state of HO margins in the adjacency. For comparison purposes, the best approaches described so far are also superimposed in the figure. From the figure, it is clear that FSHMC+FOSLC performs extremely well throughout the whole trajectory. In equilibrium, FSHMC+FOSLC provides the lowest $\overline{OR}$ (i.e., 0.9%), with only 0.7% absolute more $\overline{BR}$ than the best solution achieved by other tuning methods. More important, for any epoch, there is no feasible solution that would enhance $\overline{BR}$ without impairing $\overline{OR}$ (i.e., the FSHMC+FOSLC solution is always non-dominated). Thus, it is expected that the FSHMC+FOSLC curve provides an accurate estimation of the Pareto-front for the tuning of HO margins. This result is just a consequence of the gradual increase of HO signal-level constraints as HO margins decrease. Finally, it is worth noting that FSHMC+FOSLC can be converted into any of the previous methods by a proper adjustment of the internal parameters.

From the previous figure, it is difficult to state which method is best among those that provide some non-dominated solutions. Even within a trajectory, it is difficult to say which network parameter setting provides the best network performance, since assessment is based on two opposite criteria. To solve this issue, the analysis focuses on the penalty figures. The overall penalty of a trajectory, P', is used to identify the best method, while the penalty of a point in the trajectory, p, is used to find the best network configuration. Table 3.4 presents the performance of different approaches in terms of overall penalty, P'. The minimum penalty of an epoch in the trajectory, p_{min}, and the penalty in the equilibrium state, p_{eq}, are shown in the second and third column, respectively. Obviously, all three indicators coincide for RRM methods, as the trajectory consists of a single solution. For each method, two different internal parameter settings are shown in the table: the default settings and the optimal settings found by a trial-and-error approach. For self-tuning methods without gain scheduling (i.e., all except FSHMC+FOSLC), the optimal settings are obtained by fixing the diffusion parameter, β (and the maximum margin step, δ) to the maximum value in FSHMC+FOSLC (i.e., δ=8dB).

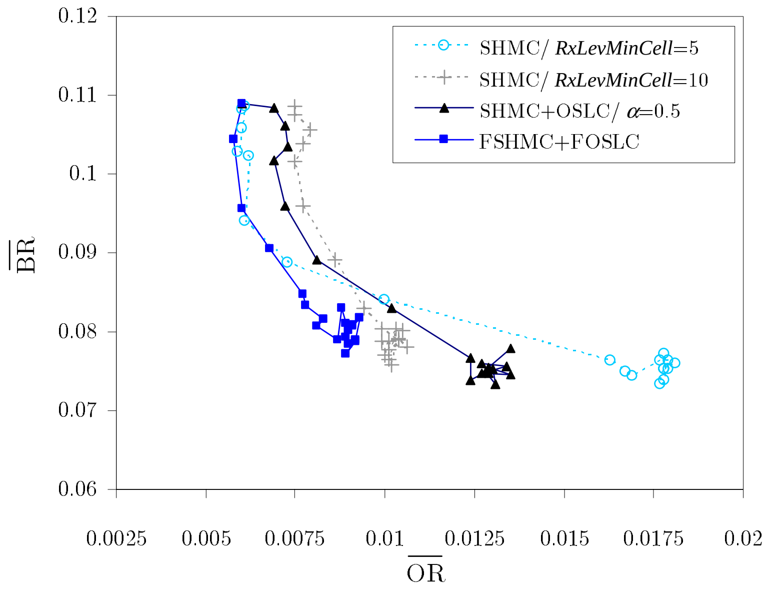


Figure 3.29: Performance of the combination of FSHMC and FOSLC.

Type	Method	P'	p_{min}	p_{eq}
RRM	DR, *DrThreshold*=15	2.52	2.52	2.52
	DR, *DrThreshold* =11	2.08	2.08	2.08
	TRHO, *TrHoMarginPBGT*=0 dB	2.42	2.42	2.42
	TRHO, *TrHoMarginPBGT*=-4 dB	2.36	2.36	2.36
Self-tuning	SHMC, *RxLevMinCell*=5, δ=2 dB	2.40	1.84	2.79
	SHMC, *RxLevMinCell*=5, δ=8 dB	2.52	2.24	2.68
	SHMC+OSLC, α=0.5, δ=2 dB	2.26	1.86	2.12
	SHMC+OSLC, α=0.5, δ=8 dB	2.12	1.88	1.99
	FSHMC+FOSLC	1.95	1.59	1.67

Table 3.4: Penalty of methods in the scenario.

The analysis is first focused on the values of overall penalty shown in the first column. At first glance, it is observed that no method achieves the performance targets in the scenario, as none of them shows a value of P' below 1 (i.e., the value with $\overline{BR} = \overline{BR}_t$ and $\overline{OR} = \overline{OR}_t$). This result shows conclusively that the test case is a limit case indeed. Among RRM methods, it is clear that DR is ineffective when the *DrThreshold* parameter is left on its default value (i.e., 15 ≡ -95dBm). In contrast, the value of P' can be decreased by 20% when the previous parameter is set to the optimum (i.e., 11 ≡ -99 dBm). This need for a low *DrThreshold* is just a consequence of the large blocking in the scenario. In contrast, TRHO is clearly inferior, even if the *TrHoMarginPBGT* parameter is optimised. From the values in the first column, it might be tempting to conclude that self-tuning methods perform worse than the optimised DR. However, this comparison based on P' is biased, since this figure in RRM methods would only reflect the performance of one state, and not the aggregation of the whole trajectory to reach that state from scratch. In contrast, self-tuning methods begin with a network state that is far from optimal (i.e., DR, *DrThreshold*=15), which is progressively improved. For a fair comparison, the assessment should be based on either the best or the last epoch (i.e., p_{min} and p_{eq}, respectively). Based on the best epoch, the performance of most self-tuning methods is close to (or above) that of the optimised DR. FSHMC+FOSLC proves to be the best method based on all criteria.

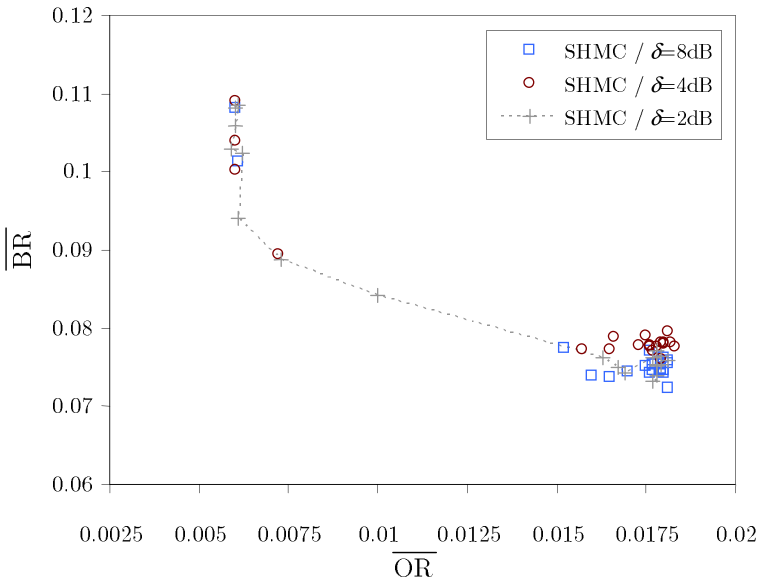


Figure 3.30: Influence of the diffusive parameter in the tuning process.

Concretely, FSHMC+FOSLC has 24% less penalty in the best epoch than the optimised DR. In self-tuning methods with fixed δ (i.e., all except FSHMC+FOSLC), a lower value of P' is achieved for large values of δ. For instance, the overall penalty in SHMC can be reduced by a 5% when changing δ from 2 to 8 dB. From the difference in p_{min}, it might be concluded that the trajectory in the $\overline{OR}$-$\overline{BR}$ plane is not the same for all settings. However, Figure 3.30 proves that the latter is not true by representing the trajectory of SHMC with three different values of δ. For clarity, only points in the curve δ=2 dB have been joint by a dotted line, since they represent the finest trajectory. It is observed that all three settings follow the same trajectory, showing similar performance in equilibrium. From this result, it can be concluded that the only performance difference between settings is the convergence speed, due to the discounted model used to calculate P'.

A closer look on Figure 3.30 reveals that the sensitivity to changes in PBGT HO margins greatly varies from epoch to epoch. This behaviour is more evident in the curve δ=2 dB. In this case, the tuning process consists of a series of small margin steps, all of similar direction and magnitude, until equilibrium is reached. At the beginning of the tuning process (i.e., upper part of the curve), the performance of consecutive epochs is really close, as the distance in the $\overline{OR}$-$\overline{BR}$ plane is small. This is a clear indication that the sensitivity to margin changes is small when margins are highly positive. As tuning evolves, HO margins become negative (i.e., knee of the curve) and consecutive epochs tend to have very different performance. Near equilibrium (i.e., left part of the curve), consecutive epochs show very similar performance again, but as a consequence of smaller margin steps. Hence, it is clear that the performance sensitivity to HO margin changes is not fixed, but depends strongly on the current state of HO margins.

From the previous result, it follows that it is beneficial to modify β as the tuning progresses to improve the convergence speed while keeping the system stable. Such a gain scheduling mechanism is included in FSHMC+FOSLC. To clarify the benefit of this action, Figure 3.31 shows the evolution of p for self-tuning methods with fixed and adaptive β. The performance of the initial state (i.e., DR, *DrThreshold*=15) can be found in the first point of all curves (i,e,, epoch 0). In fixed-β methods, the curve δ=2 dB is roughly a scaled version of the curve δ=8

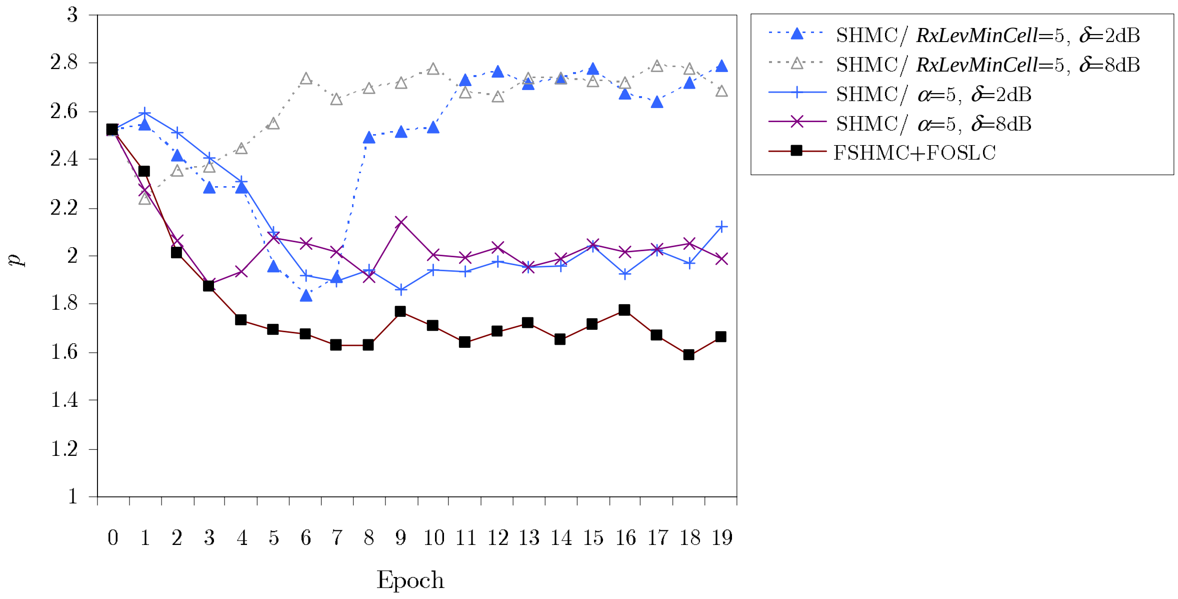


Figure 3.31: Penalty of methods with and without gain-scheduling across epochs.

dB, with a scaling factor of 8/2=4 (i.e., the performance of epoch 1 in curve δ=8 dB should be equal to the one of epoch 4 in curve δ=2 dB). It is then obvious that setting large values of β speeds up the tuning process, which results in a lower overall penalty in the discounted model. It is also observed that the speed of FSHMC+FOSLC is almost the same as the other methods δ=8 dB, as it shares the same δ in the first epochs.

Figure 3.31 also reveals that the self-tuning methods proposed do not necessarily have the lowest penalty in equilibrium. On the contrary, some methods end up with a penalty that is even larger than the one at the origin. This behaviour can also be observed in Table 3.4 from the fact that p_{eq} is significantly larger than p_{min} for some methods. This result can be explained by the fact that the methods described so far aim at balancing blocking between adjacent cells, without checking the final assessment figure during the tuning process. Thus, even if p is initially reduced due to blocking relief, the associated loss of network quality makes that, at some epoch, p starts to increase again. As a result, the final network configuration may be significantly worse in terms of penalty. In Figure 3.31, it is observed that FSHMC+FOSLC has the best penalty performance in the limit, which is confirmed by the fact that FSHMC+FOSLC is the method with the smallest difference between p_{eq} and p_{min} in Table 3.4.

The next analysis checks the stability of self-tuning methods. Figure 3.32 shows the evolution of the average HO margin step, $\overline{\delta HoMarginPBGT_{i \to j}}$, in the adjacencies of the scenario for the fastest methods. For all methods, it is observed that the amplitude of changes diminishes gradually. This behaviour was expected for methods with fixed δ=8 dB, as these should reach the maximum margin values in a few epochs. Likewise, FSHMC+FOSLC has a good convergence rate due to the gain-scheduling mechanism. From the figure, it is also evident that the length of the simulations is enough to ensure that the system has reached equilibrium, as the average HO margin step is less than 0.015 dB. Nonetheless, some performance fluctuations are still observed in Figure 3.31 at the end of the horizon, which are possibly due to the slow-return mechanism and the stochastic nature of simulations.

The following analysis quantifies the impact of the previous strategies on network signalling

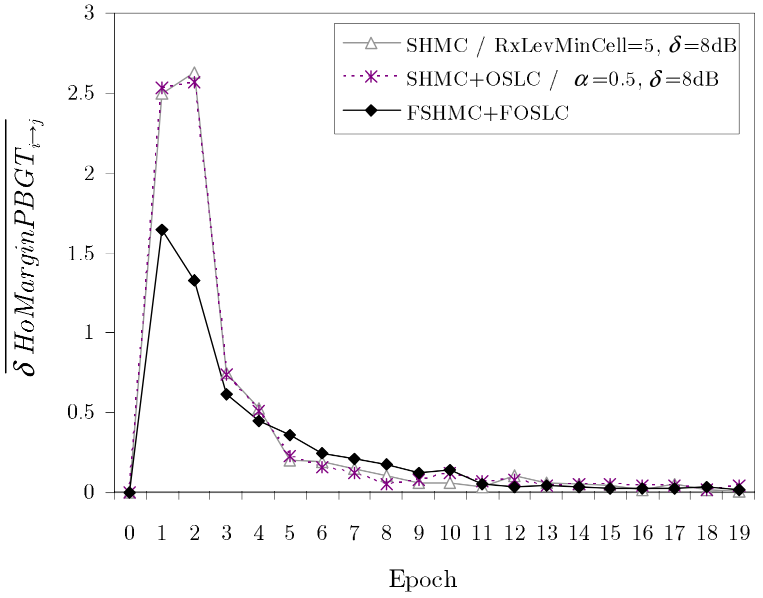


Figure 3.32: Evolution of the average handover margin step in FSHMC+FOSLC.

Type	Method	$\overline{HR}$	MHT[s]
RRM	DR, *DrThreshold*=15	0.32	60.36
	DR, *DrThreshold* =11	0.47	54.28
	TRHO, *TrHoMarginPBGT*=-4 dB	0.39	57.28
Self-tuning	SHMC, *RxLevMinCell*=15, δ=8 dB	1.91	27.24
	SHMC+OSLC, α=0.5, δ=8 dB	1.58	30.91
	FSHMC+FOSLC	1.13	37.47

Table 3.5: Impact of methods on handover signalling load.

load. For simplicity, the analysis for self-tuning methods is restricted to equilibrium. Table 3.5 presents the overall HO ratio, $\overline{HR}$, and the mean holding time, MHT, for different network configurations. It is observed that those strategies that modify cell service area on a permanent basis tend to increase $\overline{HR}$ significantly. Concretely, $\overline{HR}$ in SHMC is six times higher than that of DR only, while MHT is more than halved. The increase of HO signal-level constraints alleviates this problem, but $\overline{HR}$ in FSHMC+FOSLC is still 3.5 times that of DR.

The origin of the decrease in MHT can be understood by studying the dominance area of cells before and after the tuning process. The dominance area of a cell is the set of locations where the cell is the preferred server. From the coverage point-of-view, the dominant cell in position (x,y) is defined as

$$D_{cvg}(x,y) = \arg\max_i \{P_{rx}(x,y,i)\} \quad \forall\, i = 1:N\ , \tag{3.59}$$

where $P_{rx}(x,y,i)$ is the signal level received from cell i in position (x,y). As HO parameters might have been modified, the previous definition is extended to consider the new dominance area defined by the HO process. Thus, the dominant cell in a position (x,y) during connection is defined as

$$D_{ho}(x,y) = \arg\max_{i} \{PBGT_{min}(x,y,i)\} \quad \forall\, i = 1:N \tag{3.60}$$

(i.e., the cell with the highest minimum power budget in the location), where

$$PBGT_{min}(x,y,i) = \min\left\{P_{rx}(x,y,i) - P_{rx}(x,y,j) + \Delta HoMarginPBGT_{i\rightarrow j}\right\} \ \forall\, j = 1:N. \tag{3.61}$$

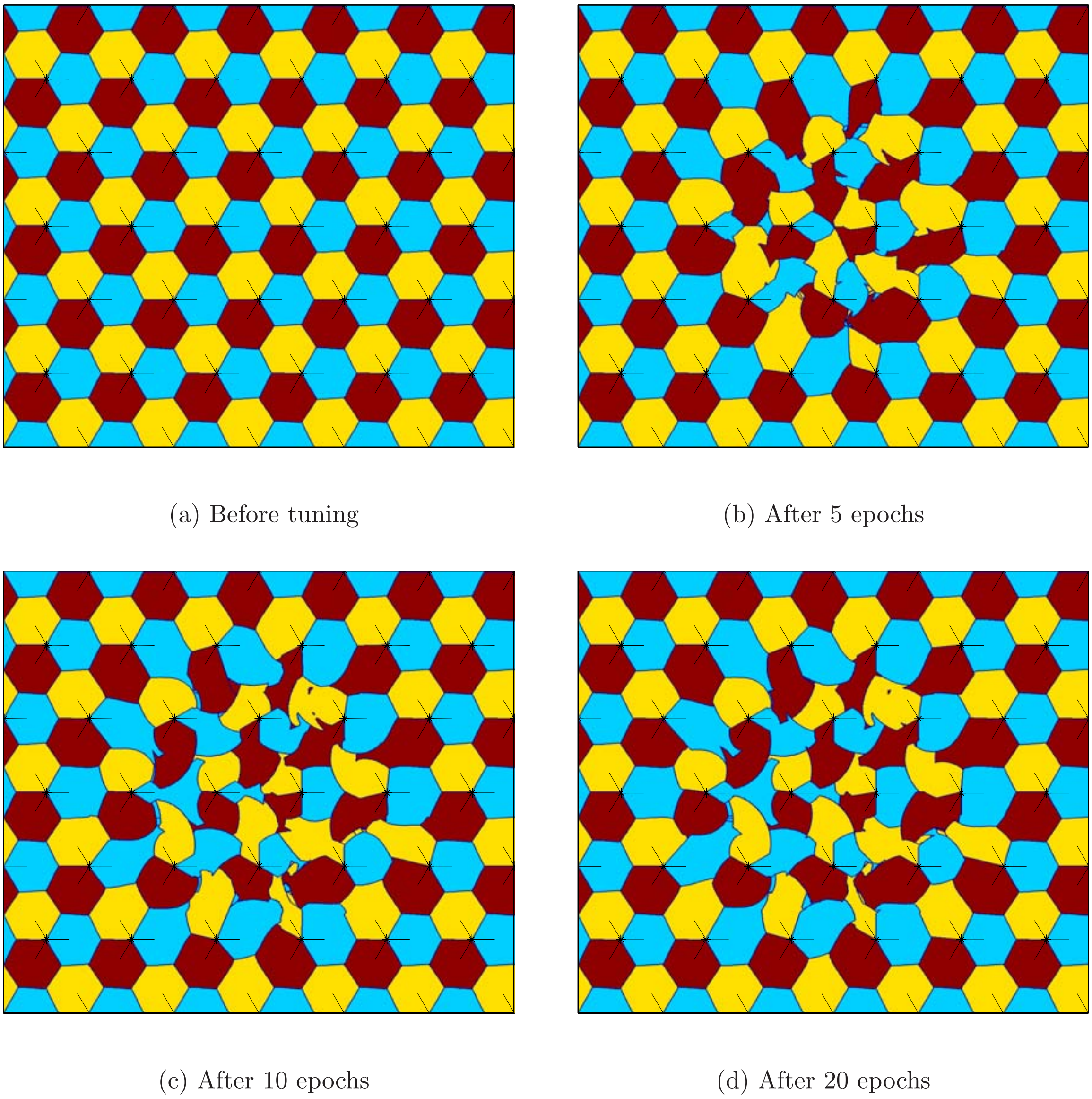
In the previous definition, it is assumed that slow fading is counteracted by hysteresis in the HO margins, and both can thus be neglected. To estimate dominance areas in the test case, the scenario is divided into a regular grid and the best server is computed for each point from static simulations. Figure 3.33 illustrates the HO dominance areas of cells in the scenario before and after the tuning process. First, it is observed that the service area of cells becomes more irregular as the tuning process progresses. Thus, cells in the centre of the scenario reduce their dominance area, while cells in the ends increase their dominance area. As a result, the dominance area of some cells is significantly reduced, which makes that many users accessing these cells are handed over to adjacent cells. This effect is magnified by the fact that some cells have discontinuous service area at the end of the tuning process.

To reduce the number of HOs, the *CellReselectOffset* parameter in CRS is tuned on a per-cell basis by ACRO. As a result, $\overline{HR}$ in FSHMC+FOSLC+ACRO in equilibrium is reduced to 0.67 (i.e., half the one in FSHMC+FOSLC and only twice that of DR only). Likewise, MHT is increased up to 47.56s (i.e., 50% larger than in FSHMC+FOSLC, and only 21% shorter than DR only). The origin of this effect can be understood by observing the dominance area defined by CRS. Thus, the dominant cell in a position during CRS is defined as

$$D_{crs}(x,y) = \arg\max_{i} \{P_{rx}(x,y,i) + \mathit{CellReselectOffset}_i\} \quad \forall\, i = 1:N. \tag{3.62}$$

Figure 3.34 (a)-(b) compare cell dominance areas defined by CRS and HO at the end of the tuning process. For clarity, only a limited area in the centre of the scenario is represented. In both figures, the contour of dominance areas in CRS and HO are shown by orange and black lines, respectively. Figure 3.34 (a) and (b) show the results without and with tuning the *CellReselectOffset* parameter, respectively. From the comparison of both figures, it can be deduced that CRS and HO dominance areas differ less in Figure 3.34 (b), which means that less users will be handed over shortly after the first access. At the same time, $\overline{BR}$ is reduced from 7.9% to 7.5% due to the combined effect of the two congestion-relief mechanisms (i.e., FSHMC and ACRO). As a side effect, the average PBGT HO margin deviation from default settings is reduced from 5.4 to 3.4 dB. From the latter, it might be expected that $\overline{OR}$ would also be lower. However, $\overline{OR}$ increases from 0.9% to 1.1% after tuning CRS offsets. This result indicates that excessive tuning of these parameters might cause that some calls are initiated in cells that fail to provide adequate connection quality.

The last experiment estimates the increase of network capacity when FSHMC+FOSLC is enabled. The old network capacity is defined as the carried traffic in the initial situation with homogeneous parameter settings (i.e., DR, *DrThreshold*=15, *RxLevMinCell*=5). The new network capacity is defined as the maximum carried traffic that ensures that, in the new equilibrium state, $\overline{BR} \leq \overline{BR}_0 = 10.8\%$ and $\overline{OR} \leq \overline{OR}_t = 1\%$ (i.e., blocking is no greater than with the current network configuration and traffic demand, $\overline{BR}_0$, while still satisfying the minimum quality



(a) Before tuning

(b) After 5 epochs

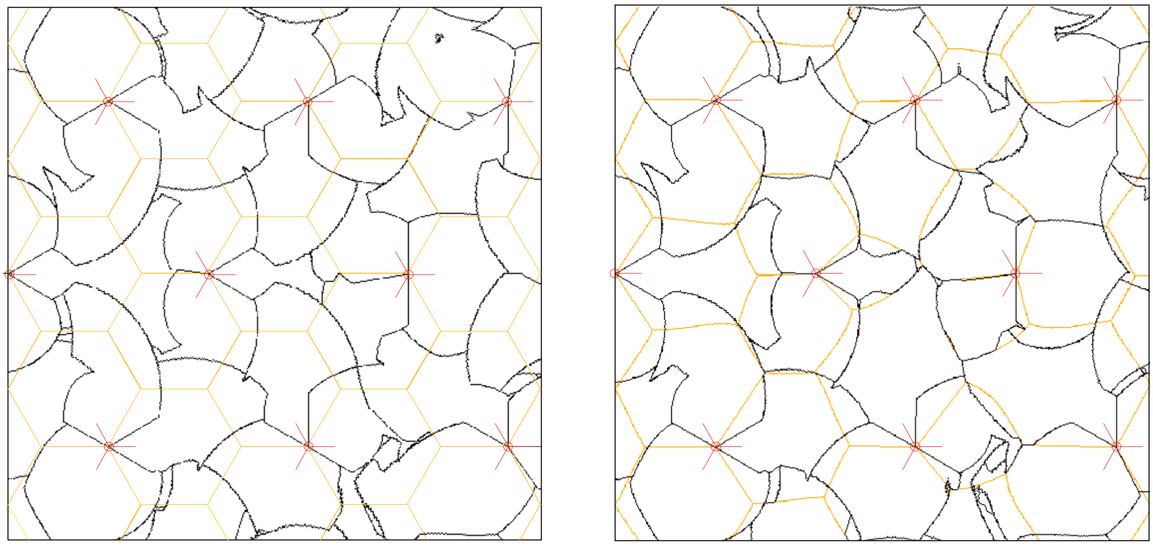
(c) After 10 epochs

(d) After 20 epochs

Figure 3.33: Dominance areas defined by the handover process.

target, $\overline{OR}_t$). In the previous definition, it has been assumed that, from the operator's perspective, there is no extra benefit in enhancing network quality, once the network quality target is ensured. To estimate the new capacity, the traffic demand in the scenario is increased gradually, maintaining the spatial distribution, until one of the two constraints is violated. Simulation results show that, after enabling FSHMC+FOSLC, the average traffic carried per cell can increase from 2.56 E to 2.95 E (15% increase), while still maintaining both requirements. It can thus be concluded that FSHMC+FOSLC is an effective method to increase network capacity when traffic demand is unevenly distributed. It is worth noting that other approaches do not provide any capacity increase in these conditions, since $\overline{OR} > \overline{OR}_t$ in equilibrium with the old traffic demand (i.e., the initial traffic demand should decrease to reach the network quality target).

Finally, it is worth noting that, although only performance averages have been presented for clarity, confidence intervals for these averages prove to be small due to the large number of simulated steps (and, consequently, calls) per epoch.



(a) No *CellReselectOffset* tuning

(b) *CellReselectOffset* tuning

Figure 3.34: Dominance areas defined by cell re-selection and handover processes.

3.6 Conclusions

This chapter has covered the tuning of HO parameters to solve localised congestion in GERAN. The problem has been formulated as a multi-criteria optimisation problem. In the formulation, the main decision variables are the PBGT HO margins and the HO signal-level constraints, while the main assessment criteria are the overall blocking and outage rates in the network. A very simple test case has shown the difficulty of treating the problem analytically, as it is a large-scale non-linear optimisation problem, which is influenced by many factors. Nonetheless, preliminary simulations have shown the potential of this technique.

Several heuristic methods have been proposed to tune HO parameters based on network measurements in a live GERAN. A first method adjusts parameters in the DR algorithm to prevent calls from being re-directed to cells interfered by the original cell. A second method equalises network traffic by adjusting PBGT HO margins on a per-adjacency basis following a diffusive approach. A third method optimises HO signal-level constraints on a per-adjacency basis based on interference conditions in the target cell. Finally, a method is proposed to combine all strategies to extend their congestion-relief capability without causing excessive deterioration of network quality. The fuzzy-logic method proposed jointly optimises HO margins and HO signal-level constraints. Thus, HO signal-level constraints are strengthened in those adjacencies where HO margins become negative. Likewise, HO margins are restricted in those adjacencies where some TRXs in the source and target cell share frequencies. In addition, a gain-scheduling mechanism is provided to speed up the convergence process, while ensuring system stability.

To prove the potential of tuning PBGT HO margins, a field trial has been conducted over a live BSC. Results have shown that network blocking caused by operator's tariff policy can be significantly reduced by a simple diffusive approach. When the method was enabled, the average BH call BR was halved, which caused that the total carried traffic was 3.3% larger.

As the trial covered a limited geographical area, a comprehensive analysis has been performed on a system-level simulator over a test case. The simulated scenario models an extreme situation where cells under congestion are adjacent to each other. During the analysis, the proposed self-tuning methods have been compared with classical congestion-relief approaches. A preliminary analysis has shown the limitations of the classical methods to solve permanent congestion problems, especially under tight frequency reuses. DR can produce a significant impairment in the overall network quality if signal-level constraints in the algorithm are not adjusted properly. Likewise, a simple diffusive approach based on tuning PBGT HO margins on a per-adjacency basis fails to give adequate results for the same reason when HO margins become negative. In the test case, the latter approach causes a three-fold increase of $\overline{OR}$ to reduce $\overline{BR}$ in only 30%. Optimising HO signal-level constraints based on interference in the target cell can help to reduce network quality problems. This restriction is more efficient if signal-level constraints become more restrictive for adjacencies with negative HO margins. In the test case, the fuzzy method that jointly tunes HO margins and signal-level constraints achieves the same reduction of $\overline{BR}$ as the naive approach with half the increase of $\overline{OR}$. The main drawback of modifying PBGT HO margins is the increase of network signalling load due to a larger number of HOs. In the test case, the number of HOs was up to 5 times larger. This problem can be alleviated by tuning CRS offsets on a per-cell basis to synchronise cell service area defined by CRS to the one defined by HO. With this method, the number of HOs after tuning PBGT HO margins was halved.

From these results, it can be concluded that tuning HO signal-level constraints and CRS offsets can greatly improve the performance of tuning PBGT HO margins slowly. Nonetheless, the achievable gains in a live environment will vary depending on the actual traffic demand, propagation conditions and deployed resources. Although the capacity gain would be smaller when a large number of TRXs are deployed per cell (as it is common practice), this would be compensated for by the fact that congested cells tend to be surrounded by cells with spare capacity (which is not the case in the simulated scenario). More importantly, this technique does not need any hardware upgrade, providing a cost-effective means to increase network capacity.

Chapter 4

Summary Conclusions

"We live on an island surrounded by a sea of ignorance.
As our island of knowledge grows, so does the shores of our ignorance."
(John A. Wheeler, American physicist, b. 1911)

The closing chapter summarises the major findings of this thesis. The first section highlights the main original contributions. The next section presents possible directions of future work. The final section gives a list of the publications arising from this thesis. For clarity, the two problems considered in this thesis are treated separately whenever possible.

4.1 Main Contributions

This thesis has dealt with two very different topics: the optimisation of the assignment of cells to PCUs in GERAN and the tuning of HO parameters for congestion-relief in GERAN. The main contributions on these topics are summarised as follows.

4.1.1 Optimisation of the Cell-to-PCU Assignment in GERAN

Many clustering problems in a cellular network can be modelled as a graph partitioning problem. Although several methods have been proposed in the mobile network literature, to the author's knowledge, no rigourous performance comparison of different graph partitioning approaches has been established in this application field so far. One of these clustering problems is the assignment of cells to PCUs in a BSC. Due to effort, expenses and lack of suitable tools, this problem has been solved manually by many operators. As a result, the solution currently configured in the network is often far from optimal.

This thesis has formulated the problem for the first time as a graph partitioning problem. Such a formulation allows to adapt methods from other fields to the problem considered here.

A field trial has shown the limitations of the current operator's approach, proving the need for the optimisation process. Likewise, trial results have shown the potential of a very simple graph partitioning algorithm over a limited geographical area of a live network.

Two graph partitioning methods have been adapted to the cellular environment: an exact method, based on the application of the classical Branch-and-Bound algorithm over an enhanced integer linear-programming model of the problem, and a heuristic method, which extends the classical multi-level refinement algorithm with adaptive multi-start techniques and connectedness checks. The resulting methods can be applied to other clustering problems in the cellular field without much changes.

Finally, this thesis has presented a comprehensive performance analysis of these and other classical graph partitioning methods over an extensive set of graphs constructed with data taken from a live network. This set of graphs should be representative of the graphs handled in this application area. Hence, the results could be extrapolated to other graph partitioning problems in the cellular field.

4.1.2 Optimisation of Handover Parameters for Congestion Relief in GERAN

Congestion relief in cellular networks has been a research topic for the last decade. Thus, a large number of methods have been proposed by the research community to solve the problem. Unfortunately, most of these methods rely on complex network features that are difficult to develop by equipment vendors and are seldom available for operators due to expenses involved. This problem is more evident in mature technologies, where the investment on new equipment must be kept to a minimum. To cope with this problem, GERAN operators often consider the tuning of the service area of cells in the network. Such an effect can be obtained by adjusting HO margins between neighbour cells. However, due to the complexity of the analysis task, tuning is left as a last resort and, when performed, it is based on very simple rules with hard safety constraints. Moreover, to the author's knowledge, no thorough investigation of the limits of this technique has been published.

This thesis has first shown the limitations of classical congestion-relief techniques to cope with persistent localised congestion in GERAN. Preliminary simulations have shown that, although these techniques can be used to effectively re-distribute network traffic, they experience severe limitations due to deterioration of network quality, especially with tight frequency reuses.

A field trial over a limited geographical area has confirmed the potential of tuning PBGT HO margins to cope with the uneven spatial traffic distribution caused by operator tariff policy.

To solve the limitation of classical methods, this thesis has proposed five algorithms to tune different parameters in the CRS, DR and HO algorithms. First, a simple rule to tune signal-level thresholds in the DR algorithm on a per-adjacency basis to prevent users from being re-directed to cells interfered by the original cell. Second, a self-tuning method to modify PBGT HO margins on an adjacency basis to equalise congestion problems between neighbour cells based on a slow diffusive approach. Third, a self-tuning method to optimise HO signal-level constraints on a cell basis based on signal-level and signal-quality statistics. Fourth, a fuzzy self-tuning method to jointly tune PBGT HO margins and HO signal-level constraints on a per-adjacency basis based on the difference in congestion between neighbours, the current value of HO margins in the adjacency and the interference conditions in the target cell. Fifth, a method to tune CRS offsets on a cell basis to synchronise cell service area for mobiles in idle and connected mode, thus minimising the number of HOs in the network.

Finally, this thesis has presented a thorough comparison of the above-mentioned methods against classical congestion-relief methods over an extreme, albeit realistic, scenario built in a system-level simulator.

4.2 Future Work

Due to the diversity and scope of the topics covered in this thesis, there are several issues that remain open and could be explored in the future. Some of these are discussed below.

4.2.1 Optimisation of Cellular Network Hierarchy

In this thesis, two methods have been proposed to solve the CPAP based on existing network resources and past network statistics. The exact method ensures the optimal solution at the expense of an increased runtime. For efficiency, operators would normally use the heuristic method to find high quality solutions. Unfortunately, in this approach, there is no indication of how far the performance of the heuristic solution is from the optimum. Hence, it would be useful to have an efficient method to find bounds for the performance of the optimal solution. Such a method would aim to solve a simplified (relaxed) version of the problem, thus obtaining an upper bound for performance [128][129]. It is worth noting that the main purpose of these methods is to find a performance bound (and not the solution itself), which makes them ideal for the dimensioning stage, when the number of PCUs has still to be decided.

Current mobile networks have a hierarchical structure to achieve scalability. Configuring network structure requires associating elements in each layer to elements of a higher layer. This clustering problem, which must be solved for each layer, can be formulated as a graph partitioning problem, and, consequently, be solved by the methods proposed in this thesis. Among the latter problems are the assignment of sites to BSCs (i.e., BSC-planning or BSC-splitting) and the assignment of cells to location areas (LAs) and routing areas (RAs) (i.e., LA-planning and RA-planning) [57][58]. These problems might have a different formulation, which may require changes in the methods proposed in this work. Such an example is the BSC-splitting problem, where the aim is to re-allocate BTSs in a set of BSCs that has been extended recently. As in other clustering problems, the main goal is to minimise the number of HOs between BTSs in different BSCs, while keeping the load of BSCs evenly distributed. However, in this case, the number of BTSs that change their BSC must be minimised, as every change might require visiting the site and re-configuring links to the BSC. Likewise, the distance between BTS and BSC sites must also be minimised to reduce the cost of trunk infrastructure.

The techniques proposed here can be adapted without much effort to configure network hierarchy in UTRAN. The problem of assigning Node-Bs to Radio Network Controllers is an example [130]. The main difference would be the meaning of HO statistics in UTRAN. In the presence of soft-HO, a terminal can be connected to several cells simultaneously. Thus, HO statistics in the Radio Network Controller reflect the number of times a neighbour cell is included in the active set of cells.

4.2.2 Network Parameter Optimisation for Congestion Relief in Cellular Networks

The self-tuning methods proposed in this thesis aim to balance congestion problems between adjacent cells, without evaluating the actual assessment figure explicitly. The equilibrium state has been proved to be optimal if the objective function is built only from network blocking indicators and network quality is handled as a constraint. However, this is not necessarily the case when the latter terms are included in the objective function, as it is for the assessment figure used in this work. Although these strategies normally achieve an improvement of the overall network performance, in some cases, the impairment of network quality from these strategies might lead to a network configuration that is actually worse than the initial one. In other words, the congestion-relief effect might be not enough to justify a severe deterioration of connection quality. To circumvent this problem, the equilibrium condition between adjacent cells may be extended to consider both blocking and outage indicators [1]. Nonetheless, equalising this new indicator between neighbours, albeit fair, might not be give the best overall performance. This aspect needs further investigation.

The methods proposed have some internal parameters that must be set a priori. Due to the heterogeneity of mobile networks, it is difficult to find settings that perform well in all conditions. Consequently, conservative settings are configured in most cases. To fully exploit the potential of these methods, internal settings may be optimised first on a network basis. For this purpose, reinforcement learning principles [127] could be applied to learn from interaction with the environment. Fuzzy controllers are especially suited for this approach, as there is a well established framework for their adaptation when implemented by neural networks [131]. Thus, parameters in the input and output membership functions, such as mean, deviation and shape, may be fine tuned based on the final assessment figure [132]. Such a strategy can be viewed as a meta-heuristic optimisation technique.

In equilibrium, the self-tuning methods provide a set of cell sizes and shapes with which blocking is equalised as much as possible, given the constraints on the maximum connection quality impairment. However, it remains unknown if other configurations can give the same blocking relief with a better overall connection quality. Thus, it might happen that re-arranging cell shapes, while still maintaining offered traffic per cell, improved the overall connection quality. Therefore, it would be interesting to compute the optimal cell layout in the scenario, which could be used as benchmark. For this purpose, the provision of service by cells in the scenario can be modelled a resource allocation problem, as in [106]. In such a problem, the scenario is divided into a grid of localised traffic demand units, which are assigned individually to one of the cells. The assignment should aim to minimise the total network path losses, such that blocking is the same for all cells and all units are served by a cell that provides adequate level in the location. This approach is valid whenever the exact spatial traffic distribution is known a priori. As this assignment problem is NP-complete, a heuristic must be used (e.g., genetic algorithms [133], utility-based [134], bubble oscillation [106]). It is worth noting that, even if the optimal layout is known, it might not be achieved by tuning PBGT HO margins (i.e., the solution might be infeasible). Hence, this solution gives an upper performance bound.

An obvious extension of this work is the application of these self-tuning techniques in UTRAN. The modification of cell service area on a permanent basis to relief local congestion problems has already been used in UTRAN. However, the mechanisms proposed to achieve this effect are somewhat different (e.g., antenna beamforming [108], regulation of pilot power

[135], load level targets [136][137] and soft handover parameters [138]). The main reason for this has been highlighted in this work. The tuning of HO margins suffers from interference problems in tight frequency reuses, as users handed over to adjacent (sub-optimal) cells may experience large interference from the original (strongest) cell. Nonetheless, similar rules are being considered for the tuning of margins in inter-system HO [139], as cells of different systems do not share frequencies. Likewise, the optimisation of signal-level constraints in UTRAN is easier, as interference is measured explicitly on a cell basis, unlike in GSM.

4.3 List of Publications

The following list presents the publications arising from this thesis.

Articles

[I] V. Wille, S. Pedraza, M. Toril, R. Ferrer, and J. Escobar, "Trial results from adaptive hand-over boundary modification in GERAN," *Electronics Letters*, vol. 39, no. 4, pp. 405–407, Feb 2003 [123].

[II] M. Toril, V. Wille, and R. Barco, "Optimization of the assignment of cells to packet control units in GERAN," *IEEE Communications Letters*, vol. 10, no. 3, pp. 219 – 221, Mar 2006 [20].

Book chapters

[III] R. Barco, A. Kuurne, S. Pedraza, M. Toril. V. Wille, S. Patel and M. Partanen, "Automation and optimization," in *GSM, GPRS and EDGE Performance: Evolution Toward 3G/UMTS*, T. Halonen, J. Melero, and J. Romero, Eds. John Wiley & Sons, 2002, pp. 467–512 [1].

Conferences

[IV] M. Toril, S. Pedraza, R. Ferrer, and V. Wille, "Optimisation of signal-level thresholds in mobile networks," in *Proc. IEEE 55th Vehicular Technology Conference*, vol. 4, May 2002, pp. 1655–1659 [117].

[V] M. Toril, S. Pedraza, R. Ferrer, and V. Wille, "Optimization of handover margins in GSM/GPRS networks," in *Proc. IEEE 57th Vehicular Technology Conference*, vol. 1, May 2003, pp. 150–154 [140].

[VI] I. Jiménez, M. Toril, R. Toril, and O. Fernández, "Análisis del rendimiento de un sistema GSM con distribución no homogénea de tráfico," in *Proc. XIX Simposium Nacional de la Unión Científica Internacional de la Radio (URSI)*, Sep 2004 [141].

Patent Applications

[VII] S. Pedraza and M. Toril, “Method and system for load sharing between a plurality of cells in a radio network system”, Patent application WO 02/104058 A1, Dic 2002. [142]

[VIII] S. Pedraza and M. Toril, “Method for setting parameters of a cellular radio network, in particular for access and handover”, Patent application WO 03/037017 A1, May 2003 [143].

[IX] S. Pedraza and M. Toril, “Method and system for harmonizing an operation area for a mobile device in a cellular radio”, Patent application WO 03/037020 A1, May 2003 [144].

[I, III-IX] are devoted to the optimisation of HO parameters in GERAN, while [II] is devoted to the optimisation of the cell-to-PCU assignment in GERAN. [I, II, V] present field trials results, while [III, VII-IX] only describe parameter tuning algorithms and [IV, VI] present results from network-level simulations.

The author was the primary author of papers [II, IV, V], in addition to being a key contributor to the rest of the papers. In [I, VI], the author was involved in the design, analysis and writing stages, while he also partly developed the simulation tool used in [VI]. In [III], the author wrote the section devoted to parameter optimisation, in addition to helping editors with editorial issues related to the whole chapter. Finally, the author shared the authorship of patent applications [VII, VIII, IX], whose rights belong to Nokia Corporation. [VII] and [VIII] have already been granted in several countries.

[I, III-V, VII-IX] have their origin in the *GERAN Automation* project developed in the *Centro de Ingeniería de Sistemas de Comunicaciones Móviles* of Nokia Spain in Málaga. [II, VI] were developed under the frame of the TIC2003-07827 grant from the Spanish Ministry of Science and Technology, in collaboration with Nokia Networks in the United Kingdom.

Appendix A

Runtime Analysis of ML-CAMS Partitioning Algorithm

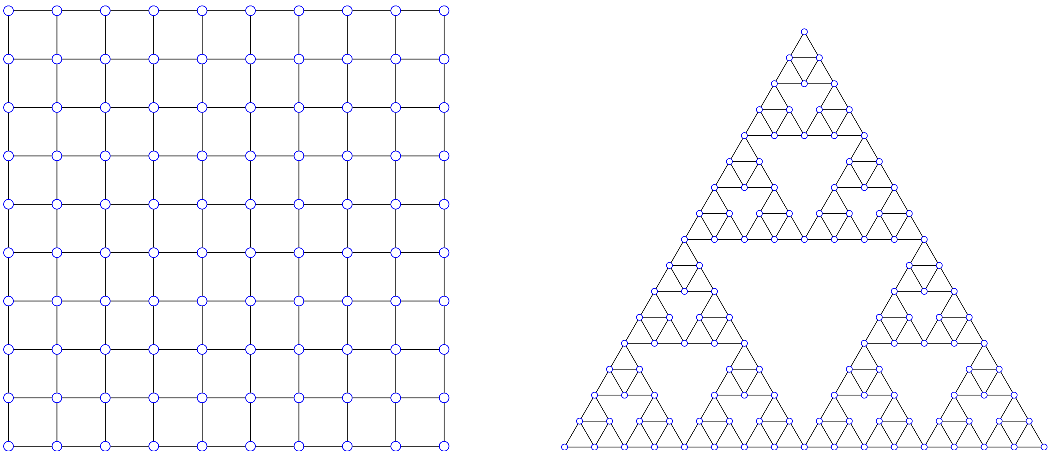
This appendix complements the theoretical time-complexity analysis performed in Section 2.3.4 by evaluating the runtime of the ML-CAMS algorithm proposed in Section 2.3.3. In particular, the main concern is how the runtime of the different parts of the algorithm scales with problem size. For simplicity, the analysis is restricted to the bisection case, although the conclusions drawn here might well be valid for the case of several subdomains. The initial section outlines the methodology behind the experiments to aid the interpretation of results presented in the subsequent section. Results will show that average runtime limits resemble those theoretical worst-case limits presented in Section 2.3.4.

A.1 Analysis Methodology

As the main concern of the analysis is the scalability of the algorithm, the set of graphs used in the experiments should vary in size from a few tens of vertices to many hundreds of vertices. At the same time, selected graphs should also maintain the same properties across different sizes. To keep things simple while still maintaining a reasonable degree of diversity, the analysis considers the two families of planar graphs shown in Figure A.1. Figure A.1 (a) depicts the ubiquitous *2-D regular grid*, typical from meshes in scientific simulations. Figure A.1 (b) depicts the *Sierpinski triangle*, which has the nice property of maintaining its structure across coarsening stages due to its fractal nature.

From the previous figures, it is clear that both types of graphs can be easily scaled up while preserving graph structure. This growth in size can be performed by incrementing the resolution of the grid in regular grid graphs or adding new levels of fractality in Sierpinski graphs. Table A.1 summarises the main parameters of the set of graphs used for the assessment of runtime performance. It is worth noting that, although it would be desirable to control the number of vertices and edges (i.e., $|V|$ and $|E|$, respectively) separately, both quantities display a linear relationship in both types of graphs. Hence, the influence of both parameters cannot be isolated and will thus remain as a single parameter in the analysis.

Graphs in Figure A.1 are unweighted. However, the analysis must deal with weighted graphs, since some of the algorithms are sensitive to vertex and edge weights. Ideally, these weights should resemble as closely as possible those in CPAP instances. This could easily be achieved



(a) 2-D regular grid

(b) Sierpinski fractal

Figure A.1: Graph classes in the analysis.

Name	$\|V\|$	$\|E\|$	$\Delta = \|E\|/\frac{\|V\|(\|V\|-1)}{2}$
Regular grid 10x10	100	180	0.036363636
Regular grid 20x20	400	760	0.00952381
Regular grid 30x30	900	1740	0.004301075
Regular grid 40x40	1600	3120	0.002439024
Regular grid 50x50	2500	4899	0.001568307
Regular grid 60x60	3600	7079	0.001092742
Regular grid 70x70	4900	9659	0.000804746
Regular grid 80x80	6400	12637	0.000617137
Regular grid 90x90	8100	16017	0.000488309
Regular grid 100x100	10000	19797	0.00039598
Sierpinski 0	3	3	1
Sierpinski 1	6	9	0.6
Sierpinski 2	15	27	0.257142857
Sierpinski 3	42	81	0.094076655
Sierpinski 4	123	243	0.032387045
Sierpinski 5	366	729	0.010913991
Sierpinski 6	1095	2187	0.003651298
Sierpinski 7	3282	6560	0.001218398
Sierpinski 8	9843	19680	0.000406298

Table A.1: The set of graphs in the analysis.

by extracting them from a real instance. Unfortunately, the number of vertices and edges in the graphs of Table A.1 differ significantly from those in CPAP instances. Thus, the only viable solution is to generate weights randomly based on their probability distribution. For that purpose, the PDF that best fitted vertex and edge weights in a CPAP instance is estimated first. As weights can only take non-negative values, it seems logical to opt for a log-normal distributional model, for which the mean and standard deviation values are estimated. Two random number generators are then built to give weights to edges and vertices in the graphs according to the estimated log-normal distributions.

Some of the algorithms require an initial partition of the graph as an input. In most cases, a random assignment of vertices to subdomains would suffice, as the final aim of the analysis is runtime and not solution quality. Such a process tends to give subdomains of equal weight (as long as that the number of vertices is not too small), but lacking the connectedness property. For the non-connected refinement algorithm, the resulting partition is often close to the worst-case from the runtime perspective, since almost every vertex movement is a valid candidate for edge-cut reduction and must therefore be evaluated. However, this is not the case for the connected refinement algorithm, as most of the runtime is expected to be spent on computing and updating the articulation vertices of connected subdomains. It is thus clear that there is no need for such operations if the original subdomains are disconnected. For this reason, some post-processing of the random initial partitions is required in the latter case to ensure that all subdomains in the initial partition are connected.

A whole set of random graphs and partitions was built by means of the previous process. For most graphs in Table A.1, the algorithms were tested over a set of 50 instances and the average runtime was measured. Only when the overall computation time was expected to exceed 1 hour, the number of instances was reduced to keep the computational load within reasonable limits.

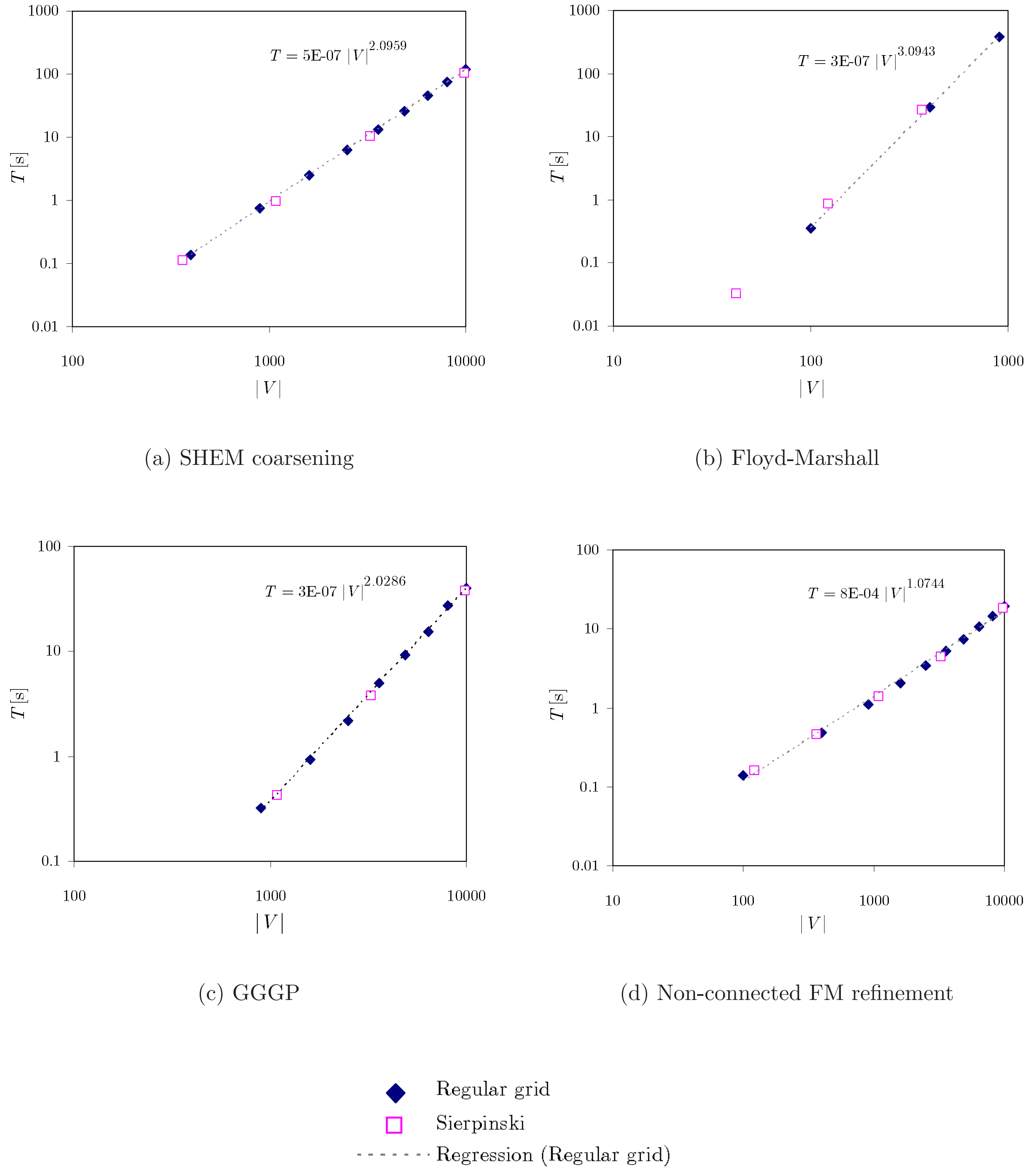
A.2 Results

Figures A.2 (a)-(h) show how the average runtime increases with graph size for the different algorithms considered in the analysis. Hence, these figures provide a rough estimation of their average time complexity. Each point in the figure represents the average runtime (i.e., T) for a certain number of vertices in the graph (i.e., $|V|$). For comparison purposes, results for regular grid and Sierpinski graphs are represented by a different symbol. Both axes in the figures are represented in a logarithmic scale, as it is a convenient way of checking if the runtime is proportional to some power of $|V|$. If this is the case,

$$T = i|V|^n \,, \tag{A.1}$$

$$logT = logi + nlog|V| \,, \tag{A.2}$$

where i is constant and n is the exponent of the power law. Under this assumption, a linear relationship exists between the logarithmic quantities in both axis. Consequently, all points in the figure must lie on a straight line with slope n. Conversely, any deviation from a straight line means that the runtime does not follow a power law. For an easy identification, a trend line is included on each figure, based on the data of regular grid graphs. The equation of this potential regression line is also included for referral purposes.



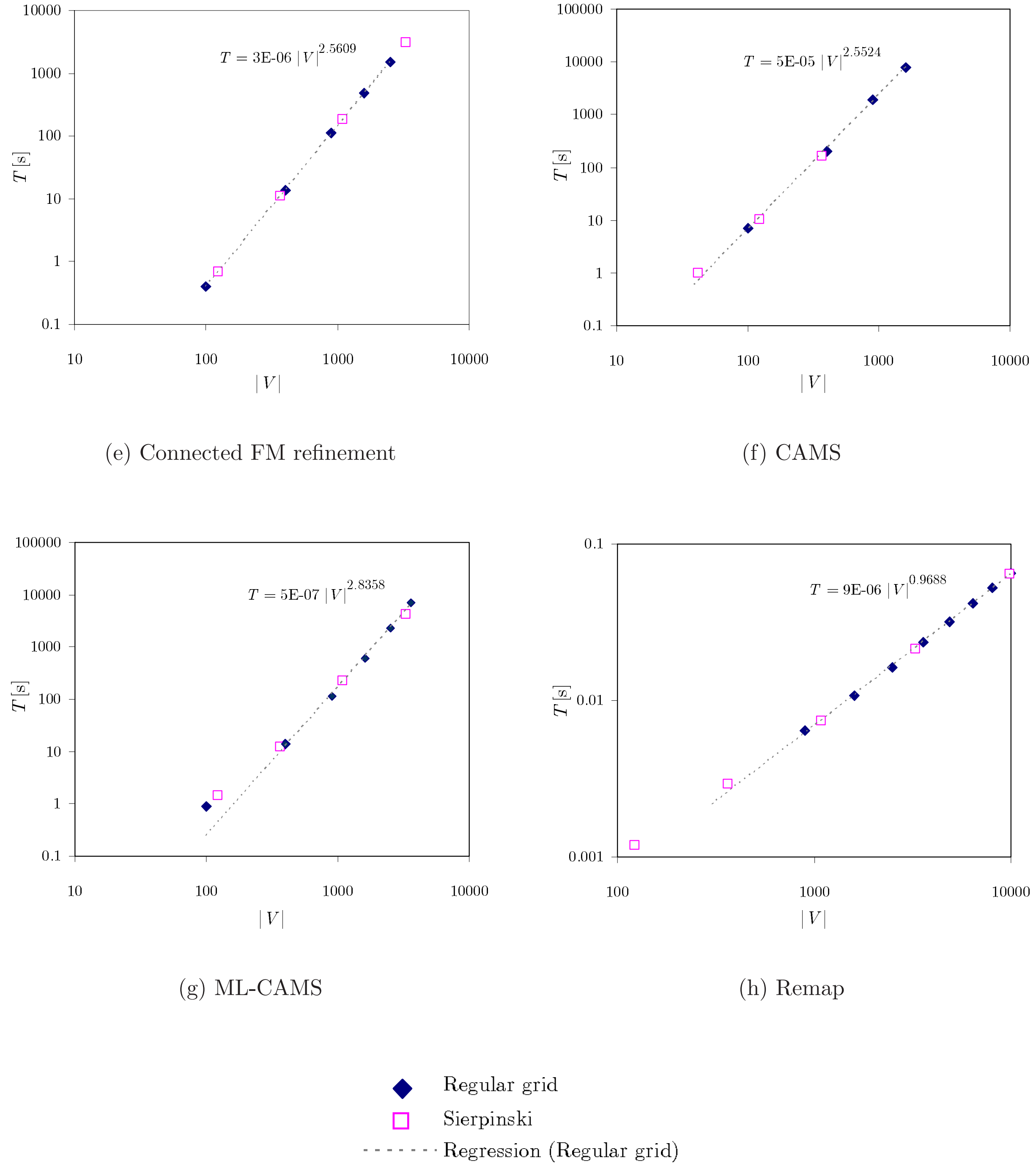
(a) SHEM coarsening

(b) Floyd-Marshall

(c) GGGP

(d) Non-connected FM refinement

Figure A.2: Average runtimes of several algorithms for different graph sizes.



(e) Connected FM refinement

(f) CAMS

(g) ML-CAMS

(h) Remap

Figure A.3: Average runtimes of several algorithms for different graph sizes (cont).

Algorithm	Worst-case	Average-case
Sorted Heavy Edge Matching Coarsening	$O(\lvert E\rvert^2)$	$O(\lvert V\rvert^{2.09})$
Floyd-Warshall	$O(\lvert V\rvert^3)$	$O(\lvert V\rvert^{3.09})$
Greedy Graph Growing Partitioning	$O(\lvert V\rvert^2)$	$O(\lvert V\rvert^{2.03})$
Non-connected FM refinement	$O(\lvert E\rvert)$	$O(\lvert V\rvert^{1.07})$
Connected FM refinement	$O(\lvert V\rvert^2(\lvert V\rvert + \lvert E\rvert))$	$O(\lvert V\rvert^{2.56})$
Clustered Adaptive Multi-Start	$O(\lvert V\rvert^2(\lvert V\rvert + \lvert E\rvert))$	$O(\lvert V\rvert^{2.55})$
Multi-Level Clustered Adaptive Multi-Start	$O(\lvert V\rvert^2(\lvert V\rvert + \lvert E\rvert))$	$O(\lvert V\rvert^{2.84})$
Remapping	$O(k^2\lvert V\rvert)$	$O(\lvert V\rvert^{0.97})$

Table A.2: Worst-case and average-case time complexity of different algorithms.

From a first analysis, it can be deduced that T is indeed related with $|V|$ by a power law in all the algorithms, as every point lies on the regression line. Likewise, no differences in the average runtime performance are observed between regular grid and Sierpinski graphs. Among all algorithms, the Floyd-Warshal algorithm has the largest exponent (i.e., $n \simeq 3$), which confirms that the advanced seed selection in FW-GGGP is extremely inefficient for large graphs. It is also noticeable the different runtime performance of both FM refinement algorithms (i.e., $n \simeq 1$ for non-connected FM refinement, while $n \simeq 2.6$ for connected FM refinement).

Table A.2 compares the average runtime limits against the theoretical worst-case time complexity presented in Section 2.3.4. In the table, it is observed that both limits coincide in most of the algorithms (note that $|E|$ and $|V|$ can be interchangeably used, due to the type of graphs used in the experiments). This is an indication that the theoretical worst-case limits are actually rather tight limits for these algorithms. The connected FM refinement algorithm, and all other algorithms that rely on it (CAMS and ML-CAMS), are exceptions to the previous rule. While the theoretical worst-case time complexity for the former is $O(V^3)$, the average runtime increases with $|V|^{2.56}$. Such a difference is maintained for CAMS and ML-CAMS, which show an exponent of 2.55 and 2.84, respectively.

A closer analysis reveals that the previous discrepancy is mainly due to the implemented version of the DFS algorithm. As explained in Section 2.3.3, a DFS must be performed to check whether a particular vertex is an articulation point. This process is repeated for several vertices after every vertex movement. In the implemented DFS algorithm, the search stops once all vertices adjacent to the analysed vertex are reached, as this condition indicates that the latter is not an articulation point. Only if the original vertex is an articulation point, a full DFS of the subdomain must be performed. From this observation, it is clear that the runtime of the DFS depends strongly on whether the analysed vertex is an articulation vertex or not. In the former case, it is expected that the runtime of the search increases with the size of the subdomain as the whole tree is traversed (i.e., $T \propto |V|$). In the latter case, the runtime should remain relatively constant, as the depth of the search depends only on the local structure of the graph and not on the entire graph. Finally, it is expected that the number of searches increases with $|V|^2$, as a DFS must be executed for every vertex in the original subdomain after each vertex exchange, and both quantities increase with $|V|$. Hence, it can be concluded that the average runtime must increase with $|V|$ as a power law with exponent between 2 and 3, depending on the number of articulation vertices found during the refinement process. This conclusion is reinforced by the results of the experiments.

Appendix B

Optimality Conditions for the Traffic Sharing Problem

This appendix derives the optimality conditions for the two traffic balancing problems described in Section 3.3.1.

B.1 Naive Model

The traffic sharing problem described in Figure 3.12 can be formulated as

$$\begin{aligned} &\text{Minimise} \qquad && A_{bT} = \sum_{i=1}^{N} A_i \cdot E(A_i, c_i) && \text{(B.1)} \\ &\text{subject to} \qquad && \sum_{i=1}^{N} A_i = A\,, && \text{(B.2)} \\ & && A_i \geq 0 \qquad \forall\, i = 1 : N\,. && \text{(B.3)} \end{aligned}$$

This problem has N independent variables, A_i $(i = 1 : N)$, an objective function consisting of a sum of N non-linear terms, $A_i \cdot E(A_i, c_i)$, a linear equality constraint and N inequality constraints.

To solve the problem, it is important to prove problem convexity first, as this assumption simplifies the analysis. The convexity of the objective function in (B.1) with respect to A_i, albeit intuitive, is difficult to prove by direct calculation of the second derivatives. On the contrary, convexity can be intuitively shown from the properties of the traffic overflowing term, $A_i E(A_i, c_i)$, which is known to be a convex function of A_i [145]. Thus, the objective function consists of a sum of convex functions, which is also a convex function. Likewise, the feasible region defined by constraints (B.2) and (B.3) is a convex set[1], because it is the intersection of two convex sets. As both the objective function and the feasible region are convex, the problem is convex. Hence, any local minimum to the problem is a global minimum, or, conversely, any method to compute a local minimum can be used to find the global minimum.

[1] In a *convex* set, the midpoint of any two points in the set is also a member of the set.

Once problem convexity has been proved, the next steps aim to re-formulate the problem as an unconstrained optimisation problem. Firstly, it is assumed that constraint (B.3) is inactive at the optimum. This assumption is easily proved from the fact that, once A_i is zero, further decrements have no effect on the overflowing term, $A_i E(A_i, c_i)$ (as the latter is always non-negative), but cause an increase of the other decision variables to maintain the equality (B.2), which worsens the objective function. Hence, (B.3) can be eliminated without affecting the optimal solution. Secondly, (B.2) is eliminated by solving for one of the decision variables (e.g., A_N) in terms of the others. As a result, the problem can be re-formulated as

$$\text{Minimise} \qquad \sum_{i=1}^{N-1} A_i\, E(A_i, c_i) \;+\; \left(A_T - \sum_{i=1}^{N-1} A_i\right) E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right). \tag{B.4}$$

In such an unconstrained problem, the optimal solution is a stationary point. Hence, the optimal solution must satisfy the stationary condition

$$\nabla A_{bT} = \left(\frac{\partial A_{bT}}{\partial A_1}, \frac{\partial A_{bT}}{\partial A_2}, \cdots, \frac{\partial A_{bT}}{\partial A_{N-1}}\right) = 0 \tag{B.5}$$

(i.e., the gradient of the objective function must be 0) in the optimum. The latter equation can be developed further by derivating the expression of A_{bT} in (B.4) with respect to the decision variables, A_j. This operation results in a set of (N-1) equations

$$\begin{aligned} \frac{\partial A_{bT}}{\partial A_j} &= E(A_j, c_j) + A_j \frac{\partial E(A_j, c_j)}{\partial A_j} \quad - E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right) \\ &+ \left(A_T - \sum_{i=1}^{N-1} A_i\right) \frac{\partial E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right)}{\partial A_j} \;=\; 0 \qquad \forall\, j = 1 : (N-1), \end{aligned} \tag{B.6}$$

which can be re-written as

$$\begin{aligned} E(A_j, c_j) + A_j \frac{\partial E(A_j, c_j)}{\partial A_j} &= E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right) \\ &- \left(A_T - \sum_{i=1}^{N-1} A_i\right) \frac{\partial E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right)}{\partial A_j} \qquad \forall\, j = 1 : (N-1). \end{aligned} \tag{B.7}$$

For symmetry reasons,

$$\frac{\partial E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right)}{\partial A_j} = \frac{\partial E\left(A_T - \sum_{i=1}^{N-1} A_i,\; c_N\right)}{\partial A_k} \qquad \forall\, j, k \tag{B.8}$$

and the right-hand side of (B.7) is equal $\forall\, j = 1 : (N-1)$. Thus, the left-hand side of (B.7) is also equal $\forall\, j = 1 : (N-1)$ and the optimality conditions can be re-formulated as

$$E(A_i, c_i) + A_i \frac{\partial E(A_i, c_i)}{\partial A_i} = E(A_j, c_j) + A_j \frac{\partial E(A_j, c_j)}{\partial A_j} \qquad \forall\, i, j = 1 : N \tag{B.9}$$

and

$$\sum_{i=1}^{N} A_i = A_T \ . \tag{B.10}$$

It is worth noting that, in the latter equations, i and j have been extended to N for symmetry reasons (i.e., the solution should be the same, regardless of the eliminated decision variable). Likewise, (B.10) is needed to avoid the trivial solution of (B.9) $A_1 = A_2 = \cdots = A_N = 0$, which is not a feasible solution.

B.2 Refined Model

The traffic sharing problem shown in Figure 3.14 can be formulated as

$$\text{Minimise} \qquad \mu\, A_{bT} = \sum_{i=1}^{N} \lambda_{f\,i} E(A_i, c_i) \tag{B.11}$$

$$\text{subject to} \qquad \sum_{i=1}^{N} A_i = A_T \ , \tag{B.12}$$

$$A_{lbi} \leq A_i \leq A_{ubi} \qquad \forall\, i = 1 : N, \tag{B.13}$$

where A_{lbi} and A_{ubi} are lower and upper bounds for the offered traffic in cell i. In (B.11), it has been used that the call service rate, μ, only depends on user behaviour and, hence, is the same for all cells. The main difference with the naive model is (B.13), which cannot be eliminated as these constraints may be active in the optimal solution. Hence, the problem must be solved as an optimisation problem with inequality constraints. In these problems, the *Karush-Kuhn-Tucker* (KKT) *multiplier method* [70] can be used to find the optimal solution. The KKT method is a variation of the Lagrange multiplier method used for problems with equality constraint.

Let a problem be formulated as

$$\text{Minimise } \ f(\mathbf{x}) \quad \text{subject to} \quad h_j(\mathbf{x}) = 0, \ g_i(\mathbf{x}) \leq 0, \quad \forall\, j = 1 : N_{eq}, \ i = 1 : N_{ineq}, \tag{B.14}$$

where f is the objective function, h_j and g_i are the equality and inequality functions, and N_{eq} and N_{ineq} are the number of equalities and inequalities, respectively. In such a problem, the KKT method builds the Lagrangian function, $\Phi(\mathbf{x}, \phi, \mathbf{u})$, from a combination of the objective function, $f(x)$, and the constraint functions, $h_j(x)$ and $g_i(x)$, as

$$\Phi(\mathbf{x},\phi,\mathbf{u}) = f(\mathbf{x}) + \sum_{j=1}^{N_{eq}} \phi_j h_j(\mathbf{x}) + \sum_{i=1}^{N_{ineq}} u_i g_i(\mathbf{x}), \qquad u_i \geq 0\,, \tag{B.15}$$

where ϕ_j and u_i are constants (known as *Lagrange multipliers*). For the problem in (B.11)-(B.13), the Lagrangian is

$$\begin{aligned}\Phi(\mathbf{A},\phi,\mathbf{u},\mathbf{z}) = & \sum_{i=1}^{N} \lambda_{f_i} E(A_i,c_i) + \phi(\sum_{i=1}^{N} A_i - A_T) \\ & + \sum_{i=1}^{N} u_i(A_{lbi} - A_i) + \sum_{i=1}^{N} z_i(A_i - A_{ubi}), \qquad u_i, z_i \geq 0\,, \end{aligned} \tag{B.16}$$

where ϕ, u_i and z_i are the Lagrange multipliers associated to constraints (B.12), $A_{lbi} \leq A_i$ and $A_i \leq A_{ubi}$, respectively [146].

The Lagrangian has the nice property that its stationary points are potential solutions to the constrained problem. Consequently, the optimality conditions can be derived by setting the gradient of the Lagrangian equal to zero. In a problem with inequalities, these necessary conditions for a solution to be optimal are referred to as *KKT conditions*. If the problem is convex, as the one considered here, KKT conditions are also sufficient for optimality. For the generalised problem in (B.14), the KKT conditions are

$$\nabla f(\mathbf{x}^*) + \sum_{j=1}^{N_{eq}} \phi_j \nabla h_j(\mathbf{x}^*) + \sum_{i=1}^{N_{ineq}} u_i \nabla g_i(\mathbf{x}^*) = 0, \tag{B.17}$$

$$u_i g_i(\mathbf{x}^*) = 0, \qquad \forall\, i = 1 : N_{ineq}, \tag{B.18}$$

$$g_i(\mathbf{x}^*) \leq 0,\ h_j(\mathbf{x}^*) = 0, \qquad \forall\, i = 1 : N_{ineq},\ j = 1 : N_{eq}, \tag{B.19}$$

$$u_i \geq 0, \qquad \forall\, i = 1 : N_{ineq}, \tag{B.20}$$

where $\mathbf{x}^*$ is the optimal solution. These conditions can be particularised for the problem in (B.11)-(B.13) as

$$\lambda_{f_i} \frac{\partial E(A_i,c_i)}{\partial A_i} + \phi - u_i + z_i = 0, \qquad \forall\, i = 1 : N, \tag{B.21}$$

$$u_i(A_{lbi} - A_i) = 0, \qquad \forall\, i = 1 : N, \tag{B.22}$$

$$z_i(A_i - A_{ubi}) = 0, \qquad \forall\, i = 1 : N, \tag{B.23}$$

$$\sum_{i=1}^{N} A_i = A_T, \tag{B.24}$$

$$u_i, z_i \geq 0, \qquad \forall\, i = 1 : N, \tag{B.25}$$

In (B.21), it has been used that λ_{f_i} does not depend on A_i when computing the Lagrangian partial derivative. While A_i (i.e., offered traffic in a cell) is controlled by the traffic sharing algorithm through optimised HO parameter settings, λ_{f_i} (i.e., new call arrival rate in a cell) is fixed, as it depends only on parameters in the access control algorithm, which remain unchanged.

As the problem is convex, any solution that satisfies (B.21)-(B.25) for any value of A_i, ϕ, u_i and z_i is the optimal solution. Unfortunately, the previous set of equations does not give any information about the values of ϕ, u_i and z_i. (B.21), (B.22), (B.23) and (B.25) can be re-formulated in a more convenient way. From (B.22) and (B.23), it can be deduced that u_i and z_i must be zero when A_i is different from A_{lb} and A_{lb}, respectively. Thus, the values of u_i and z_i reflect whether the inequality constraints (B.13) are active or not in the optimal solution. Therefore, it follows that:

a) If $A_i = A_{lb}$ then $u_i \geq 0$, $z_i = 0$, and, from (B.21),

$$\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} = -\phi + u_i \geq -\phi. \tag{B.26}$$

b) If $A_i = A_{ub}$ then $u_i = 0$, $z_i \geq 0$, and, from (B.21),

$$\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} = -\phi - z_i \leq -\phi. \tag{B.27}$$

c) If $A_{lb} < A_i < A_{ub}$ then $u_i = z_i = 0$, and, from (B.21),

$$\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} = -\phi. \tag{B.28}$$

As ϕ is a constant, it can be deduced from (B.28) that

$$\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} = \lambda_{f_j} \frac{\partial E(A_j, c_j)}{\partial A_j} \tag{B.29}$$

$\forall\ i, j$ where constraint (B.13) is inactive (i.e., $A_{lbi} < A_i < A_{ubi}$). Likewise, from (B.26) and (B.27), it follows that

$$\lambda_{f_u} \frac{\partial E(A_u, c_u)}{\partial A_u}\bigg|_{A_u = A_{ubu}} \leq \lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i} \leq \lambda_{f_l} \frac{\partial E(A_l, c_l)}{\partial A_l}\bigg|_{A_l = A_{lbl}} \tag{B.30}$$

$\forall\ l, u$ where constraint (B.13) is active due to the lower and upper bound, respectively. Thus, the KKT conditions in (B.21)-(B.25) can be substituted by (B.29), (B.30) and (B.24). The latter equations are the basis of the iterative solution methods that aim to equalise the term $(\lambda_{f_i} \frac{\partial E(A_i, c_i)}{\partial A_i})$ $\forall\ i$, and fix $A_i = A_{ubi}$ or $A_i = A_{lbi}$ when decision variables reach their upper or lower limits, respectively [146].

Appendix C

Summary (Spanish)

Este apéndice presenta un resumen en español del trabajo realizado en esta tesis. En primer lugar, se describen brevemente los problemas abordados. Posteriormente, se analiza el estado actual de la investigación y la tecnología, justificando la necesidad del estudio. A continuación se plantean los objetivos de la investigación, junto a la metodología y el plan de trabajo seguido. Finalmente, se presentan los resultados obtenidos, identificando las principales contribuciones de esta tesis.

C.1 Antecedentes

La complejidad de las redes de comunicaciones móviles ha venido incrementándose de manera exponencial. Por un lado, los operadores han ampliado sus redes para hacer frente al incremento de la demanda de servicios móviles. En paralelo, se han introducido nuevas tecnologías y servicios para satisfacer la expectativas de los usuarios. Como problema añadido, el entorno móvil está en continua evolución, lo que a menudo exige la adaptación de la red. Todo ello dificulta enormemente la gestión de este tipo de redes.

En el pasado, los operadores de red han solventado el problema incrementando su plantilla y aceptando la pérdida de eficacia producida por una configuración inadecuada. Como resultado de esta política, los costes de equipamiento y operación han venido incrementándose al mismo ritmo que la complejidad de las redes. Sin embargo, con la entrada de nuevos operadores en el mercado, esta estrategia ha dejado de ser válida, dado el grado de competitividad en el sector. Esta afirmación es especialmente cierta para tecnologías maduras, como GERAN, donde los operadores desean reducir costes cuanto sea posible. Por este motivo, es en estas tecnologías radio donde una gestión de red eficiente resulta más necesaria para ofrecer una adecuada calidad de servicio a un mínimo coste.

Para mejorar la eficiencia de operación, las tareas de gestión de red se han automatizado progresivamente. El objetivo de este proceso de automatización es doble. Por un lado, se persigue librar al operador de trabajos realizados de manera manual, que han de repetirse geográfica o periódicamente. Por otro lado, se desea incrementar el rendimiento de la red diseñando algoritmos de optimización de parámetros que aprovechen la capacidad de cálculo de los computadores. Este hecho justifica el creciente interés por el campo de las *redes auto-ajustables* [1][2][3]. En este contexto, la propiedad de auto-ajuste refleja la capacidad de la red de regular sus parámetros para obtener un funcionamiento óptimo sin intervención del operador.

El problema del ajuste automático de parámetros de una red celular puede solventarse de diferentes maneras. Por un lado, los fabricantes de equipamiento concentran sus esfuerzos en el desarrollo de nuevos algoritmos de gestión de recursos radio (en inglés, *Radio Resource Management*, RRM), que extiendan la funcionalidad de los equipos existentes. Estos algoritmos RRM avanzados permiten modificar el valor de parámetros de algoritmos tradicionales en tiempo real, basándose en medidas instantáneas del estado de la red. Por otro lado, los operadores de red tratan de desarrollar métodos de optimización basados en el equipamiento disponible, para amortizar la inversión ya realizada. Con este objetivo, se extraen medidas estadísticas de los principales indicadores de rendimiento de la red, almacenados de manera periódica en el sistema de gestión de red (*Network Management System*, NMS). A partir del análisis de estos datos, se elabora una propuesta de cambio de parámetros, que, una vez implementada en forma de fichero, se descarga posteriormente en la red. Debido al esfuerzo de desarrollo y coste asociado, el desarrollo de nuevos algoritmos RRM pierde interés conforme se consolida una tecnología radio. Como los operadores dejan de invertir en tecnologías con un horizonte limitado, los fabricantes desvían su atención hacia tecnologías emergentes. Por lo tanto, la replanificación de parámetros desde la NMS juega un papel crucial en tecnologías maduras, tales como GERAN.

El conjunto de parámetros que se puede optimizar en GERAN es extraordinariamente extenso y, por tanto, el alcance de esta tesis es limitado. Este trabajo se concentra en la optimización de parámetros en dos procesos RRM: (a) la (re)selección de celda, y (b) el traspaso [116]. Numerosas razones justifican la selección de dichos procesos. En primer lugar, ambos procesos influyen de manera importante en el rendimiento de una red celular, ya que son los principales encargados de la gestión de la movilidad. Así, la asignación de celdas a unidades de control de paquetes (*Packet Control Units*, PCUs) influye en el retardo del proceso de (re)selección celda. Dicho retardo resulta de vital importancia para los servicios por conmutación de paquetes en las redes actuales, basadas en GPRS, donde no existen mecanismos para ofrecer servicios con restricciones de retardo [4]. Al mismo tiempo, los parámetros de traspaso son el principal mecanismo de control de la calidad de los servicios por conmutación de circuitos en una red celular, ya que aseguran que cada usuario está conectado en todo momento a la mejor celda servidora [5]. En segundo lugar, los métodos de optimización automática de estos parámetros propuestos hasta la fecha son irrealizables o ineficaces. En el caso de la asignación de PCUs, aunque los fabricantes suministran procedimientos de configuración automática en sus BSCs, los operadores rara vez los utilizan debido a sus pobres resultados. En el caso del ajuste de los parámetros de traspaso, los métodos propuestos, o bien se basan en herramientas de análisis que no están actualmente disponibles para el operador, o no garantizan el rendimiento óptimo. Como resultado, los operadores deben optimizar los parámetros mencionados de manera manual. Desafortunadamente, la complejidad del análisis impide que los operadores puedan realizar este proceso de manera periódica, para hacer frente a los cambios en la red. Por esta razón, rara vez se considera la optimización de parámetros, siendo configurados a valores por defecto, aunque esto conlleve una degradación del rendimiento de la red [1]. Con todo ello, se puede concluir que cualquier método que optimice parámetros de los procesos anteriores con el equipamiento existente puede mejorar de manera significativa el rendimiento de las redes basadas en GERAN.

C.2 Estado Actual

A continuación se describe el estado de la investigación y la tecnología, presentando los métodos más relevantes propuestos en el ámbito académico, así como las herramientas de las que disponen los operadores para resolver los dos problemas abordados en esta tesis: la asignación de PCUs y el alivio de problemas de congestión local persistente en GERAN. Por claridad, ambos problemas se tratan de manera independiente.

a) Optimización de la asignación de unidades de control de paquetes en GERAN

Desde el punto de vista teórico, el problema de la asignación de celdas a PCUs se puede formular como un problema de partición de grafos [21]. En este problema se trata de agrupar los elementos de un grafo en subdominios de manera que se minimice la relación entre elementos de diferentes subdominios, cumpliendo un serie de restricciones. La forma más simple de resolver este problema combinatorio es la enumeración del espacio completo de soluciones, pero esta estrategia sólo es válida para grafos de tamaño trivial. Como alternativa, la mayoría de los métodos exactos tratan de reducir el espacio de soluciones enumerado de forma explícita. La estrategia más utilizada es la formulación del problema mediante un modelo de programación lineal entera (*Integer Linear Programming*, ILP) [70], que se puede resolver con el algoritmo de *ramificación y corte* [71] disponible en la mayoría de paquetes de optimización comercial [46].

A pesar de los intentos por mejorar su eficiencia, los métodos exactos siguen siendo costosos computacionalmente. Por este motivo, se han propuesto numerosos algoritmos heurísticos para encontrar soluciones aproximadas de manera eficiente [21]. El algoritmo de *Kernighan-Lin* [47] es el más utilizado en la bibliografía. Otros métodos prometedores son el *refinamiento multinivel* [48][26] y las técnicas *multiarranque adaptativas* [52][28]. El interés por estos métodos heurísticos ha crecido en los últimos años, dadas sus aplicaciones en el ámbito de la supercomputación, el diseño de circuitos integrados y la interconexión de redes de ordenadores. En el contexto de las redes celulares, estos métodos se han aplicado con éxito para la asignación de celdas a centrales de conmutación [40][55][56] y áreas de localización [53][54][57][58] durante la fase de diseño de la red. Sin embargo, el problema de la planificación de PCUs no ha sido estudiado en la literatura, presentando diferencias significativas respecto a los anteriores que justifican su estudio diferenciado.

En cuanto al estado de la tecnología, el proceso de asignación de celdas a PCUs en los equipos actuales se puede llevar a cabo de forma manual o automática, siendo decisión del operador. El algoritmo de asignación automática se ejecuta cuando el operador activa GPRS en una celda. Como principal desventaja, el resultado del método automático es fuertemente dependiente del orden en el que se activa GPRS en las celdas de la BSC. Asimismo, este método se basa únicamente en información estática establecida durante el proceso de planificación (p.ej. adyacencias definidas) y no en estadísticas de red extraídas durante la fase de operación (p.ej. estadísticas de traspaso entre celdas). Por último, los métodos automáticos no consideran la reubicación de celdas en otras PCUs distintas de la inicial cuando, como consecuencia de la inclusión de nuevas celdas, la solución anterior se demuestra inadecuada. Como resultado, estos métodos automáticos tienden a agrupar la mayoría de las celdas en una única PCU, que, al cabo del tiempo, acaba alcanzando su límite de capacidad, mientras que el resto de PCUs quedan vacías. Las celdas que se incluyen posteriormente deben asignarse a PCUs libres, aunque estén fuertemente relacionadas con las celdas existentes, lo que aumenta el número de reselecciones de celda entre PCUs, y, con ello, el retardo en la transmisión de paquetes de datos [31].

Como consecuencia de estas limitaciones, los operadores suelen utilizar el procedimiento de asignación manual, siendo ésta la única opción en EDGE. En dicha aproximación, la diversidad de los datos de configuración de GPRS y las estadísticas de traspaso, distribuidos por diferentes tablas en la NMS, y la complejidad de los métodos de solución, obligan al operador a dar por buenas soluciones ineficaces. Por la misma razón, la optimización de un plan existente casi nunca se considera y, consecuentemente, el número de cambios de un plan de PCUs se restringe a la inclusión de una nueva celda o PCU. Las nuevas celdas se asignan a la PCU que contiene las celdas con mayor número de adyacencias comunes, tal como lo haría el método automático. Como resultado, el número de reselecciones de celdas entre PCUs es alto y la carga se distribuye de manera irregular entre PCUs, lo que deteriora la calidad de los servicios de datos.

Una vez identificada la posibilidad de formular el problema de asignación de PCUs mediante un modelo ILP, existen diferentes paquetes de optimización comercial para resolverlo de manera exacta. Entre otros destacan CPLEX [59], Xpress-MP [60] y LINDO [61]. En el campo de la partición de grafos, existen numerosas herramientas que implementan los métodos heurísticos mencionados anteriormente. Entre las más utilizadas están CHACO [62], PARTY [63], METIS [64], SCOTCH [65] and JOSTLE [66]. Algunas de estas aplicaciones (en concreto, CPLEX, METIS y JOSTLE) se utilizarán en esta tesis.

b) Ajuste de parámetros de traspaso para alivio de congestión en GERAN

La gestión de carga es el problema central de la operación de sistemas distribuidos de gran tamaño. Esto justifica que gran parte de la literatura existente al respecto pertenezca al ámbito de la computación distribuida y las redes de telecomunicación fija.

Desafortunadamente, la mayoría de los algoritmos en los ámbitos anteriores asume que, aunque con retraso, la carga se puede reubicar en cualquier nodo de la red. Obviamente, esta suposición no es válida para el entorno celular, donde los usuarios sólo pueden conectarse a aquellas estaciones base con cobertura suficiente. Por ello, la aplicación de los algoritmos de balance de carga de otros ámbitos no es inmediata.

En al ámbito celular, las fluctuaciones temporales del tráfico se manejan por medio de estrategias dinámicas que reaccionan de manera instantánea a situaciones de congestión, incrementando los recursos de tráfico o reduciendo la demanda de tráfico. Estos métodos se implementan como algoritmos RRM, siendo conocidos como mecanismos de *alivio de congestión*. Entre los más utilizados en GERAN se encuentran la *codificación a velocidad mitad dinámica* [103], el *reintento directo* [104] y el *reparto dinámico de carga* [105][8][9].

Los procesos reactivos anteriores están concebidos para hacer frente a la aleatoriedad del proceso de llamada. Sin embargo, son incapaces de solventar la congestión causada por la concentración espacial del tráfico. Al igual que otros procesos dinámicos, estas técnicas son tendentes a la inestabilidad, lo cual fuerza el ajuste de sus parámetros internos de manera conservadora, con la correspondiente pérdida de rendimiento. Por ello, estos problemas de congestión local persistente se manejan a largo plazo con estrategias de replanificación, como la extensión del número de transceptores o la división de celdas. A corto plazo, la adaptación lenta del área de servicio de las celdas del sistema se mantiene como la única solución para aquellas celdas que no puedan actualizarse rápidamente o simplemente no justifiquen la inclusión de nuevos recursos (p.ej. congestión debida a tráfico estacional).

Para modificar el área de servicio de una celda en GERAN, se han propuesto diferentes técnicas. Un primer grupo modifica parámetros físicos de la estación base, tales como la potencia transmitida [107] o el patrón de radiación de la antena [109][108]. Como estas técnicas requieren acciones de mantenimiento, se utilizan en contadas ocasiones. Como alternativa, un segundo grupo modifica parámetros de algoritmos RRM, lo que es mucho más inmediato. En concreto, el ajuste de los parámetros de (re)selección de celda y traspaso destaca frente a otras técnicas por su simplicidad y efectividad. Un análisis más exhaustivo demuestra que la modificación de parámetros de (re)selección de celda sufre los mismos problemas que el reintento directo, al ser efectivos únicamente durante el establecimiento de llamada [110]. Por ello, la modificación de los márgenes de traspaso es el método más utilizado [6][8][9][7].

Seleccionado el mecanismo de ajuste, faltaría por decidir el algoritmo de optimización. Para solventar problemas localizados de congestión, se han propuesto estrategias proactivas que modelan el ajuste de parámetros como un problema clásico de optimización [6][7]. Como principal ventaja frente a las estrategias reactivas, el uso de un criterio de optimización global en lugar de una regla simple de balance entre celdas vecinas permite asegurar la solución óptima al problema. El proceso clave en estos métodos es el modelado de la distribución espacial de tráfico a partir de medidas de nivel de señal [6] o posicionamiento [7]. Por medio de esta información, el problema de ajuste se modela como un problema tradicional de optimización no lineal multivariable con restricciones, que se resuelve por medio de métodos iterativos [70].

Los métodos proactivos anteriores están concebidos para ser aplicados en la NMS, donde es posible construir modelos analíticos de la red que se optimiza a partir de medidas estadísticas. Sin embargo, las herramientas de recolección de datos y análisis necesarias para la construcción de estos modelos analíticos (o, en su defecto, de simulación) no suelen estar disponibles para el operador. Por ello, en la mayoría de los casos, el algoritmo de optimización debe interactuar directamente con la red real, lo que limita las posibilidades de experimentación. Incluso si se dispusiera de un modelo de simulación, la carga computacional de las simulaciones limitaría el número de posibles configuraciones de parámetros que se pueden evaluar a decenas de combinaciones. Este número es claramente insuficiente para el problema en cuestión, donde cada celda tiene tanto márgenes de traspaso como celdas adyacentes. Como resultado, el problema de ajuste de los márgenes de traspaso suele resolverse mediante métodos heurísticos.

Por simplicidad, la mayoría de los operadores suelen fijar los parámetros a valores por defecto. El ajuste posterior debe realizarse de manera manual después de un arduo trabajo de análisis que debe realizarse celda por celda (o adyacencia por adyacencia), por lo que se realiza en contadas ocasiones. Con ello, se pierde la oportunidad de adaptar estos parámetros, definidos a nivel de celda o adyacencia, a las condiciones del entorno local. En aquellas ocasiones en las que se realiza el ajuste, los operadores suelen restringir el margen de variación de los parámetros, lo que limita el beneficio del proceso de ajuste. Así, no se permite que los márgenes de traspaso tomen valores negativos, lo que se demuestra que limita la capacidad para reubicar la demanda de tráfico en entornos de baja movilidad, donde los fenómenos de congestión son más habituales. Al mismo tiempo, las restricciones de nivel de señal en el traspaso se dejan a valores excesivamente bajos, para evitar un descarte innecesario de celdas candidatas. De esta manera, se elimina el principal mecanismo de restricción que evita los problemas de calidad de conexión generados por los mecanismos de reparto de carga.

En cuanto al estado de la tecnología, se puede decir que la mayoría de los algoritmos RRM de alivio de congestión citados se incluyen de manera opcional en los equipos suministrados por los fabricantes. Sin embargo, dado que los operadores tienen que pagar cantidades importantes por esta funcionalidad adicional, muy pocos se utilizan en la práctica. Mientras que el reintento directo es de uso común, dada su simplicidad y efectividad, el reparto dinámico de carga es poco común por su elevado coste y dificultad para conseguir un funcionamiento estable. Por estas razones, la optimización de parámetros desde la NMS resta como único recurso para aquellas celdas en las que el reintento directo no solventa los problemas de congestión.

La llegada de las primeras herramientas de optimización automática de red ha favorecido el desarrollo de métodos de optimización basados en la NMS. Estas aplicaciones liberan al operador de tareas rutinarias, encargándose de las tareas de recolección de datos, análisis e implementación de los cambios. Como resultado, los operadores pueden ahora desarrollar algoritmos que combinen datos de diversas fuentes, repitiéndose, sin esfuerzo, en el espacio y en el tiempo. Haciendo uso de estas herramientas, en esta tesis se pretende desarrollar métodos de optimización más elaborados que ajusten de manera conjunta varios parámetros de traspaso, adaptándose a las condiciones de tráfico e interferencia del entorno local. De esta manera, se persigue aprovechar al máximo las posibilidades de ajuste de estos parámetros, definidos a nivel de adyacencia.

Para favorecer su puesta en funcionamiento, los algoritmos de ajuste de parámetros desarrollados en esta tesis emularán el proceso de razonamiento del operador, siendo implementados como sistemas basados en reglas. Los sistemas de inferencia difusa [119] resultan especialmente indicados para el diseño de este tipo de controladores, donde ya se dispone de la experiencia de un operador y se ha de manejar información imprecisa, flexible o incierta.

C.3 Objetivos de la Investigación

El principal objetivo de esta tesis es desarrollar procedimientos de optimización automática de parámetros en los procesos de (re)selección de celda y traspaso que puedan ser implementados con la infraestructura de red existente. De manera más específica, esta tesis pretende:

a) Desarrollar métodos de optimización de la asignación de celdas a PCUs en una BSC, basados en estadísticas de red, para reducir el número de usuarios que experimentan un cambio de PCU en GERAN, y

b) Desarrollar métodos de optimización conjunta de los márgenes y restricciones de nivel de señal en el traspaso, basados en estadísticas de red, para solventar problemas de congestión local persistente de tráfico de voz en GERAN.

Como requisito fundamental, todos los métodos propuestos en esta tesis estarán concebidos para interactuar con la NMS por medio de archivos, no requiriendo modificación alguna en el equipamiento de red existente.

C.4 Metodología de Trabajo y Diseño Experimental

En los objetivos anteriores se aprecia que esta tesis cubre dos problemas diferentes, que podrían haberse tratado de manera completamente independiente. Sin embargo, se realizará un esfuerzo por dar a ambos problemas un tratamiento unificado, siguiendo la metodología descrita a continuación.

Todo trabajo científico suele comenzar con la formulación del problema, basándose en una descripción cualitativa del mismo. Tras identificar el problema matemático subyacente, se analiza el estado de la investigación y la tecnología. Las pruebas sobre entornos controlados, reales o de simulación, permiten detectar las principales limitaciones de las técnicas actuales. Posteriormente, se conciben métodos más desarrollados que son validados en entornos de simulación sobre casos simples de prueba. Una vez validados estos métodos por la comunidad científica, los fabricantes de equipamiento evalúan el beneficio en rendimiento y el esfuerzo de desarrollo, realizando diversas simplificaciones sobre el método inicial. Finalmente, estos métodos se despliegan en la red, siendo evaluados en un entorno real, lo que permite en último término evaluar el beneficio real del método simplificado.

En esta tesis se ha seguido la aproximación anterior. Sin embargo, este trabajo tiene varias peculiaridades que merece la pena comentar.

a) Este trabajo trata de solventar problemas que tienen hoy en día los operadores durante los procesos de replanificación de sus redes, siendo, por ello, su marco una red madura como GERAN. Teniendo esto en mente, la formulación inicial del problema en términos cualitativos la realizará directamente el operador. Consecuentemente, tanto los parámetros optimizados como los principales criterios de rendimiento serán seleccionados por el operador, cuyas decisiones no son siempre fáciles de justificar.

b) A partir de esta descripción general, el problema se formulará analíticamente, realizando, cuando sea necesario, las pertinentes simplificaciones. Esta formulación matemática permitirá identificar el tipo de problema solventado, facilitando la búsqueda de métodos de solución a problemas similares en otros ámbitos científicos. Asimismo, la formulación analítica permitirá entender las características de la solución óptima y, si el modelo es preciso, plantear métodos de resolución exacta.

c) Para evitar esfuerzos innecesarios, el enfoque inicial será en métodos simples que puedan probarse de manera rápida en una red real sin mucho trabajo de desarrollo. El objetivo de estas pruebas de campo iniciales será doble: por un lado, evaluar la sensibilidad del rendimiento de la red a la modificación de estos parámetros; por otro lado, poner de manifiesto la ineficacia de los métodos actuales, disponiendo de una cota inferior de la ganancia obtenida por la técnica de optimización. Sólo si se demuestra que un método simple puede tener un impacto significativo en el rendimiento de la red, se considerarán métodos más sofisticados. Este hecho justifica que la validación de los métodos propuestos en esta tesis se inicie con una prueba de campo de un método muy simple, y no con la prueba de métodos más sofisticados sobre un modelo simplificados del sistema. Durante el desarrollo de todos los algoritmos, se tendrán en cuenta las limitaciones de los equipos existentes y las restricciones impuestas por el operador, imprescindible si se pretende probar los algoritmos en una red real.

d) Tras demostrar el potencial de las técnicas propuestas, el estudio se centrará en métodos más sofisticados. La validación se realizarán sobre modelos analíticos construidos a partir de medidas extraídas de la red, cuando sea posible. En su defecto, se realizarán simulaciones en entornos realistas. Idealmente, la evaluación final debería haberse realizado sobre la red real. Sin embargo, los operadores son reticentes a probar algoritmos complejos que modifiquen de manera automática parámetros con gran impacto sobre la red. Aun así, se espera que los resultados obtenidos en los casos de prueba se mantengan en la red real, porque (a) los casos son representativos de una situación real, especialmente en el caso del modelo analítico ajustado con medidas de red, (b) la técnica básica ha sido probada previamente en una red real, y (c) los algoritmos propuesto son bastante intuitivos.

C.5 Plan de Trabajo

A continuación se esboza el plan de trabajo seguido en esta tesis.

a) *Optimización de la asignación de PCUs en GERAN*

  a.1) Búsqueda de bibliografía

    i) Búsqueda de información sobre el proceso de (re)selección de celda y la estrategia actual de asignación de PCUs en GERAN.
    ii) Búsqueda de técnicas ILP.
    iii) Búsqueda de técnicas de partición de grafos en diversas áreas de conocimiento.

  a.2) Formulación de la asignación de PCUs mediante teoría grafos

    i) Formulación de la asignación de PCUs como un problema de partición de grafos clásico en supercomputación.
    ii) Adaptación de la formulación clásica del problema de partición de grafos al entorno celular.

  a.3) Justificación del potencial de la formulación anterior por medio de una campaña de medidas desde un vehículo en una red real.

    i) Desarrollo de herramientas para la recolección de datos de rendimiento de GPRS en un terminal [subcontratado].
    ii) Selección de una aplicación de partición de grafos comercial, basada en un algoritmo de partición de grafos heurístico convencional.
    iii) Desarrollo de interfaz para introducir datos de red en la aplicación anterior, consistente en la configuración de GPRS, las estadísticas de traspaso y el plan de PCUs.
    iv) Realización de campaña de medidas sobre el área cubierta por una BSC antes y después del proceso de optimización [subcontratado].
    v) Análisis de resultados para validar la necesidad del proceso de optimización, basándose en la mejora del rendimiento de GPRS tras descargar la nueva solución.

  a.4) Desarrollo de método exacto de resolución del problema de la asignación de PCUs

    i) Desarrollo de diferentes modelos ILP del problema de asignación de PCUs en una BSC.
    ii) Selección del paquete de optimización para resolver los anteriores modelos mediante el algoritmo de ramificación y corte.
    iii) Desarrollo de estrategia de reparto de tiempo entre instancias cuando existen varias instancias del problema (es decir, BSCs) y restricciones de tiempo de ejecución.

  a.5) Desarrollo de método heurístico de asignación de PCUs

    i) Adaptación de algoritmos heurísticos clásicos de partición de grafos al problema.
    ii) Desarrollo de método heurístico basado en la combinación de técnicas de refinamiento multinivel, técnicas multiarranque adaptativas y chequeos de conectividad.

  a.6) Validación de métodos sobre grafos construidos a partir de datos de una red real

    i) Recolección de configuración de GPRS y estadísticas de traspaso en NMS completa.
    ii) Implementación sobre Matlab de algoritmos de resolución concebidos en esta tesis, junto a algoritmos clásicos propuestos en la bibliografía.
    iii) Selección de modelo ILP, ajuste de parámetros de configuración interna y justificación del beneficio de técnicas multiarranque sobre una instancia del problema.

iv) Aplicación del conjunto de algoritmos sobre el conjunto completo de instancias, variando las restricciones del problema.

v) Análisis comparativo de los métodos en base a los resultados obtenidos.

a.7) Conclusiones

i) Extracción de conclusiones e identificación de líneas futuras de continuación.

ii) Elaboración de documento de tesis.

b) *Ajuste de parámetros de traspaso para alivio de congestión en GERAN*

b.1) Búsqueda de bibliografía

i) Búsqueda de información sobre el proceso de traspaso y la estrategia actual de ajuste de sus parámetros en GERAN.

ii) Búsqueda de técnicas de optimización iterativa, control discreto y lógica difusa.

iii) Búsqueda de técnicas de reparto de carga en diversos ámbitos.

b.2) Formulación del ajuste de márgenes de traspaso como problema de optimización clásico

i) Desarrollo e implementación de simulador dinámico de red GSM sobre Matlab [parcialmente realizado en esta tesis].

ii) Prueba sobre escenario básico en simulador para identificar el tipo de problema de optimización y justificar las limitaciones de la estrategia de ajuste actual del operador.

iii) Modelado analítico simplificado del problema para justificar el balance de tráfico entre celdas adyacentes como solución óptima y derivar ecuaciones de balance óptimo.

b.3) Justificación del potencial de la técnica de regulación de márgenes de traspaso con una prueba de campo en una red real.

i) Desarrollo e implementación de algoritmo básico de difusión de carga entre celdas adyacentes, adaptando el algoritmo RRM clásico para su uso en la NMS.

ii) Aplicación del algoritmo anterior en una BSC de una red real.

iii) Análisis de resultados para justificar la necesidad del proceso de optimización, basándose en la mejora del rendimiento tras descargar la nueva solución.

b.4) Desarrollo de métodos heurísticos de ajuste simultáneo de márgenes y restricciones de nivel en traspaso

i) Desarrollo de algoritmo de difusión de carga entre celdas adyacentes basado en la modificación de márgenes de traspaso a nivel de adyacencia.

ii) Desarrollo de algoritmo de adaptación de restricciones de nivel de señal en el traspaso para considerar interferencia recibida en cada celda.

iii) Desarrollo de algoritmo de ajuste conjunto de márgenes y restricciones de nivel en traspaso para difusión de carga y adaptación a interferencia recibida a nivel de adyacencia.

iv) Desarrollo de algoritmo de ajuste de parámetros de compensación en (re)selección de celda para reducir el número total de traspasos en la red.

b.5) Validación de los métodos anteriores sobre simulador dinámico de red GSM

i) Concepción, desarrollo e implementación de escenario de simulación realista.

ii) Implementación de algoritmos propuestos en simulador, junto a otras técnicas clásicas de alivio de congestión en GERAN.

iii) Análisis comparativo de los métodos en base a los resultados obtenidos.

b.6) Conclusiones

i) Extracción de conclusiones e identificación de líneas futuras de continuación.

ii) Elaboración de documento de tesis.

C.6 Resultados

En esta sección se resumen los principales resultados del trabajo realizado en esta tesis. Al igual que en el resto del documento, los dos problemas abordados se tratan de forma separada.

a) Optimización de la asignación de PCUs en GERAN

Este problema se ha formulado como un problema de partición de grafos. En este formulación, el área de red que se optimiza se modela como un grafo, cuyos vértices y aristas son las celdas y adyacencias de la red, respectivamente. El problema de agrupamiento de vértices para minimizar la relación entre vértices en diferentes subdominios modela la asignación de celdas a las PCU existentes.

Se han propuesto tres modelos matemáticos, basados en la formulación tradicional del problema de partición de grafos. Basándose en la formulación anterior, se han propuesto dos métodos de solución, utilizadas para obtener soluciones exactas o aproximadas al problema. El primer método utiliza el algoritmo de ramificación y corte tradicional para resolver una versión mejorada del modelo ILP convencional del problema de partición de grafos. Este método permite obtener la solución óptima a costa de una gran carga computacional. El segundo método combina el algoritmo de refinamiento multinivel con restricciones de conectividad con técnicas multiarranque adaptativas. El método resultante permite encontrar soluciones de gran calidad de manera eficiente, que, en la mayoría de los casos, cumplen que las celdas de una PCU estén relacionadas geográficamente. Esta última propiedad facilita la comprobación visual de la solución por parte del operador.

Para demostrar la relevancia del problema, se ha realizado una prueba de campo sobre un área geográfica limitada de una red real. Basándose en la campaña de medidas sobre vehículo, la prueba de campo ha demostrado que la interrupción del servicio asociada a las reselecciones de celda entre PCUs es mucho mayor que las realizadas entre celdas de una misma PCU. Concretamente, la interrupción media del servicio para el primer caso es más del doble del valor que en el segundo caso, pudiendo superar en algunos casos los 10 segundos. Este resultado evidencia que el número de reselecciones entre celdas de distintas PCUs deben minimizarse para mejorar el rendimiento de la transmisión de datos de paquetes. Asimismo, la prueba ha puesta de manifiesto que la configuración realizada por el operador de manera manual esté muy lejos de ser óptima. Así, se ha demostrado que incluso un algoritmo de optimización muy básico puede mejorar de forma significativa el rendimiento de la red.

Como la prueba anterior solo cubrió un área geográfica limitada, se ha realizado un análisis exhaustivo sobre un conjunto de grafos construídos a partir de datos de un red GERAN real. El escenario del análisis corresponde al área cubierta por 61 BSCs. Se considera que este conjunto de 61 instancias del problema es suficientemente amplio como para aportar resultados significativos. En ausencia de las estadísticas de movilidad asociadas a los servicios de transmisión de paquetes, se han empleado las estadísticas de traspaso de los servicios orientados a conexión para la construcción de los grafos. Asumiendo que la movilidad de los usuario es similar en ambos modos (es decir, con y sin conexión), el error cometido en el análisis debería ser relativamente pequeño.

Durante el análisis, los métodos propuestos se han comparado con otros métodos clásicos. Los resultados del análisis confirman que la solución actual de la red puede mejorar por cualquier método de partición de grafos heurístico. Este mejora afecta sobre todo a la tasa de (re)selección de celda entre PCUs y el desequilibrio de carga entre PCUs.

El análisis de los métodos exactos ha demostrado los beneficios del modelo ILP del problema mejorad. El algoritmo de ramificación y corte obtiene soluciones con menor tasa de (re)selección de celda entre PCUs (en concreto, un 15% menor) que los método heurísticos. Aunque la carga computacional de estos métodos hace que no puede ser aplicados diariamente, éstos pueden ser utilizados durante la fase de planificación de red, donde la exigencias temporales son menores. Con la disposición de códigos más eficientes y máquina de mayor capacidad de cálculo, estos métodos podrían convertirse en el futuro en una opción válida incluso para la fase de operación.

Entre los método heurístico, el método que combina el algoritmo de refinamiento multinivel con la técnica multiarranque adaptativa es el que obtiene el mejor compromiso entre calidad de solución y tiempo de ejecución. Concretamente, el método consigue reducir la tasa de (re)selección de celda entre PCUs de la red actual en un 80%, que coincide con el mejor método heurístico, basado en la realización de un número exagerado de intentos independientes de partición. Además, el método propuesto reduce a la mitad el desequilibrio de carga entre PCUs en la solución actual, mientras que reduce el número de subdominios desconectados en un orden de magnitud. A partir de estos resultados, se pude concluir que el método propuesto supera a las técnicas multinivel convencionales a coste de un pequeño incremento del tiempo de ejecución. Debe matizarse que, como el tiempo de ejecución para la red completa es del orden de minutos, el cuello de botella no se encuentra exclusivamente en la ejecución del algoritmos, sino también en el acceso alas bases de datos para obtener los datos de entrada a la algoritmo.

A pesar de que el algoritmo propuesto es estocástico (es decir, diferentes ejecuciones pueden dar soluciones diferentes), los resultados demuestran la gran robustez del método. Asimismo, la tasa de (re)selección entre PCUs se degrada de manera ligera con el refuerzo de las restricciones del problema. En particular, el anterior indicador se incrementa únicamente en un 4.6% cuando el máximo desequilibrio de carga entre PCUs de una BSC se modifica de 2 a 1.1. Igualmente, el mismo indicador aumenta en sólo un 1.9% cuando se fuerza que las celdas de un mismo emplazamiento estén en la misma PCU. Al mismo tiempo, la restricción de conectividad entre las celdas de una PCU sólo conlleva un incremento absoluto del 0.2% en la anterior tasa. Finalmente, la combinación de las dos anteriores restricciones da como resultado soluciones mucho más fáciles de comprobar en un mapa, lo que es preferido por la mayoría de los operadores.

A la luz de estos resultados, se puede concluir que el método heurístico propuesto es un firme candidato para la replanificación de la asignación de celdas a PCUs que deben realizar los operadores de GERAN como parte de su rutina diaria.

b) Ajuste de parámetros de traspaso para alivio de congestión en GERAN

El problema se ha formulado como un problema de optimización multiobjetivo. En dicha formulación, las principales variables de decisión con los márgenes de traspaso por balance de potencia, y los principales criterios de rendimiento son las tasas totales de bloqueo y pérdida del enlace. Un caso de prueba sencillo sobre un simulador dinámico de red ha mostrado la dificultad de tratar el modelo de forma analítica, al tratarse de un sistema de optimización no lineal multivariable de gran tamaño muy dependiente del entorno. Aun así, estos primeros experimentos han demostrado el potencial de esta técnica.

Se han propuesto diversos métodos heurísticos para afinar varios parámetros de traspaso basándose en medidas estadísticas de red actualmente disponibles en GERAN. En un primer método, se han ajustado los parámetros del algoritmo de reintento directo para evitar el envío de usuario hacia celdas interferidas por la celda original. El segundo método trata de ecualizar el tráfico en la red siguiendo una estrategia de difusión basada en el ajuste de los márgenes de traspaso por balance de potencia a nivel de adyacencia. El tercer método optimiza las restricciones de nivel de señal a nivel de adyacencia. Finalmente, se propone un método que combina todas las estrategias anteriores para mejorar la capacidad de resolver problemas de congestión sin causar problemas de calidad en la red. El algoritmo difuso presentado optimiza los márgenes y las restricciones de nivel de traspaso conjuntamente, de manera que estas últimas se refuerzan en aquellas adyacencias en las que los primeros pasan a tener valores negativos, como consecuencia del proceso de balance. Asimismo, los márgenes de traspaso se restringen en aquellas adyacencias en las que las celdas origen y destino comparten frecuencias en sus transceptores. Además, se incluye un mecanismo de adaptación de la ganancia de lazo para acelerar el proceso de convergencia, sin producir problemas de inestabilidad.

Para demostrar el potencial del ajuste de los márgenes de traspaso, se ha realizado una prueba de campo sobre una BSC real. Los resultados han demostrado que el bloqueo en la red se puede reducir significativamente con una simple estrategia de difusión del tráfico entre celdas adyacentes. Con este estrategia, la tasa de bloqueo de llamadas en la hora cargada se redujo a la mitad, incrementándose el tráfico total en un 3.3%.

Como la prueba de campo solo cubrió un área geográfica limitada, se consideró adecuado realizar un análisis exhaustivo sobre un caso de prueba en un simulador dinámico de red GSM. El escenario simulador trata de modelar una situación en las celdas congestionadas están próximas entre sí, representando un caso extremo. Durante el análisis, los métodos de autoajuste propuestos se han comparado con técnicas de alivio de congestión clásica.

El análisis preliminar ha demostrado las limitaciones de las técnicas clásicas para solventar problemas de congestión local, sobre todo cuando se utilizan esquemas de reutilización de frecuencia ajustados. Aunque el reintento directo es una técnica con grandes posibilidades, esta técnica puede producir un deterioro de la calidad de la red inaceptable si las restricciones de nivel de señal no se ajustan de manera apropiada. Asimismo, la estrategia de balance de tráfico por difusión básica, basada en el ajuste lento de los márgenes de traspaso, no proporciona buenos resultados por la misma razón cuando los márgenes de traspaso pasan a ser negativos. En el caso de prueba, esta última estrategia triplica la tasa de pérdida de enlace, reduciendo la tasa de bloqueo únicamente en un 30%. Optimizando las restricciones de nivel en traspaso en función de la interferencia de la celda destino se pueden reducir enormemente los problemas de calidad de conexión causados. Este tipo de restricciones es más eficaz si se refuerzan en aquellas adyacencias con márgenes de traspaso negativos. En el caso de prueba, el algoritmo difuso de ajuste de márgenes y restricciones consigue la misma reducción de la tasa global de bloqueo que la estrategia básica, con la mitad del incremento de tasa de pérdida de conexión.

El principal inconveniente de modificar los márgenes de traspaso para equilibrar tráfico en la red es el incremento de señalización asociado al mayor número de traspasos. En el caso de prueba, el número de traspasos con el método clásico de balance de tráfico resultó ser cinco veces mayor. Este problema se puede paliar ajustando los parámetros de (re)selección de celda para sincronizar el área de servicio de celda durante el primer acceso a la red y la conexión posterior. Con este método, el número de traspasos se redujo a la mitad.

A partir de estos resultados, se puede concluir que el método propuesto, que ajusta parámetros en los algoritmos de (re)selección de celda, reintento directo y traspaso es un herramienta interesante para aliviar los problems de congestión localizada con los recursos de red actuales. Aun así, la ganancia final obtenida en un entorno real dependerá de la distribución espacial de tráfico, las condiciones de propagación y recursos desplegados en la red.

C.7 Conclusiones

Con el desarrollo de estos métodos automáticos de optimización se pretende mejorar el rendimiento de las redes celulares actuales, a la vez que se incrementa la eficiencia operacional. Ello debería redundar en una mejor calidad de servicio y un menor coste de operación, influyendo así en el grado de satisfacción del usuario y el coste del servicio.

Un aspecto distintivo de este trabajo es la consideración de aspectos prácticos que a menudo se omiten. Los métodos propuestos tienen en cuenta las restricciones de los equipos actuales, pudiendo implementarse sin mucho esfuerzo. Durante el diseño de los algoritmos, se consideran las restricciones del operador, prestando especial atención a la facilidad de manejo de los algoritmos y la gestión de las soluciones. Esta precaución ha permitido llevar a cabo parte del proceso de evaluación sobre redes reales.

A nivel científico, las principales contribuciones de esta tesis se resumen a continuación.

a) Optimización de la asignación de PCUs en GERAN

a) Se ha formulado por primera vez el problema de las asignación de PCUs como un problema de partición de grafos. La formulación analítica presentada ha permitido resolver el problema de manera exacta.

b) Se han adaptado dos métodos de partición de grafos concebidos en otros ámbitos al entorno celular. Esta adaptación afecta tanto a la formulación del problema, como a los métodos de resolución, consiguiendo así soluciones que se ajustan mejor a las necesidades del operador. Se han propuesto dos métodos novedosos de resolución del problema: (a) un método exacto, basado en el algoritmo de ramificación y planos de corte tradicional, inicializado con la solución de un método heurístico de refinamiento multinivel, que se aplica sobre un modelo ILP del problema, (b) un método heurístico, basado en la combinación de dos técnicas que han sido consideradas hasta la fecha de manera independiente: el refinamiento multinivel y las técnicas multiarranque adaptativas. Como extensión del método anterior, se han considerado las restricciones de conectividad entre celdas de un subdominio y asignación de celdas del mismo emplazamiento a la misma PCU, no consideradas en los métodos clásicos concebidos en el ámbito de la supercomputación. Ambas restricciones pretenden mejorar la consistencia geográfica de las soluciones.

c) Los resultados de las pruebas de campo sobre un área geográfica limitada han demostrado las limitaciones de la aproximación actual del operador, así como el potencial de un algoritmo de partición de grafos básico. Con ello, se ha justificado la necesidad del proceso de optimización.

d) Existen diversos problemas en una red celular que se pueden modelar como un problema de partición de grafos. Aunque muchos de estos problemas se han tratado en la bibliografía de redes celulares, no se ha presentado hasta la fecha una comparación rigurosa de las diferentes técnicas de partición de grafos en este ámbito. Esta tesis ha presentado por primera vez los resultados de diferentes técnicas clásicas de partición de grafos, junto al algoritmo propuesto, sobre grafos construidos a partir de datos de una red celular real. El análisis ha demostrado que estos grafos presentan peculiaridades que justifican la necesidad de comprobar el rendimiento de técnicas ya validadas en otros ámbitos. Los resultados presentados se podrían extrapolar a otros problemas similares en el diseño de la jerarquía de una red celular.

b) Ajuste de parámetros de traspaso para alivio de congestión en GERAN

a) Se han identificado por primera vez las limitaciones de las técnicas clásicas de alivio de congestión para solventar problemas de congestión localizada en GERAN. Los experimentos sobre un simulador dinámico de red han mostrado que, aunque algunas de estas técnicas pueden conseguir redistribuir el tráfico de red, producen problemas de calidad de conexión con esquemas de reutilización de frecuencia ajustados.

b) Se han presentado los resultados de una prueba de campo, que confirman el potencial del ajuste de parámetros de traspaso para hacer frente a la distribución espacial no uniforme del tráfico en un entorno real causada por la política de tarificación del operador.

c) Para resolver los problemas de los métodos clásicos, se han propuesto cinco métodos para ajustar de manera automática diferentes parámetros en los procesos de (re)selección de celda, reintento directo y traspaso: primero, una regla simple para regular las restricciones de nivel de señal en el reintento directo a nivel de adyacencia para evitar el envío de usuario hacia celdas interferidas por la celda original; segundo, un método e ajuste de los márgenes de traspaso a nivel de adyacencia para ecualizar problemas de congestión entre celdas adyacentes basado en un método difusivo; tercero, un método de ajuste de las restricciones de nivel en traspaso a nivel de celda basada en el cálculo de la interferencia recibida a partir de estadísticas de nivel y calidad de señal; cuarto, un método difuso de ajuste conjunto de los márgenes y restricciones de nivel de traspaso a nivel de adyacencia, basado en la diferencia de congestión entre vecinos, el valor actual de los márgenes de traspaso en la adyacencia y la interferencia en la celda destino; quinto, un método de ajuste a nivel de celda de los parámetros de compensación en la (re)selección de celda para sincronizar el área de las celdas durante el primer acceso y la conexión posterior, minimizando así el número de traspasos en la red.

d) Se ha presentado una comparación exhaustiva de los métodos anteriores con estrategias clásicas de alivio de congestión sobre un caso de prueba realista en un simulador dinámico de red.

C.8 Lista de Publicaciones

En el marco de esta tesis se han elaborado las publicaciones que se enumeran a continuación.

Artículos

[I] V. Wille, S. Pedraza, M. Toril, R. Ferrer, and J. Escobar, "Trial results from adaptive hand-over boundary modification in GERAN," *Electronics Letters*, vol. 39, no. 4, pp. 405–407, Feb 2003.

[II] M. Toril, V. Wille, and R. Barco, "Optimization of the assignment of cells to packet control units in GERAN," *IEEE Communications Letters*, vol. 10, no. 3, pp. 219 – 221, Mar 2006.

Capítulo de libro

[III] R. Barco, A. Kuurne, S. Pedraza, M. Toril. V. Wille, S. Patel and M. Partanen, "Automation and optimization," en *GSM, GPRS and EDGE Performance: Evolution Toward 3G/UMTS*, T. Halonen, J. Melero, and J. Romero, Eds. John Wiley & Sons, 2002, pp. 467–512.

Conferencias

[IV] M. Toril, S. Pedraza, R. Ferrer, and V. Wille, "Optimisation of signal-level thresholds in mobile networks," en *Proc. IEEE 55th Vehicular Technology Conference*, vol. 4, May 2002, pp. 1655–1659.

[V] M. Toril, S. Pedraza, R. Ferrer, and V. Wille, "Optimization of handover margins in GSM/GPRS networks," en *Proc. IEEE 57th Vehicular Technology Conference*, vol. 1, May 2003, pp. 150–154.

[VI] I. Jiménez, M. Toril, R. Toril, and O. Fernández, "Análisis del rendimiento de un sistema GSM con distribución no homogénea de tráfico," en *Proc. XIX Simposium Nacional de la Unión Científica Internacional de la Radio (URSI)*, Sep 2004.

Peticiones de patente

[VII] S. Pedraza and M. Toril, "Method and system for load sharing between a plurality of cells in a radio network system", Patent application WO 02/104058 A1, Dic 2002.

[VIII] S. Pedraza and M. Toril, "Method for setting parameters of a cellular radio network, in particular for access and handover", Patent application WO 03/037017 A1, May 2003.

[IX] S. Pedraza and M. Toril, "Method and system for harmonizing an operation area for a mobile device in a cellular radio", Patent application WO 03/037020 A1, May 2003.

[I, III-IX] tratan sobre la optimización de parámetros de traspaso en GERAN, mientras que [II] se dedica a la optimización de la asignación de PCUs en GERAN. [I, II, V] presentan los resultados de las pruebas de campo iniciales, mientras que [III, VII-IX] describen algoritmos de ajuste de parámetros más sofisticados y [IV, VI] presentan resultados de simulaciones de red.

El autor de esta memoria es el autor principal en [II, IV, V], además de contribuir activamente en el resto de artículos. En [I, VI], el autor estuvo involucrado en las fases de diseño, análisis de resultados y escritura, mientras que desarrolló parcialmente la herramienta de simulación empleada en esta tesis, descrita en [VI]. En [III], del cual se han vendido más de 6000 copias entre diversas ediciones, el autor escribió la sección dedicada a la optimización de parámetros. Finalmente, el autor comparte la autoría de las peticiones de patente [VII, VIII, IX], cuyos derechos pertenecen a Nokia Corporation.

[I, III-V, VII-IX] tienen su origen en el proyecto *GERAN Automation* desarrollado en el *Centro de Ingeniería de Sistemas de Comunicaciones Móviles* de Nokia Spain en Málaga. [II, VI] se desarrollaron en el marco del proyecto TIC2003-07827 del Ministerio de Ciencia y Tecnología, con la colaboración de Nokia Networks en el Reino Unido en [II].

References

[1] V. Wille, S. Patel, R. Barco, A. Kuurne, S. Pedraza, M. Toril, and M. Partanen, "Automation and optimization," in *GSM, GPRS and EDGE Performance: Evolution Toward 3G/UMTS*, T. Halonen, J. Melero, and J. Romero, Eds. John Wiley & Sons, 2002, pp. 467–512.

[2] J.Laiho, A.Wacker, and T.Novosad, *Radio Network Planning and Optimisation for UMTS*. John Wiley & Sons, 2002.

[3] A. Mishra, *Fundamental of Cellular Network Planning and Optimisation*. John Wiley & Sons, 2004.

[4] V. Rexhepi, M. Moissio, S. Hamiti, and R. Vaittinen, "Performance of streaming services in GERAN A/Gb mode," in *Proc. IEEE 60th Vehicular Technology Conference*, vol. 6, Sep 2004, pp. 4511–4515.

[5] T. T. Nielsen and J. Wigard, *Performance Enhancements in a Frequency Hopping GSM Network*. Kluwers Academic Publishers, 2000.

[6] T.Chandra, W.Jeanes, and H.Leung, "Determination of optimal handover boundaries in a cellular network based on traffic distribution analysis of mobile measurement reports," in *Proc. 47th IEEE Vehicular Technology Conference*, vol. 1, May 1997, pp. 305–309.

[7] J.Steuer and K.Jobmann, "The use of mobile positioning supported traffic density measurements to assist load balancing methods based on adaptive cell sizing," in *Proc. 13th IEEE Int. Symp. on Personal Indoor and Mobile Radio Communications*, vol. 3, Jul 2002, pp. 339–343.

[8] J.Wigard, T.T.Nielsen, P.H.Michaelsen, and P.Morgensen, "On a handover algorithm in a PCS1900/GSM/DCS1800 network," in *Proc. 49th IEEE Vehicular Technology Conference*, vol. 3, Jul 1999, pp. 2510–2514.

[9] S.Kourtis and R.Tafazolli, "Adaptive handover boundaries: a proposed scheme for enhanced system performance," in *Proc. 51st IEEE Vehicular Technology Conference*, vol. 3, Jul 2000, pp. 2344–2349.

[10] T. Halonen, J. Melero, and J. Romero, *GSM, GPRS and EDGE Performance: Evolution Toward 3G/UMTS*. John Wiley & Sons, 2002.

[11] V. Wille, M. Toril, and R. Barco, "Impact of antenna downtilting on network performance in geran systems," *IEEE Communications Letters*, vol. 9, no. 7, pp. 598 –600, July 2005.

[12] H. Olofsson, S. Magnusson, and M. Almgren, "A concept for dynamic neighbor cell list planning in a cellular system," in *Proc. 7th IEEE Int. Symp. on Personal, Indoor and Mobile Radio Communications*, vol. 1, Oct 1996, pp. 138–142.

[13] V. Wille, A. Kuurne, S. Burden, G. Dunn, and R. Barco, "Simulations and trial results for mobile measurement based frequency planning in GERAN networks," in *Proc. 56th IEEE Vehicular Technology Conference*, vol. 1, Sep 2002, pp. 625–628.

[14] E. Horowitz, S. Sahni, and S. Rajasckaran, *Computer Algorithms: C++*. New York: W. H. Freeman & Co., 1996.

[15] D. E. Knuth, "Big omicron and big omega and big theta," *SIGACT News*, vol. 8, no. 2, pp. 18–24, 1976.

[16] M. Garey and D. Johnson, *Computers and Intractability: A Guide to NP-Completeness*. California: W.H. Freeman and Company, 1979.

[17] The MathWorks, Inc., "Getting started with Matlab (version 6)," July 2002, available at *http : //www.mathworks.com.*

[18] ——, "Matlab function reference, volume 3: P-Z," July 2002, available at *http : //www.mathworks.com.*

[19] W. Rugh and J. Shamma, "Research on gain scheduling," *Automatica*, vol. 36, no. 10, pp. 1401–1425, 2000.

[20] M. Toril, V. Wille, and R. Barco, "Optimization of the assignment of cells to packet control units in GERAN," *IEEE Communications Letters*, vol. 10, no. 3, pp. 219 – 221, Mar 2006.

[21] K. Schloegel, G. Karypis, and V. Kumar, "Graph partitioning for high performance scientific simulations," in *CRPC Parallel Computing Handbook*, J. Dongarra, I. Foster, G. Fox, K. Kennedy, and A. White, Eds. Morgan Kaufmann, 2000.

[22] ——, "Multilevel diffusion schemes for repartitioning of adaptive meshes," *Journal of Parallel and Distributed Computing*, vol. 52, no. 2, pp. 109–124, 1997.

[23] L. Oliver and R. Biswas, "PLUM: Parallel load balancing for adaptive unstructured meshes," *Journal of Parallel and Distributed Computing*, vol. 47, no. 2, pp. 150–177, 1998.

[24] K. Schloegel, G. Karypis, and V. Kumar, "Wavefront diffusion and LMSR: Algorithms for dynamic repartitioning of adaptive meshes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 5, pp. 451–66, May 2001.

[25] B. Hendrickson and R. Leland, "A multilevel algorithm for partitioning graphs," in *Proc. 1995 ACM/IEEE Conference on Supercomputing*. ACM Press, 1995.

[26] G. Karypis and V. Kumar, "Multilevel k-way partitioning scheme for irregular graphs," *Journal of Parallel and Distributed Computing*, vol. 48, no. 1, pp. 96–129, 1998.

[27] C. Walshaw and M. Cross, "Mesh partitioning: a multilevel balancing and refinement algorithm," *SIAM Journal of Scientific Computing*, vol. 22, no. 1, pp. 63–80, 2000.

[28] L. Hagen and A. Kahng, "Combining problem reduction and adaptive multi-start: A new technique for superior iterative partitioning," *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 7, pp. 709–717, 1997.

[29] *3GPP TS 05.08, Digital cellular telecommunications system (Phase 2); Radio subsystem link control*, 3GPP Std., Rev. 8.7.0, Nov 2000.

[30] *3GPP TS 23.060, General Packet Radio Service (GPRS); Service Description; Stage 2*, 3GPP Std., Rev. 6.7.0, Dec 2004.

[31] M. Toril, R. Ferrer, S. Pedraza, V. Wille, and J. J. Escobar, "Optimization of half-rate codec assignment in GERAN," *Wireless Personal Communications*, vol. 34, no. 3, pp. 321 – 331, Aug 2005.

[32] *3GPP TS 23.107, UMTS QoS Architecture*, 3GPP Std., Rev. 6.2.0, Dec 2004.

[33] *3GPP TS 22.105, Service Aspects; Services and Services Capabilities*, 3GPP Std., Rev. 3.5.0, May 1999.

[34] *3GPP TS 43.129, Packet Switched Handover for GERAN A/Gb Mode; Stage 2*, 3GPP Std., Rev. 0.6.0, May 2004.

[35] R. Diestel, *Graph Theory, 3rd ed.; Graduate Texts in Mathematics, Vol. 173.* Springer Verlag, 2005.

[36] C. Walshaw, "Multilevel refinement for combinatorial optimisation problems," *Annals of Operations Research*, vol. 131, pp. 325–372, 2004.

[37] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity.* Prentice-Hall, 1982.

[38] R. Krishnan, R. Ramanathan, and M. Steentrup, "Optimization algorithms for large self-structuring networks," in *Proc. INFOCOM '99*, vol. 1, Mar 1999, pp. 71–78.

[39] S. Chopra and M. R. Rao, "The partition problem," *Mathematical Programming*, vol. 59, no. 1, pp. 87–115, 1993.

[40] A. Merchant and B. Sengupta, "Assignment of cells to switches in PCS networks," *IEEE/ACM Transactions on Networking*, vol. 3, no. 5, pp. 521–526, Oct 1995.

[41] S. Holm and M. Sørensen, "The optimal graph partitioning problem," *OR Spectrum*, vol. 15, no. 1, pp. 1–8, 1993.

[42] E. M. Macambira, N. Maculan, and C. C. de Souza, "A column generation approach for SONET ring assignment," *Networks*, vol. 47, no. 3, pp. 157–171, 2006.

[43] M. Boulle, "Compact mathematical formulation for graph partitioning," *Optimization and Engineering*, vol. 5, no. 3, Sep 2004.

[44] O. Goldschmidt and D. S. Hochbaum, "Polynomial algorithm for the k-cut problem," in *Proc. 29th Annual Symp. On the Foundations of Computer Science*, 1988, pp. 444–451.

[45] ——, "A polynomial algorithm for the k-cut problem for fixed k," *Mathematics of Operations Research*, vol. 19, no. 1, pp. 24–37, 1994.

[46] A. Atamtürk and M. W. Savelsbergh, “Integer programming software systems,” *Annals of Operations Research*, vol. 140, pp. 67–124, 2005.

[47] B. W. Kernighan and S. Lin, “An efficient heuristic procedure for partitioning graphs,” *Bell System Technical Journal*, vol. 49, pp. 291–307, 1970.

[48] T. N. Bui and C. Jones, “A heuristic for reducing fill-in in sparse matrix factorization,” in *Proc. 6th SIAM Conference on Parallel Processing for Scientific Computing*, 1993, pp. 445–452.

[49] C. Walshaw and M. Cross, “Mesh partitioning: A multilevel balancing and refinement algorithm,” *SIAM Journal on Scientific Computing*, vol. 22, no. 1, pp. 63–80, 2000.

[50] M. Cross, A. J. Soper, and C. Walshaw, “A combined evolutionary search and multilevel approach to graph partitioning,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000)*, D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds. Morgan Kaufmann, 2000, pp. 674–681.

[51] P. Korosec, J. Silc, and B. Robic, “Solving the mesh-partitioning problem with an ant-colony algorithm,” *Journal of Parallel Computing*, vol. 30, no. 5-6, pp. 785–801, 2004.

[52] K. D. Boese, A. Khang, and S. Muddu, “A new adaptive multi-start technique for combinatorial global optimizations,” *Operation Research Letters*, vol. 16, pp. 101–113, 1994.

[53] P. Gondim, “Genetic algorithms and location area partitioning problem in cellular networks,” in *Proc. 46th IEEE Vehicular Technology Conference*, May 1996, pp. 1835–1838.

[54] J. Plehn, “The design of location areas in a GSM-network,” in *Proc. 45th IEEE Vehicular Technology Conference*, Jun 1995, pp. 871–875.

[55] D. Saha, A. Mukherjee, and P. S. Bhattacharjee, “A simple heuristic for assignment of cells to switches in a PCS network,” *Wireless Personal Communications*, vol. 12, pp. 209–224, 2000.

[56] S. Pierre and F. Houeto, “A tabu-search approach for assigning cells to switches in cellular mobile networks,” *Computer Communications*, vol. 25, no. 5, pp. 465–478, Mar 2002.

[57] I. Demirkol, C. Ersoy, M. U. Caglayan, , and H. Delic, “Location area planning and cell-to-switch assignment in cellular networks,” *IEEE Transactions on Wireless Communications*, vol. 3, no. 3, pp. 880–890, May 2004.

[58] P. S. Bhattacharjee, D. Saha, and A. Mukherjee, “An approach for location area planning in a personal communication services network (PCSN),” *IEEE Transactions on Wireless Communications*, vol. 3, no. 4, pp. 1176–1187, Jul 2004.

[59] K. Holmstrom, A. Goran, and M. Edvall, *User’s guide for TOMLAB/CPLEX v9.1*, TOMLAB Inc., May 2005.

[60] C. Guéret, C. Prins, and M. Sevaux, *Applications of optimization with Xpress-MP.* Dash Optimization, 2002.

[61] *LINDO API Users Manual*, LINDO Systems Inc., 2002.

[62] B. Hendrickson and R. Leland, *The Chaco user's guide; Version 2.0, Technical Report SAND94-2692*, Sandia National Laboratories, 1994.

[63] R. Preis and R. Diekmann, *The PARTY Partitioning-Library, User Guide, Technical Report TR-RSFB-96-024*, University of Paderborn, Germany, 1996.

[64] G. Karypis and V. Kumar, *A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices Version 4.0*, University of Minnesota, Department of Computer Science / Army HPC Research Center, 1998.

[65] F. Pellegrini, *Scotch and libScotch 3.4, User's Guide, Research Report 1264-01*, Universite of Bourdeaux, France, 2001.

[66] C. Walshaw, *The JOSTLE executable user guide: Version 3.1*, School of Computing & Mathematical Sciences, University of Greenwich, 2005.

[67] A. Doig and A. Land, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, pp. 497–520, 1960.

[68] C. Roucairol and P. Hansen, "Cut cost minimization in graph partitioning," *Numerical and Applied Mathematics*, pp. 585–587, 1989.

[69] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms.* McGraw-Hill Higher Education, 2001.

[70] S.S.Rao, *Engineering optimization: Theory and Practice, 3^{rd} edition.* John Wiley & Sons, 1996.

[71] G. Nemhauser and L. Wolsey, *Integer and Combinatorial Optimization.* John Wiley & Sons, 1999.

[72] E. L. Johnson, G. L. Nemhauser, and M. W. Savelsbergh, "Progress in linear programming-based algorithms for integer programming: An exposition," *INFORMS J. on Computing*, vol. 12, no. 1, pp. 2–23, 2000.

[73] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 359–392, 1998.

[74] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite.* Prentice Hall Professional Technical Reference, 1981.

[75] M. T. Heath and P. Raghavan, "A cartesian parallel nested dissection algorithm," *SIAM J. Matrix Anal. Appl.*, vol. 16, no. 1, pp. 235–253, 1995.

[76] J. R. Gilbert, G. L. Miller, and S.-H. Teng, "Geometric mesh partitioning: Implementation and experiments," *SIAM Journal on Scientific Computing*, vol. 19, no. 6, pp. 2091–2110, 1998.

[77] C. Fiduccia and R. Mattheyses, "A linear time heuristic for improving network partitions," in *Proc. 19th ACM/IEEE Design Automation Conference*, 1982, pp. 175–181.

[78] C. Farhat, “A simple and efficient automatic fem decomposer,” *Computers and Structures*, vol. 28, no. 5, pp. 579–602, 1988.

[79] A. Gupta, “Fast and effective algorithms for graph partitioning and sparse matrix ordering,” *IBM Journal of Research and Development*, vol. 41, no. 1/2, pp. 171–184, 1997.

[80] C. Walshaw and M. G. Everett, “Multilevel Landscapes in Combinatorial Optimisation; Tech. Rep. 02/IM/93,” Tech. Rep., Aug 2002.

[81] R. Martí, “Multistart methods,” in *Handbook on Metaheuristics*, F. Glover and G. Kochenberger, Eds. Kluwer, 2000, pp. 355–368.

[82] C. G. E. Boender, A. H. G. Rinnooy Kan, G. T. Timmer, and L. Stougie, “A stochastic method for global optimization,” *Mathematical Programmming*, vol. 22, pp. 125–140, 1982.

[83] H. Mühlenbein, M. Gorges-Schleuter, and O.Kramer, “Evolution algorithms in combinatorial optimization,” *Journal of parallel Computing*, vol. 7, no. 1, pp. 65–85, Apr 1988.

[84] S. Dash, “An exponential lower bound on the length of some classes of branch-and-cut proofs,” in *Proc. 9th Int. Conference on Integer Programming and Combinatorial Optimization*. Springer-Verlag, 2002, pp. 145–160.

[85] R. D’Agostino and M. Stephens, *Goodness-of-fit techniques*. Marcel Dekker, 1986.

[86] J. Chambers, W. Cleveland, B. Kleiner, and P. Tukey, *Graphical Methods for Data Analysis*. Wadsworth, 1983.

[87] A. Baier and K. Bandelow, “Traffic engineering and realistic network capacity in cellular radio networks with inhomogeneous traffic distribution,” in *Proc. 47th IEEE Vehicular Technology Conference*, May 1997, pp. 780–784.

[88] D. Grillo, R. Skoog, S. Chia, and K. Leung, “Teletraffic engineering for mobile personal communications in ITU-T work: The need to match practice and theory,” *IEEE Personal Communications Magazine*, vol. 5, no. 6, pp. 38–58, Dec 1998.

[89] U.Gotzner, A.Gamst, and R.Rathgeber, “Spatial traffic distribution in cellular networks,” in *Proc. 48th IEEE Vehicular Technology Conference*, vol. 2, May 1998, pp. 1994–1998.

[90] S.Almeida, J.Queijo, and L.Correia, “Spatial and temporal traffic distribution models for GSM,” in *Proc. 50th IEEE Vehicular Technology Conference*, vol. 1, May 1999, pp. 131–135.

[91] V. Iversen, *Teletraffic Engineering Handbook*. ITU-D SG 2/16 & ITC, 2002.

[92] W. Mande, “Evaluation of a proposed handover algorithm for the GSM cellular system,” in *Proc. 40th IEEE Vehicular Technology Conference*, May 1990, pp. 264–269.

[93] G. P. Pollini, “Trends in handover design,” *IEEE Communications Magazine*, vol. 34, pp. 82–90, Mar 1996.

[94] Y. Okumura, E. Ohmoni, T. Kawato, and K. Fukuda, “Field strength and its variability in VHF and UHF land-mobile radio service,” *Review of the Electrical Communication Laboratory*, vol. 16, no. 9-10, pp. 825–873, 1968.

[95] R. Vijayan and J. M. Holtzman, “A model for analyzing handoff algorithms,” *IEEE Transactions on Vehicular Technology*, vol. 42, pp. 351–356, 1993.

[96] S. T. S. Chia, “The control of handover initiation in microcells,” in *Proc. 41st IEEE Vehicular Technology Conference*, May 1991, pp. 531–536.

[97] W. Stadler, “Fundamentals of multicriteria optimization,” in *Multicriteria Optimization in Engineering and in the Sciences*, W. Stadler, Ed. New York: Plenum Press, 1988, pp. 1–25.

[98] R. Lüling, B. Monien, and F. Ramme, “Load balancing in large networks: A comparative case study,” in *3th IEEE Symposium on Parallel and Distributed Processing*, 1991.

[99] O. Kremien and J. Kramer, “Methodical analysis of adaptive load sharing algorithms,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 3, no. 6, November 1992.

[100] H. Kameda, J. Li, C. Kim, and Y. Zhang, *Optimal load balancing in distributed computer systems*. Springer-Verlag, 1997.

[101] T. L. Casavant and J. G. Kuhl, “A taxonomy of scheduling in general-purpose distributed computing systems,” *IEEE Transactions on Software Engineering*, vol. 14, no. 2, pp. 141–154, 1988.

[102] P.-O. Osland and P. J. Emstad, “On load balancing in distributed telecommunication systems,” in *SMARTNET*, 1999, pp. 363–374.

[103] *3GPP TS 06.02, Half-rate speech; Half-rate speech processing functions; GSM-Phase2+, Release 99*, 3GPP Std., Rev. 8.0.0, Jul 2000.

[104] B.Eklund, “Channel utilization and blocking probability in a cellular mobile telephone system with directed retry,” *IEEE Transactions on Communications*, vol. 34, no. 4, pp. 329–337, Apr 1986.

[105] J.Karlsson and B.Eklund, “A cellular mobile telephone system with load sharing - an enhancement of directed retry,” *IEEE Transactions on Communications*, vol. 37, no. 5, pp. 530–535, May 1989.

[106] L. Du, J. Bigham, and L. G. Cuthbert, “A bubble oscillation algorithm for distributed geographic load balancing in mobile networks.” in *INFOCOM*, 2004.

[107] J. Kojima and K. Mizoe, “Radio mobile communication system wherein probability of loss of calls is reduced without a surplus of base station equipment,” U.S. Patent 4435840, Mar 1984.

[108] C. Saraydar and A. Yener, “Adaptive cell sectorization for CDMA systems,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 6, pp. 1041 – 1051, Jun 2001.

[109] D. Lee and C. Xu, “Mechanical antenna downtilt and its impact on system design,” in *Proc. 47th IEEE Vehicular Technology Conference*, vol. 2, May 1997, pp. 447–451.

[110] N. Papaoulakis, D. Nikitopoulos, and S. Kyriazakos, “Practical radio resource management techniques for increased mobile network performance,” in *12th IST Mobile and Wireless Communications Summit*, June 2003.

[111] J. Nelder and R. Mead, “A simplex method for function minimization,” *Computer Journal*, vol. 19, no. 7, pp. 308–311, 1965.

[112] P. Stone and M. Veloso, “Multiagent systems: A survey from a machine learning perspective,” *Autonomous Robots*, vol. 8, no. 3, pp. 345–383, July 2000.

[113] D. L. Jagerman, “Some properties of the erlang loss function,” *Bell System Tech. J.*, vol. 53, no. 3, pp. 525–551, Mar 1974.

[114] D. Hong and S. Rappaport, “Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedure,” *IEEE Transactions Vehicle Technol.*, vol. 35, no. 3, pp. 77–92, 1986.

[115] C. Kelley, *Iterative solution of linear and non-linear equations.* Society for Industrial an Applied Mathematics, 1995.

[116] *3GPP TS 45.008, Radio subsystem link control*, 3GPP Std., Rev. 5.4.0, Nov 2001.

[117] M. Toril, S. Pedraza, R. Ferrer, and V. Wille, “Optimisation of signal-level thresholds in mobile networks,” in *Proc. 55th IEEE Vehicular Technology Conference*, vol. 4, May 2002, pp. 1655–1659.

[118] M. Chiani, E. Agrati, M. Mezzetti, and O. Andrisano, “Frequency and interference diversity in slow frequency hopping multiple access systems,” in *Proc. 7th IEEE Personal Indoor and Mobile Radio Communications*, vol. 2, Oct 1996, pp. 648–652.

[119] T. Ross, *Fuzzy logic with engineering applications.* McGraw-Hill, 1995.

[120] M. S. T. Takagi, “Fuzzy identification of systems and its application to modelling and control,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, pp. 116–132, Jan 1985.

[121] C. C. Lee, “Fuzzy logic in control systems: Fuzzy logic controller, Part II,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 20, no. 2, pp. 404–435, 1990.

[122] C.-Z. Xu and F. C. M. Lau, “Optimal parameters for load balancing using the diffusion method in k-ary n-cube networks.” *Information Processing Letters*, vol. 47, no. 4, pp. 181–187, 1993.

[123] V. Wille, S. Pedraza, M. Toril, R. Ferrer, and J. Escobar, “Trial results from adaptive hand-over boundary modification in GERAN,” *Electronics Letters*, vol. 39, no. 4, pp. 405–407, Feb 2003.

[124] B. Ahn, H. Yoon, and J. Cho, “A design of macro-micro CDMA cellular overlays in the existing big urban areas,” *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2094–2104, Oct 2001.

[125] S. Engstrom, T. Johansson, F. Kronestedt, M. Larsson, S. Lidbrink, and H. Olofsson, “Multiple reuse patterns for frequency planning in GSM networks,” in *Proc. 48th IEEE Vehicular Technology Conference*, vol. 3, May 1998, pp. 18–21.

[126] M. E. Anagnostou and G. C. Manos, “Handover related performance of mobile communication networks,” in *Proc. 41th IEEE Vehicular Technology Conference*, vol. 1, Jun 1994, pp. 111–114.

[127] L. Kaebling, M. Littman, and A. Moore, "Reinforcement-learning: A survey," *Journal of Artificial Intelligence Research*, vol. 4, pp. 237–285, May 1996.

[128] P. Pardalos, F. Rendl, and H. Wolkowicz, "The quadratic assignment problem: a survey and recent developments," in *Quadratic assignment and related problems*, P. Pardalos and H. Wolkowicz, Eds. Amer. Math. Soc., 1994, pp. 1–42.

[129] N. Sensen, "Lower bounds and exact algorithms for the graph partitioning problem using multicommodity flows," in *Europ. Symp. on Algorithms, 2001, Lecture Notes in Computer Science*, 2001, vol. 2161, pp. 391–403.

[130] H. Holma and A. Toskala, Eds., *WCDMA for UMTS*. John Wiley & Sons, 2002.

[131] C. Lin and C. Lee, "Neural-network-based fuzzy logic control and decision system," *IEEE Transactions on Computers*, vol. 40, no. 12, pp. 1320–1336, Dec 1991.

[132] L. Giupponi, J. Perez-Romero, R. Agusti, and O. Sallent, "A novel joint radio resource management approach with reinforcement learning mechanisms," in *Proc. 24th IEEE International Performance, Computing, and Communications Conference*, Apr 2005, pp. 621–626.

[133] L. Du, J. Bigham, L. Cuthbert, C. Parini, and P. Nahi, "Cell size and shape adjustment depending on call traffic distribution," in *Proc. IEEE Wireless Communications and Networking Conference*, Mar 2002.

[134] L. Du, J. Bigham, and L. Cuthbert, "Utility-based distributed geographic load balancing in mobile cellular networks," in *Proc. 4th Int. Conf. on 3G Mobile Communication Technologies*, June 2003, pp. 58–62.

[135] K. Valkealahti, J. P. A. Hoglund and, and A. Hamblainen, "Wcdma common pilot power control for load and coverage balancing," in *Proc. 13th IEEE Int. Symp. on Personal Indoor and Mobile Radio Communications*, vol. 4, Sep 2002, pp. 2244 – 2247.

[136] A. Hoglund and K. Valkealahti, "Quality-based tuning of cell downlink load target link power maxima in wcdma," in *Proc. 57th IEEE Vehicular Technology Conference*, vol. 4, Sep 2002, pp. 2248 – 2252.

[137] A. Hoglund, J. Pollonen, K. Valkealahti, and J. Laiho, "Quality-based auto-tuning of cell uplink load level targets in wcdma," in *Proc. 57th IEEE Vehicular Technology Conference*, vol. 4, April 2003, pp. 2847–2851.

[138] R. Nasri, Z. Altman, H. Dubreil, and Z. Nouir, "WCDMA downlink load sharing with dynamic control of soft handover parameters," in *Proc. 63rd IEEE Vehicular Technology Conference*, vol. 2, May 2006, pp. 942 – 946.

[139] P. Stuckmann, "The EUREKA GANDALF project: Monitoring and self-tuning techniques for heterogeneous radio access networks," in *Proc. 61th IEEE Vehicular Technology Conference*, vol. 4, May 2005, pp. 2238–2242.

[140] M. Toril, S. Pedraza, R. Ferrer, and V. Wille, "Optimization of handover margins in GSM/GPRS networks," in *Proc. 57th IEEE Vehicular Technology Conference*, vol. 1, May 2003, pp. 150–154.

[141] I. Jimenez, M. Toril, R. Toril, and O. Fernandez, “Análisis del rendimiento de un sistema gsm con distribución no homogénea de tráfico,” in *Proc. XIX Simposium Nacional de la Unión Científica Internacional de la Radio (URSI)*, Sep 2004.

[142] S. Pedraza and M. Toril, “Method and system for load sharing between a plurality of cells in a radio network system,” Patent application WO 02/104058, Dic 2002.

[143] ——, “Method for setting parameters of a cellular radio network, in particular for access and handover,” Patent application WO 03/037017, May 2003.

[144] ——, “Method and system for harmonizing an operation area for a mobile device in a cellular radio,” Patent application WO 03/037020, May 2003.

[145] K. Krishnan, “The convexity of loss rate in an erlang loss system and sojourn in an erlang delay system with respect to arrival and service rates,” *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1314 – 1316, Sep 1990.

[146] S. M. Stefanov, “Convex separable minimization subject to bounded variables,” *Computational Optimization and Applications*, vol. 18, no. 1, pp. 27–48, 2001.