

OPTIMIZATION AND PARALLELIZATION  
METHODS FOR THE DESIGN  
OF NEXT-GENERATION  
RADIO NETWORKS

Lucas Benedičič

**Doctoral Dissertation**

**Jožef Stefan International Postgraduate School**

**Ljubljana, Slovenia, September 2013**

**Supervisor:** Assist. Prof. Dr. Peter Korošec, Jožef Stefan Institute, Ljubljana, Slovenia

**Co-Supervisor:** Assist. Prof. Dr. Tomaž Javornik, Jožef Stefan Institute, Ljubljana, Slovenia

**Evaluation Board:**

Assist. Prof. Dr. Jurij Šilc, Chairman, Jožef Stefan Institute, Ljubljana, Slovenia

Assist. Prof. Dr. Aleš Šviglej, Member, Jožef Stefan Institute, Ljubljana, Slovenia

Prof. Dr. Marian Vajteršic, Member, Department of Computer Science, University of Salzburg, Austria

MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA  
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL



Lucas Benedičič

**OPTIMIZATION AND PARALLELIZATION  
METHODS FOR THE DESIGN  
OF NEXT-GENERATION  
RADIO NETWORKS**

Doctoral Dissertation

**OPTIMIZACIJSKE IN VZPOREDNE METODE ZA  
NAČRTOVANJE RADIJSKIH OMREŽIJ  
NASLEDNJE GENERACIJE**

Doktorska disertacija

*Supervisor:* Assist. Prof. Dr. Peter Korošec

*Co-Supervisor:* Assist. Prof. Dr. Tomaž Javornik

Ljubljana, Slovenia, September 2013



## Abstract



# Povzetek

Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek.  
Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek.  
Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek.  
Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek. Prvi odstavek.

Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek.  
Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi  
odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi  
odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi  
odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek. Drugi odstavek.



# Contents

<b>Abbreviations</b>	<b>xxi</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Problem statement . . . . .	2
1.2 Hypotheses and approaches . . . . .	3
1.3 Scientific contributions . . . . .	4
1.4 Organization . . . . .	5
1.5 Publications . . . . .	5
<b>2 Overview of optimization methods</b>	<b>7</b>
2.1 Gradient-based methods . . . . .	8
2.2 Linear and non-linear programming . . . . .	9
2.3 Metaheuristics . . . . .	10
2.3.1 Differential evolution . . . . .	12
2.3.2 Differential ant-stigmergy algorithm . . . . .	13
2.3.3 Simulated annealing . . . . .	14
2.4 Black-box optimization . . . . .	14
<b>3 Principles of mobile radio networks</b>	<b>17</b>
3.1 Handover . . . . .	17
3.1.1 Hard handover . . . . .	18
3.1.2 Soft handover . . . . .	18
3.2 Pilot signal and power . . . . .	19
<b>4 Overview of radio-network optimization</b>	<b>21</b>
4.1 Optimizing base-station locations . . . . .	22
4.1.1 Related work . . . . .	23
4.2 Optimizing antenna parameters . . . . .	23
4.2.1 Related work . . . . .	24
4.3 Optimizing coverage . . . . .	25
4.3.1 Related work . . . . .	25
4.4 Discussion . . . . .	26
4.5 Summary . . . . .	26
<b>5 Principles of GPU programming</b>	<b>27</b>
5.1 CUDA . . . . .	27
5.2 OpenCL . . . . .	29
5.3 Summary . . . . .	29
<b>6 A parallel framework for radio-network planning and optimization</b>	<b>31</b>
6.1 Motivation . . . . .	32

6.2	Related work . . . . .	33
6.3	Radio-coverage prediction for mobile networks . . . . .	34
6.3.1	Background . . . . .	34
6.3.2	Radio-propagation model . . . . .	34
6.4	Design and implementation . . . . .	35
6.4.1	Geographic Resources Analysis Support System . . . . .	35
6.4.2	Multi-paradigm parallel programming . . . . .	36
6.4.3	Design of the serial version . . . . .	36
6.4.3.1	Input parameters . . . . .	36
6.4.3.2	Isotropic path-loss calculation . . . . .	37
6.4.3.3	Antenna diagram influence . . . . .	37
6.4.3.4	Transmitter path-loss prediction . . . . .	38
6.4.3.5	Coverage prediction . . . . .	38
6.4.4	Computational complexity . . . . .	38
6.4.5	Design of the parallel version . . . . .	40
6.4.5.1	Master process . . . . .	41
6.4.5.2	Worker processes on CPU . . . . .	43
6.4.5.3	Worker processes on GPU . . . . .	44
6.4.5.4	Master-worker communication . . . . .	45
6.5	Simulations . . . . .	46
6.5.1	Test networks . . . . .	46
6.5.2	Weak scalability . . . . .	47
6.5.2.1	Results . . . . .	47
6.5.3	Strong scalability . . . . .	49
6.5.3.1	Results . . . . .	49
6.5.3.2	Speedup . . . . .	50
6.5.3.3	Efficiency . . . . .	51
6.5.4	GPU performance . . . . .	52
6.6	Summary . . . . .	53
<b>7</b>	<b>The service-coverage problem</b>	<b>55</b>
7.1	Motivation . . . . .	56
7.2	Related work . . . . .	56
7.3	Radio-network model . . . . .	57
7.3.1	Basic elements . . . . .	57
7.3.2	Coverage . . . . .	57
7.4	Problem definition . . . . .	58
7.5	Optimization approaches . . . . .	58
7.5.1	Attenuation-based approach . . . . .	58
7.5.2	Parallel-agent approach . . . . .	59
7.5.2.1	Objective-function evaluation . . . . .	59
7.5.2.2	Autonomous agents . . . . .	60
7.6	Simulations . . . . .	62
7.6.1	Test networks . . . . .	62
7.6.2	Parameter settings of the parallel-agent approach . . . . .	63
7.6.3	Experimental environment . . . . .	64
7.6.4	Results . . . . .	64
7.6.4.1	Performance analysis . . . . .	65
7.7	Summary . . . . .	68

<b>8 The soft-handover balancing problem</b>	<b>69</b>
8.1 Motivation . . . . .	70
8.2 Related work . . . . .	70
8.3 Radio-network model . . . . .	72
8.3.1 Soft-handover areas . . . . .	72
8.4 Problem definition . . . . .	73
8.5 Optimization approaches . . . . .	74
8.5.1 Differential evolution . . . . .	74
8.5.2 Differential ant-stigmergy algorithm . . . . .	75
8.5.3 Simulated annealing . . . . .	75
8.6 Simulations . . . . .	76
8.6.1 Test network . . . . .	76
8.6.2 Algorithm parameters . . . . .	76
8.6.3 Experimental environment . . . . .	78
8.6.4 Results . . . . .	78
8.6.4.1 Performance analysis . . . . .	79
8.7 Summary . . . . .	82
<b>9 Framework automated tuning</b>	<b>83</b>
9.1 Motivation . . . . .	83
9.2 Related work . . . . .	84
9.3 Parameter tuning of the radio-propagation model . . . . .	85
9.3.1 Field measurements . . . . .	85
9.3.2 Linear least squares . . . . .	86
9.3.3 Simulations . . . . .	87
9.3.3.1 Test networks . . . . .	87
9.3.3.2 Experimental environment . . . . .	88
9.3.4 Results . . . . .	88
9.4 Clutter optimization . . . . .	89
9.4.1 Optimization objective . . . . .	91
9.4.2 Differential ant-stigmergy algorithm . . . . .	91
9.4.3 Simulations . . . . .	91
9.4.4 Results . . . . .	92
9.5 Summary . . . . .	98
<b>10 Performance assessment within real network-planning scenarios</b>	<b>99</b>
10.1 Measurements and simulation comparison . . . . .	99
10.2 Coverage-prediction performance analysis . . . . .	99
10.2.0.1 Performance analysis . . . . .	100
10.3 Summary . . . . .	101
<b>11 Conclusion and further work</b>	<b>103</b>
11.1 Scientific contributions . . . . .	104
<b>12 Bibliography</b>	<b>107</b>
<b>13 Acknowledgments</b>	<b>121</b>



# List of Figures

Figure 1.1 A diagram of the classic decision-making process, adapted from [159]. Multiple iterations of this process improve the optimization algorithm and/or model until an acceptable solution is found.	1
Figure 2.1 Gradient descent with a negative slope, i.e. $x$ is increasing.	8
Figure 2.2 A saddle point or point of inflection, where the derivative is zero.	9
Figure 2.3 Graphical representation of a linear-programming example with two constraints, $c_1$ and $c_2$ . The greyed area is the polytope representing the region of feasible solutions.	10
Figure 2.4 A metaheuristic algorithm process using a black box for the objective-function evaluation, $f(x)$ , of a solution, $x$ .	15
Figure 3.1 Example of two UEs in soft handover with two BSs. Adapted from [219??].	19
Figure 4.1 A typical optimization cycle for radio networks. This sequence is repeated until the achieved results are acceptable.	21
Figure 4.2 A graphical representation of the set-covering problem: (a) the problem input and (b) the solution.	22
Figure 4.3 An example of an antenna-azimuth pattern.	24
Figure 4.4 The antenna tilt, showing its angle with the horizontal plane.	24
Figure 5.1 A simplified diagram of the architecture of a modern GPU. Adapted from [131].	28
Figure 6.1 Flow diagram of the serial version.	37
Figure 6.2 Example of a raster map, showing the result of a path-loss calculation from an isotropic source.	37
Figure 6.3 Example of a raster map, showing the antenna influence over the isotropic path-loss result, as depicted in Figure 6.2.	38
Figure 6.4 Example of a raster map, displaying the final coverage prediction of 136 transmitters over a geographical area. The color scale is given in dBm, indicating the received signal strength. Darker colors denote areas with a reduced signal due to the fading effect of the hilly terrain and clutter.	39
Figure 6.5 Example of a geographical-location mapping of the input-spatial data into a 2D matrix, the lower-left corner of which indicates the nearest point to the origin of the map-projection system.	41
Figure 6.6 Flow diagram of the master process.	42
Figure 6.7 Flow diagram of the “Processing loop” step of the master process.	42
Figure 6.8 Flow diagram of a worker process.	44
Figure 6.9 Communication diagram, showing the message passing between the master and a worker process.	44

Figure 6.10 Measured wall-clock time for weak-scalability experiments, featuring MW and MWD setups. Experiments allocated one MPI worker process per core. The wall-clock time axis is expressed in a base-10 logarithmic scale, whereas the axis representing the number of cores is expressed in a base-2 logarithmic scale. . . . .	48
Figure 6.11 Measured wall-clock time for strong-scalability experiments, featuring MW and MWD setups. Experiments assigned one MPI worker process per core. The wall-clock time axis is expressed in a base-10 logarithmic scale, whereas the axis representing the number of cores is expressed in a base-2 logarithmic scale. . . . .	50
Figure 6.12 Average speedup for strong-scalability experiments. Both axes are expressed in a base-2 logarithmic scale. . . . .	51
Figure 6.13 Average parallel efficiency for strong-scalability experiments. The parallel-efficiency axis is expressed in a linear scale, whereas the axis representing the number of cores is expressed in a base-2 logarithmic scale. . . . .	51
 Figure 7.1 Architecture of the parallel, agent-based optimization system on GPU.	59
Figure 7.2 Convergence profile of the parallel-agent approach for the test network Net <sub>1</sub> . . . . .	65
Figure 7.3 Convergence profile of the parallel-agent approach for the test network Net <sub>2</sub> . . . . .	66
Figure 7.4 Convergence profile of the parallel-agent approach for the test network Net <sub>3</sub> . . . . .	66
 Figure 8.1 HSUPA traffic and uplink interference with: (a) balanced downlink and uplink SHO conditions, and (b) unbalanced downlink and uplink SHO conditions. . . . .	71
Figure 8.2 Area under radio coverage, $A_{\text{covered}}$ , and without radio coverage, $\overline{A}_{\text{covered}}$ , within the complete geographical area, $A_{\text{total}}$ , of test network Net <sub>7</sub> . . . . .	77
Figure 8.3 Convergence analysis for each of the three algorithms, i.e., DE, DASA and SA, showing the best results obtained for the SHO-balancing problem. . . . .	79
Figure 8.4 Spatial distribution of the SHO areas before the optimization. . . . .	80
Figure 8.5 Spatial distribution of the SHO areas after the optimization. . . . .	81
 Figure 9.1 Terrain profile of test network Net <sub>8</sub> , dominated by a flat agricultural area. . . . .	88
Figure 9.2 Terrain profile of test network Net <sub>10</sub> , dominated by forested hills. . . . .	88
Figure 9.3 Error distribution of the radio prediction for network Net <sub>8</sub> : (a) with default parameter values, and (b) with fitted parameter values. . . . .	89
Figure 9.4 Error distribution of the radio prediction for network Net <sub>9</sub> : (a) with default parameter values, and (b) with fitted parameter values. . . . .	90
Figure 9.5 Error distribution of the radio prediction for network Net <sub>10</sub> : (a) with default parameter values, and (b) with fitted parameter values. . . . .	90
Figure 9.6 PRATO architecture and data flow during the clutter-optimization phase. The optimization algorithm runs on the master process, while the radio-propagation predictions of the involved network cells run in parallel over several worker processes. . . . .	92
Figure 9.7 Error distribution of the radio prediction for network Net <sub>8</sub> : (a) with fitted parameter values and default clutter losses, and (b) with fitted parameter values and optimized clutter losses. . . . .	94

Figure 9.8 Error distribution of the radio prediction for network Net <sub>9</sub> : (a) with fitted parameter values and default clutter losses, and (b) with fitted parameter values and optimized clutter losses. . . . .	94
Figure 9.9 Error distribution of the radio prediction for network Net <sub>10</sub> : (a) with fitted parameter values and default clutter losses, and (b) with fitted parameter values and optimized clutter losses. . . . .	95
Figure 9.10 Box plots representing the statistical-analysis values of Table 9.5, for the solutions of the clutter-optimization process of each test network: (a) Net <sub>8</sub> , (b) Net <sub>9</sub> , and (c) Net <sub>10</sub> . . . . .	96



# List of Tables

Table 5.1 Properties of different memory types on a GeForce 8800 GPU. Adapted from [131]. . . . .	29
Table 5.2 Naming-convention translation between OpenCL and CUDA. Adapted from [90]. . . . .	29
Table 6.1 Running-time gain (in percent) of the simulations for the weak-scalability of the MWD setup relative to the classic MW approach. . . . .	48
Table 7.1 Sizes of the test networks used for experimentation of the service-coverage problem, in terms of equipment and geographical area. . . . .	63
Table 7.2 Network parameters of the test networks used for the service-coverage problem. . . . .	63
Table 7.3 Parameter settings of the parallel-agent approach for each test network. . . . .	64
Table 7.4 Optimization results after applying two different approaches for solving the service-coverage problem. All values are expressed in Watts. . . . .	64
Table 7.5 Wall-clock times (in seconds) and speed-up factors for the different implementations of the objective-function evaluation and the parallel agents, during the experimentation of the service-coverage problem. . . . .	67
Table 8.1 Technical characteristics of Net <sub>7</sub> , the test network used for the SHO-balancing problem. . . . .	76
Table 8.2 Solution-quality performance of the three algorithms, after 30 independent runs. . . . .	78
Table 8.3 Improvement analysis of the best solution that each algorithm achieved for the SHO-balancing problem. . . . .	80
Table 9.1 Clutter-category label numbers and descriptions for the signal-loss due to clutter of the radio-propagation model. . . . .	86
Table 9.2 Several properties of Net <sub>8</sub> , Net <sub>9</sub> and Net <sub>10</sub> , the test networks used for the experimental simulation of the framework automated tuning. . . . .	87
Table 9.3 Clutter-category proportions, expressed in percent, in terms of the surface area of each of the test networks. The category legend is given in Table 9.1. . . . .	88
Table 9.4 Clutter-category losses after the optimization. The default losses for each clutter category are given along the solutions for each of the test networks. All values are expressed in dB. . . . .	93
Table 9.5 Statistical analysis of the optimization solutions for each test network. All values are expressed in dB. The corresponding box plots are depicted in Figure 9.10. . . . .	97

Table 10.1 Measured and estimated running times of the clutter-optimization process using the adapted serial implementation. Times are depicted in seconds for the three test networks. . . . .	101
Table 10.2 Measured running times (in seconds) of the clutter-optimization process using PRATO over multiple processors. The number of deployed worker processes and speed-up factors are also depicted for the three test networks.	102

# List of Algorithms

2.1	Pseudo-code of the simulated-annealing metaheuristic. Adapted from [23]. . .	14
6.1	Pseudo-code of the radio-coverage prediction algorithm. The time complexity is given per line. . . . .	39
7.1	Pseudo-code representing the behaviour of an agent. . . . .	61
7.2	Pseudo-code representing the step set $SS_0$ , which is applied by the agents in areas with no service coverage. . . . .	61
7.3	Pseudo-code representing the step set $SS_1$ , which is applied by the agents in areas with service coverage. . . . .	61
8.1	A move in the search space of SA for solving the SHO-balancing problem. . .	75



# Abbreviations

## Abbreviations

2G	= Second Generation of mobile networks.
3G	= Third Generation of mobile networks.
4G	= Fourth Generation of mobile networks.
ACO	= Ant-colony optimization.
BSs	= Base station.
CDMA	= Code division multiple access.
CUDA	= Compute unified device architecture.
DASA	= Differential ant-stigmergy algorithm.
DB	= Database system.
DE	= Differential evolution.
GPU	= Graphics processing unit.
GRASS	= Geographic Resources Analysis Support System.
GSM	= Global System for Mobile communications.
HPC	= High-perfomance computing.
HPC	= High-performance computing.
HSDPA	= High Speed Downlink Packet Access.
HSPA	= High Speed Packet Access.
HSUPA	= High Speed Uplink Packet Access.
I/O	= Input/output.
KPI	= Key performance indicator.
LTE	= Long Term Evolution.
MW	= Master-worker parallel paradigm.
MWD	= Master-worker-database parallel paradigm.
PRATO	= Parallel radio-prediction tool.

QoS	= Quality of service.
RSG	= Regular square grid.
SA	= Simulated annealing.
SHO	= Soft handover.
SIMT	= Single instruction multiple thread.
SM	= Streaming multiprocessor.
SP	= Streaming processor.
SPMD	= Single-program multiple-data.
TDMA	= Time division multiple access.
UE	= User equipment or mobile device.
UE	= User equipment.
UMTS	= Universal Mobicel Telecommunications System.
WCDMA	= Wide-code division multiple access.

## Symbols

$\gamma^{\text{cov}}$	= Signal-to-interference ratio (SIR) coverage threshold.
$\gamma^{\text{sho}}$	= SHO window.
$A_{\text{covered}}$	= Area under service coverage of the mobile network.
$A_{\text{total}}$	= Complete geographical area under optimization.
$as^{\max}$	= Maximum number of neighbouring cells in the active set.
$C$	= Set of antenna installations (cells) in a mobile network.
$C_m$	= Subset of cells, $C_m \subset C$ , that cover a mobile $m \in M$ .
$c_m^*$	= Best-serving cell of mobile $m \in M$ .
$cov(c, m)$	= Binary function to assert the coverage of a mobile $m \in M$ by from a cell $c \in C$ .
$cov(x, y)$	= Returns 1 if the coordinate $(x, y)$ is under mobile-network coverage.
$f_{\text{cov}}$	= Objective function for the service-coverage optimization problem.
$f_{\text{sho}}$	= Objective function of the SHO-balancing problem.
$L_{cm}^{\downarrow}$	= Downlink attenuation factor between cell $c \in C$ and mobile $m \in M$ .
$L_{mc}^{\uparrow}$	= Uplink attenuation factor between mobile $m \in M$ and cell $c \in C$ .
$M$	= Set of mobile devices or users of a mobile network.
$P_c$	= Set of candidate CPICH power settings for cell $c \in C$ .

$p_c$	= Pilot-power setting of cell $c \in C$ .
$p_m^\uparrow$	= Uplink transmit power of mobile $m \in M$ .
$SHO_m^\downarrow$	= Cell set to which a mobile may maintain concurrent downlink connections, i.e. downlink SHO.
$SHO_m^\uparrow$	= Cell set to which a mobile may maintain concurrent uplink connections, i.e. uplink SHO.
$sir(c, m)$	= Signal-to-interference ratio (SIR) from cell $c \in C$ to mobile $m \in M$ .



# 1 Introduction

Many researchers believe the computer has become the third method to do research, behind theory and experimentation, for both science and engineering. Although there is no complete agreement on the position intended for scientific computing with respect to the other two methods, it is undeniable that computational methods are an essential tool in most disciplines, particularly in those related to decision making.

Nowadays, decision making is present practically everywhere. As scientists, engineers and managers have to make decisions in more complex and competitive circumstances every day, decision making involves dealing with rational and optimal approaches. According to Talbi [159], decision making consists of the following steps:

- formulating the problem,
- modeling the problem,
- optimizing the problem, and
- implementing a solution for the problem.

Formulating a decision problem means making an initial statement about it. Although this first formulation may be imprecise, the objectives of the problem are outlined, together with the internal and external factors that have some degree of influence over it. During the modeling of the problem, an abstract mathematical model is built for it. Sometimes this model is inspired by similar models in the literature, making it possible to tackle the problem with well-studied methods. After a model of the problem is available, the optimization step, i.e., generating “good” solutions for the problem, may begin. It is important to note that the resulting solutions are given for the abstract model, and not for the original problem itself. Therefore, the performance of the obtained solution is indicative when the model is an accurate one [159]. In the last step, the obtained solution is practically tested by the decision maker and implemented if it is an “acceptable” or “good” one. In case of “bad” or “unacceptable” solutions, the decision-making process is repeated, possibly improving the model and/or the optimization algorithm. The process, as described here, is depicted in Figure 1.1.

Scientific computing, by means of computer-science methodology, enables the study of problems that are too complex to be treated analytically, or those that are very expensive

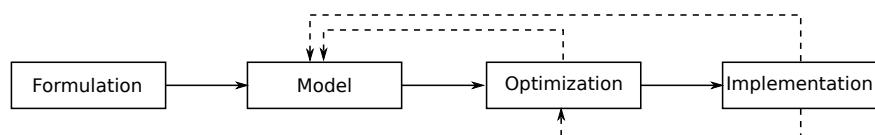


Figure 1.1: A diagram of the classic decision-making process, adapted from [159]. Multiple iterations of this process improve the optimization algorithm and/or model until an acceptable solution is found.

or dangerous to be studied by direct experimentation. Real-world problems are typically very complex systems to be directly assessed by analytical models, and require a numerical simulation for their study. Computer simulations provide a resource to mimic the behavior of complex systems, by numerically evaluating a model and gathering its data to estimate their true characteristics [96].

A model is a simplified representation of a studied problem, and one of its purposes is to predict the effects of the variations within the system. A good model is a balance between realism and simplicity. The system simulation, on the other hand, is the operation of the model. Its configuration can be changed, thus allowing multiple experimental executions, something that might not be possible with the real system it represents [103]. However, it is important to understand that the models used in scientific simulations and engineering never offer a perfect representation of the system they resemble, but only a subset of its composition and dynamics. For this reason, experimentation and expert observation will always be essential as reference points for understanding the studied phenomena. Consequently, problems categorized as of large size and of considerable complexity represent a challenge, because of the different involved disciplines for their study and the degree of difficulty of their modeling. Cellular radio networks in general, and those of the lastest generations in particular, fall under this categorization.

## 1.1 Problem statement

Radio networks represent one of the most fast-growing technology markets since the introduction of the Global System for Mobile communications (GSM) [1]. As an implementation of the second generation of mobile networks (2G), GSM appeared more than twenty years ago. Its successor, the Universal Mobile Telecommunications System (UMTS) [2] marks an evolution from 2G, representing a milestone for the third generation of mobile radio networks (3G). In recent years, the first commercial networks implementing the Long Term Evolution (LTE), also known as fourth generation (4G) LTE, have also appeared [61]. The always increasing demand for more bandwidth has been one of the main forces behind the standardization and later implementation of systems delivering higher-speed data services in order to improve the user's experience.

This evolution, first from 2G to 3G and later from 3G to 4G, has introduced not only the technology needed to increase data-transfer capacity and voice quality, but also a greater complexity in terms of radio-network planning, deployment, and configuration. This fact has attracted the attention of the research community into areas such as the design and optimization of radio-networks.

During the design phase of a radio network, a traditional or manual approach comprises the software tool, i.e., the model in Figure 1.1, that executes the analysis, and the human that makes the configuration changes, i.e., the optimization in Figure 1.1. Therefore, a radio engineer manually adjusts the network parameters and the software tool analyzes the given configuration. If the obtained results are not acceptable, the analysis process has to be repeated several times, until the goal is achieved and the changes are implemented, i.e., the implementation in Figure 1.1. In the context of this thesis, this process is referred to as manual radio-network optimization.

Advances in the past few years have improved the manual optimization process by introducing different problem-solving approaches that increase the role of the computer during the optimization of radio networks, consequently enlarging the scope of problems and instance sizes that may be subject to optimization. Still, there are some important aspects that restrict the automation level of these methods, not only in real-world environments, but also when doing research in the area of radio networks:

- A selected optimization method is typically a compromise between solution quality and computational-time complexity. The proposed state-of-the-art approach for the evaluation of radio networks is the Monte-Carlo snapshot analysis [165, 172]. However, real-world environments, where radio-network design is carried out, require the evaluation of networks with thousands of base stations in a reasonable amount of time. Moreover, for applications involving radio-network optimization, usually millions of evaluations are required to find a good solution, in which case also snapshot simulations are too time-consuming for practical use. Therefore, for such applications and environments, methods with improved time efficiency are required.
- A considerable number of publications in the field of radio-network optimization, a few references of which are given for illustration purposes [8, 29, 32, 66, 143, 146], base their simulations on platforms for which it is not possible to reproduce the experiments, either because they have used proprietary software or because the data is not available. This fact reduces the possibilities for comparing different approaches among each other, and significantly contrasts with other research areas, such as evolutionary computing or numerical optimization, where several sets of open and well-known benchmarks are available for the community to use. Consequently, an open and unified framework should allow researchers to compare different methods and results in a simplified and objective manner.
- Available commercial tools for radio-network evaluation present several drawbacks. In particular, regarding to the size of networks that can be considered and their computational-time complexity. Yet, even if precise and fast methods were commercially available, they would lack the level of flexibility required by the scientific community. Consequently, an essential attribute of the framework is to be open source, so that anyone can extend it to meet some specific requirements. In the long term, this process should also extend the set of built-in functionality.
- Particularly for path-loss predictions, estimated with empirical or deterministic mathematical models, the inaccuracy of the input data directly deteriorates the precision of the calculated results. Moreover, since the physical properties that influence the propagation of radio signals are not constant and every environment introduces its own deviations, the calculated prediction may be considered as not more than a rough approximation. Therefore, there is a need for a technique that improves the accuracy of the state-of-the-art mathematical models for radio propagations, despite the various sources of error noted before.

This thesis introduces methods and tools to mitigate the above-mentioned drawbacks from a radio-planning perspective.

## 1.2 Hypotheses and approaches

The work presented in this thesis is based on the following hypotheses and their related approaches:

- Applying parallelization techniques should reduce the computational-time requirement of the radio-propagation prediction, thus making it possible to process larger, real-world radio networks.
  - Unlike several examples in the literature of radio-network simulators, which do not include support for parallel methods in order to reduce the computational-time complexity, the unified framework includes this feature in its basic structure.

In this sense, parallel techniques are used even when targeting a single computing host.

- Using hardware, specialized for parallel execution (e.g., GPU), should improve the execution time of parallel algorithms using threads or message-passing mechanisms.
  - From its arrival a few years ago, general GPU programming is continuously gaining the attention of the research community and it is extending to several areas of science. In particular, the combination of parallel-programming techniques with GPU hardware should reduce the computational-time requirement of the radio-propagation prediction even further.
- Using the parallel framework for the objective-function evaluation of different optimization problems should improve the quality of solutions and enable tackling larger problem instances.
  - On the one hand, a larger number of evaluations in the same amount of time translates in a better search of the search space by the optimization algorithm used. On the other hand, more complex or exact models may be evaluated in the same amount of time, which implies more accurate solutions.
- Using the parallel framework to evaluate the objective function of an automatic-optimization system should provide the means for solving new, previously inaccessible optimization problems.
  - The improved performance at the evaluation level opens new possibilities for formally defining and tackling optimization problems that were previously out-of-reach of state-of-the-art methods, either because of their complexity or size.

### 1.3 Scientific contributions

The contributions of this thesis to the fields of telecommunications and computer sciences include the following:

- Design and development of an unified framework for radio-network planning and optimization, incorporating the following two elements, as well as experimentation on real-world radio networks. The experimentation includes a comparison with a commercial tool that is currently being used for real radio-network planning. Moreover, the parallel implementation of the framework exploits the computing resources of computer clusters and GPUs.
- Proposal of a new approach for parallel programming that combines a classic master-worker method with an external database. The proposed approach improves the scalability of the classic master-worker paradigm by preventing the master process to become the bottleneck of a parallel system.
- Quality improvement of renowned radio-propagation models by applying metaheuristic-optimization techniques that enhance the results of current state-of-the-art methods.
- A new algorithm, based on autonomous agents, to tackle the service-coverage problem in radio networks. The algorithm deals with problem instances that are out-of-reach of other state-of-the-art techniques, reaching good quality of solutions if compared to

classic network-planning techniques. This approach is applicable for large real-world radio networks.

- Identification of a new optimization problem in 3G radio networks that deals with soft-handover alignment of downlink and uplink areas. By solving this problem, network malfunctioning is avoided in areas where there is soft-handover capability in the uplink, but none in the downlink. So far this problem has never been solved by means of automatic optimization.

## 1.4 Organization

So far, a context has been provided for the following content of this thesis. An introduction of the three research areas addressed by this work, i.e., optimization, radio networks, and parallel methods, is given in Chapters 2, 3 and 5, respectively. They provide an overview of the theoretical elements needed for a better understanding of the rest of the thesis. Additionally, Chapter 4 provides a short literature survey about the optimization of radio networks.

The unified framework for radio-network planning and optimization, incorporating elements from the above-mentioned fields of study, is presented in Chapter 6.

Chapters 7 and 8 demonstrate how the framework is applied to tackle two optimization problems. In Chapter 7, the problem of minimizing the total amount of pilot power subject to a full coverage constraint is addressed with a novel approach. Next, Chapter 8 formally introduces a new optimization problem that deals with the balancing of downlink and uplink soft-handover areas in UMTS networks.

The applicability and performance of the framework for radio-network planning is validated in Chapters 9 and 10. Chapter 9 further extends the framework with automated-tuning capabilities that enable its adaptation to different environments, thus improving the accuracy of the radio-propagation predictions. In Chapter 10, the framework is tested in real network-planning conditions and compared to a commercial, enterprise-level tool in terms of solution quality and speed performance.

Finally, Chapter 11 gives a short summary of the thesis, outlines its main contributions and discusses further work.

## 1.5 Publications

The work presented in this thesis is supported by a number of previous publications ...???

A comprehensive list of related publications is given in Appendix ???.



## 2 Overview of optimization methods

Optimization may be informally defined as the procedure of finding better solutions to a given problem that usually models some physical phenomenon. In our every day life, we are constantly solving small optimization problems, like choosing the shortest route to a friend's house, or organizing the appointments in our agenda. In general, these problems are small enough for us to find a good solution without extra help, but as they become larger and more complex, the aid of computers for their resolution is unavoidable.

Complex multidimensional optimization problems are popular in engineering, economics, physics and other scientific fields. When solving an optimization problem, the objective is to find a "good" solution in a "reasonable" computational time. In this respect, the field of mathematical optimization has received a lot of attention by the scientific community during the last decades. However, both "good" and "reasonable" are problem, application and context-specific concepts, in which the biggest challenge of selecting an appropriate optimization approach usually lays.

Mathematical optimization involves the process of finding solutions from a group of possible decisions, which may be defined as:

$$\min f(\vec{x}) \quad \vec{x} \in \Omega \subseteq \mathbb{R}^n, \quad (2.1)$$

where  $\vec{x} = (x_1, \dots, x_n)$  is a vector representing the decision variables,  $f(\vec{x})$  is the objective function measuring the quality of the decisions and  $\Omega$  is the set of feasible solutions of the problem, also known as search space. Note that the objective function  $f$  makes it possible to define a total order relation between any pair of solutions in  $\Omega$ .

The search space  $\Omega$  may also be expressed as a solution to a system of equalities or inequalities, e.g.:

$$\begin{aligned} g(x_1, \dots, x_n) &\leq 0 \\ h(x_1, \dots, x_n) &= 0. \end{aligned} \quad (2.2)$$

Optimization problems involving the maximization of the objective function also fall into this category, since:

$$\max f(\vec{x}) = -\min(-f(\vec{x})). \quad (2.3)$$

A point  $\vec{x}^*$  is considered to be an unrestricted local minimum of a function  $f$  if it holds a better value than all its neighbours, i.e., there exists  $\epsilon > 0$  so that:

$$f(\vec{x}^*) \leq f(\vec{x}) \quad \forall \vec{x} \in \mathbb{R}^n \quad |\vec{x} - \vec{x}^*| < \epsilon. \quad (2.4)$$

Similarly, a point  $\vec{x}^*$  is considered to be an unrestricted global minimum of a function  $f$  if it holds a better value than all others, i.e.:

$$f(\vec{x}^*) \leq f(\vec{x}) \quad \forall \vec{x} \in \mathbb{R}^n. \quad (2.5)$$

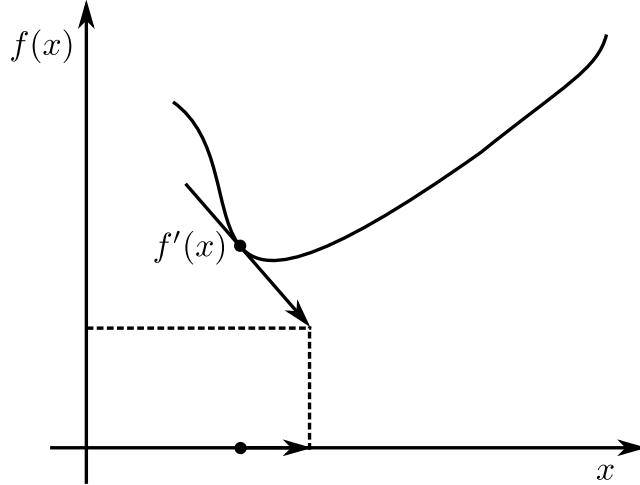


Figure 2.1: Gradient descent with a negative slope, i.e.  $x$  is increasing.

The concepts of local and global minimum are considered strict if the inequalities of Equations 2.4 and 2.5 are strict. Likewise, the definition of local and global maximum is given by the existing relation between a minimization and a maximization problem, as specified in Equation 2.3, i.e., a point  $\vec{x}^*$  is a local or global maximum of a function  $f$  if and only if  $\vec{x}^*$  is a local or global minimum of function  $-f$ , respectively.

## 2.1 Gradient-based methods

Gradient-based methods are among the oldest and most studied optimization approaches. They are based on the derivative of the optimized function, using the first and even the second derivate of a function  $f$ . The name gradient follows from the derivative of multidimensional functions,  $\nabla f(\vec{x})$ , which is simply a vector where each element is the slope of  $\vec{x}$  in that dimension, i.e.,  $\langle \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \rangle$  [101].

The principle behind gradient-based methods is rather simple. Starting from an arbitrary value for  $x$ , a subtraction (or addition) of a small positive value is iteratively applied to it, e.g., for gradient descent:

$$x \leftarrow x - \alpha f'(x), \quad (2.6)$$

where  $\alpha$  is a small positive value. Consequently, a positive slope will make  $x$  decrease, whereas a negative slope will make it increase. Figure 2.1 shows an example of this behaviour. Therefore,  $x$  will gradually move down the function until it finds its minimum, where  $f'(x)$  is zero, causing it to stop.

However, gradient methods have certain drawbacks that make them unsuitable for tackling a wide range of optimization problems. Take, for example, the time they take to converge. As gradient descent approaches a function minimum, it will skip this point and land on the other side. In the next step, something similar will happen, but this time from the other side of the minimum point, thus slowly approaching to the target in a “zig-zag” way. This behavior is directly related to the slope of the function at the given point, i.e., a steepest slope translates into a larger jump, and may be alleviated by adjusting the value of  $\alpha$ . However, some functions (or regions of functions) may require smaller values, while for

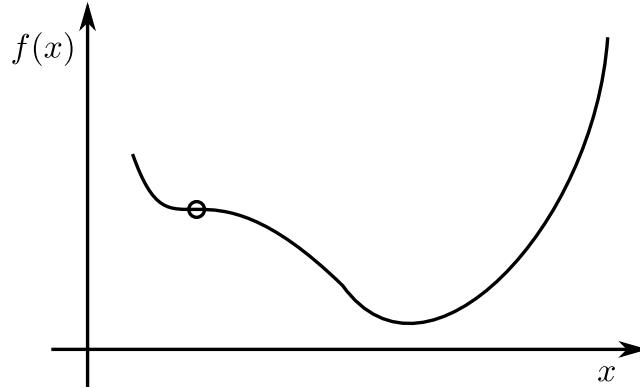


Figure 2.2: A saddle point or point of inflection, where the derivative is zero.

others a bigger value would be more appropriate. Newton's method improves this by taking the second derivative of the function into account, i.e.:

$$x \leftarrow x - \alpha \frac{f'(x)}{f''(x)}, \quad (2.7)$$

thus adjusting the value of  $\alpha$  as it converges towards a point with zero slope [101].

Another issue is how other points are handled. Beside maxima and minima points, some functions also contain saddle points (known as inflection points in one-dimensional functions). Clearly, the first derivative of a saddle point is zero, meaning gradient descent will stop looking for the minimum, even though it hasn't found it (see Figure 2.2). Newton's method, on the other hand, does not help either, even trying to divide by zero in this case. These observations clearly show how gradient methods get caught in local optima. Local optima of a function is defined as the optima (or minima in this case) of a local region. Similarly, global optima are defined as the optima of the whole domain of a function. It follows that gradient methods, as gradient descent or Newton's method, are local optimization algorithms [101].

But maybe the biggest concern with gradient-based methods is they assume the function under optimization is derivable. This assumption holds only when optimizing a well-formed mathematical function. Unfortunately, this is generally not true, since in most cases the gradient is not computable because the function is not known. The only available approach in such situations is creating inputs to the function in order to assess their quality. Meta-heuristics (see Section 2.3) are good candidates for this class of problems, for solving both moderate and large instances.

## 2.2 Linear and non-linear programming

It was in the early 40s of the twentieth century, through the work of teams formed by mathematicians, economists and physicists, that the basis were established for the resolution of problems with a set of techniques known as linear and non-linear programming. Their initial goal was to solve different kinds of logistic problems during the second world war.

In a linear programming optimization problem, both the objective function  $f$  and a given set of constraints are linear functions. The constraints impose restrictions over  $\vec{x}$ , i.e., they must meet certain requirements as, for example, fullfil a limited availability of resources. A problem may be formulated as follows:

$$\min f(\vec{x}) = c \cdot \vec{x} \quad (2.8)$$

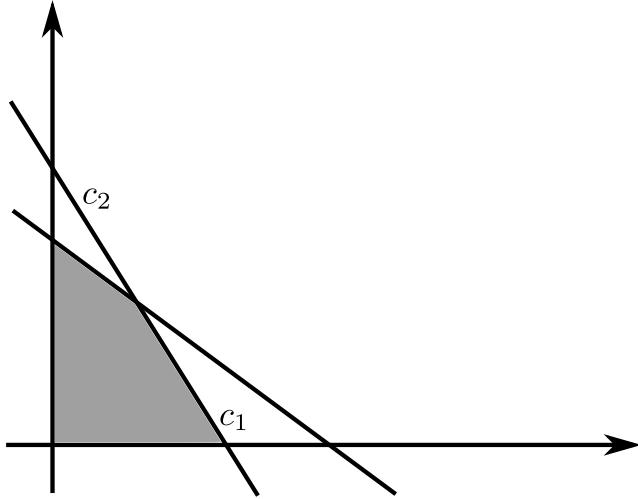


Figure 2.3: Graphical representation of a linear-programming example with two constraints,  $c_1$  and  $c_2$ . The greyed area is the polytope representing the region of feasible solutions.

subject to

$$\begin{aligned} A \cdot \vec{x} &\leq b \\ \vec{x} &\geq \vec{0} \end{aligned} \tag{2.9}$$

In the example above, the inequalities defined in 2.9 are the constraints to the linear program defined in 2.8.

For solving continuous, linear-optimization problems, efficient exact algorithms exist, such as the simplex method [43] or the interior-points method [88]. Indeed, linear programming is one of the most satisfactory models for solving optimization problems, since the feasible region of the problem is a convex set and the objective function is a convex function. It follows that the global optimum is a node of the polytope representing the feasible region [159]. See Figure 2.3 for a linear-programming example with several constraints.

Non-linear programming models, on the other hand, consider problems where the objective function  $f$  and/or the constraints are non-linear [15]. However, non-linear continuous problems are more difficult to solve. Despite several existing techniques to linearize such models, they often not only introduce extra variables and constraints, but also some degree of approximation [63]. Moreover, some problem properties such as high dimensionality, parameter interaction, and multi-modality make these approaches ineffective.

Generally speaking, when dealing with real-world problems, the availability of analytical optimization models, such as those required by gradient methods or (non-)linear programming, is not guaranteed. Indeed, for some applications, only simulations or physical models are the available means for objective-function evaluation [53]. Once again, metaheuristics appear as good candidates to solve different instance sizes of this class of problems.

## 2.3 Metaheuristics

Metaheuristics, a term proposed by Glover in [64], represent a group of approximation algorithms designed to combine basic, heuristic principles with advanced high-level guidance methods, targeted at improving the efficiency of a search process. These techniques are

meant to find good solutions to a given problem, for which the mathematical function is not available or its search space is big enough for an exhaustive search to be unfeasible [91].

From the theoretical point of view, metaheuristics represent a subset of stochastic optimization, since they use some degree of randomness to find optimal (or as good as possible) solutions to hard problems. They are the most general of these kinds of algorithms, and are applied to a wide range of problems [101]. Unlike the exact optimization methods introduced in the previous sections, metaheuristics do not guarantee the optimality of the obtained solutions [159]. Moreover, they do not define how close the obtained solutions are from the optimal ones, as approximation algorithms do.

The characterization given by Blum and Roli [25] provides a clear overview of the fundamental properties associated with metaheuristics:

- metaheuristics are strategies that “guide” the search process;
- their goal is to efficiently explore the search space in order to find optimal or near-optimal solutions;
- they build upon techniques which range from simple local search procedures to complex learning processes;
- they are approximate and usually non-deterministic;
- they may incorporate mechanisms to avoid getting trapped in confined areas of the search space;
- their basic concepts permit an abstract-level description, which is not problem-specific;
- they may make use of domain-specific knowledge in the form of heuristics that are controlled by the upper level strategy;
- advanced metaheuristics use search experience (implemented as some form of memory) to guide the search process.

The strategies used by metaheuristics should provide a dynamic balance between the exploitation of the accumulated search experience (commonly called intensification) and the exploration of the search space (commonly called diversification) [25]. This balance provides the necessary means to quickly identify promising regions, and early discarding those which have already been explored or don't provide solutions of better quality. Promising regions within the search space, which are identified by the obtained “good” solutions, are thoroughly explored during the intensification phase, hoping to find better solutions. On the other hand, during the diversification phase, not-yet-visited regions are explored, making sure the search space as a whole is evenly explored, thus avoiding confining the search to a reduced number of regions. In this context, the ultimate search algorithm in terms of diversification is random search. Random search generates a random solution in the search space at each iteration, without using memory [159]. In terms of intensification, iterative local search is a representative algorithm. The steepest local search algorithm selects, at each iteration, the best neighboring solution that improves the current one [159].

Metaheuristics are applicable where state-of-the-art exact algorithms cannot tackle the given instances within the required time, either because of their size or structure. The meaning of “required time” within this context directly depends on the target optimization problem itself. A feasible or acceptable time may vary from some seconds to several months, again, depending on the target optimization problem, e.g., real-time decisions against structural-design problems.

Based on the characterization given by Talbi [160], a summary of the essential properties of optimization problems that justify the use of metaheuristics follows:

- Very large problem instances. Even though exact polynomial-time algorithms might be known for solving the target problem, they are too expensive due to the size of the instances.
- Problems with hard real-time constraints, where a “good solution” has to be found online. Metaheuristics appear as an alternative to exact algorithms in order to reduce the search time.
- A difficult problem of moderate size, which input instances have an intricate structure.
- Optimization problems with time-consuming objective function(s) and/or constraints. Indeed, various real-world optimization problems are characterized by the huge computational cost of the objective functions. Several radio-network design problems fall into this category.
- Problems that cannot be solved with exhaustive search due to the non-analytical models on which they are based. These problems are defined by a black-box evaluation of the objective function (see Section 2.4).

The influence of these conditions may increase in the presence of non-deterministic optimization models, e.g., problems with complex Monte Carlo simulations [39].

Undoubtedly, metaheuristics are rapidly gaining popularity as optimization problems are increasing in both size and complexity. Indeed, as the computing power of commodity hardware increases, the possibility of building models of greater complexity is available for developing more accurate models of real-world problems in engineering and science.

Related literature groups metaheuristic algorithms due to their behaviour, for example:

- differential evolution, from the family of evolutionary algorithms;
- differential ant-stigmergy algorithm, from the family of swarm-intelligence algorithms; and
- simulated annealing, from the group of classic metaheuristic algorithms.

Each of these algorithms minimize an objective-function value by adopting essentially distinct approaches. In the following sections, a short description about their functioning is given.

### 2.3.1 Differential evolution

Differential Evolution (DE) [153] is a simple and powerful evolutionary algorithm proposed for numerical optimization. The version of DE used for this thesis is known as DE/rand/1/bin, or "classic DE" [125]. At the beginning, the classic DE initializes a population of  $N$ ,  $D$ -dimensional vectors with parameter values that are distributed with random uniformity between pre-specified lower and upper initial parameter bounds,  $x_{i,\text{low}}$  and  $x_{i,\text{high}}$ , respectively [130], i.e.:

$$x_{i,j,g} = x_{i,\text{low}} + \text{uniform}[0, 1] \cdot (x_{i,\text{high}} - x_{i,\text{low}}), \\ i = (1, 2, \dots, D), j = (1, 2, \dots, N), g = 0. \quad (2.10)$$

In Equation 2.10,  $g$  denotes the generation index, while  $i$  and  $j$  indicate the parameter and population indices, respectively. Consequently,  $x_{i,j,g}$  is the  $i^{\text{th}}$  parameter of the  $j^{\text{th}}$  population vector in generation  $g$ . The (pseudo)-random number generator uniform[0, 1) is represented by a function for selecting a uniformly-distributed random number from the interval [0, 1).

DE generates solutions by applying the mutation and cross-over operations. A vector from the current population is mutated by adding the scaled difference of two other vectors from the same population, combined with the mutation-scale factor. A proportion of the parameters of the mutated vector are crossed with those belonging to the  $j^{\text{th}}$  population vector, resulting in a new solution vector. The cross-over constant controls the proportion of parameters that the mutant vector contributes to the new vector.

The generated solution vector is retained if it yields a lower objective-function value than the  $j^{\text{th}}$  population vector, otherwise the  $j^{\text{th}}$  vector is kept for at least one more generation.

The four parameters to control the search process of DE are: the population size ( $NP$ ), the number of generations for the algorithm to run ( $G_{\max}$ ), the crossover constant ( $CR$ ), and the mutation scaling factor ( $F$ ).

An extensive description of DE and its variants may be found in [126]. A wide range of optimization problems have been solved by applying DE [44], also in the area of radio-network optimization [19, 42, 107].

### 2.3.2 Differential ant-stigmergy algorithm

The Differential Ant-Stigmergy Algorithm (DASA) [93] is based on the metaheuristic Ant-Colony Optimization (ACO) [46]. ACO was pioneered by Dorigo in 1992 and is based on the foraging behaviour of social ants, which use pheromone as a chemical messenger. When looking for a food source, a swarm of ants interacts in their local environment. An ant is able to follow the route marked with pheromone laid by other ants. When an ant finds a food source, it will mark the trail to and from it with pheromone.

The pheromone concentration,  $p$ , evaporates at a constant rate,  $\gamma$ , over time,  $t$ , i.e.:

$$p(t) = p_0 e^{-\gamma t}, \quad (2.11)$$

where  $p_0$  is the initial pheromone concentration at  $t = 0$ . The evaporation is important, as it provides the means for algorithm convergence and self-organization.

The pheromone concentration varies over time and the ants follow the route with higher concentration. In turn, the pheromone concentration increases with the number of ants. That is, as more and more ants follow the same route, it becomes the favored path, producing a positive feedback mechanism. Consequently, as the system evolves in time, it converges to a self-organized state, which is the essence of any ant-based algorithm.

As it was mentioned before, DASA is an ant-based algorithm that provides a framework to successfully cope with high-dimensional numerical optimization problems. It creates a fine-grained discrete form of the search space, representing it as a graph. This graph is then used as the walking paths for the ants, which iteratively improve the temporary best solution.

At the end of every iteration, and after all the ants have created solutions, they are evaluated to establish if any of them is better than the best solution found so far.

There are six parameters that control the way DASA explores the search space: the number of ants ( $m$ ), the discrete base ( $b$ ), the pheromone dispersion factor ( $q$ ), the global scale-increasing factor ( $s_+$ ), the global scale-decreasing factor ( $s_-$ ), and the maximum parameter precision ( $e$ ).

---

**Algorithm 2.1** Pseudo-code of the simulated-annealing metaheuristic. Adapted from [23].

---

**Require:**  $x \leftarrow$  initial solution  
**Require:**  $T_1 \leftarrow$  temperature parameter  
**for**  $k=1,2,\dots$  **do**  
     $y \leftarrow$  random solution from  $\Omega$   
    **if**  $f(y) \leq f(x)$  **then**  
         $x \leftarrow y$   
    **else if**  $e^{(\frac{f(x)-f(y)}{T_k})} \leq \text{uniform}[0, 1]$  **then**  
         $x \leftarrow y$   
    **end if**  
     $T_{k+1} \leftarrow$  update temperature  $T_k$   
**end for**

---

A more in-depth explanation about these parameters and the DASA algorithm itself can be found in [93].

### 2.3.3 Simulated annealing

This metaheuristic was introduced in the area of combinatorial optimization by Kirkpatrick et al. [89]. The algorithm resembles the physical annealing process, where particles of a solid material arrange themselves into a thermal equilibrium. Simulated Annealing (SA) is based on the principle of local-search heuristics, and uses a pre-defined neighborhood structure on the search space  $\Omega$ . The temperature parameter controls the search behavior. Its name follows the analogy to the physical annealing process. In each iteration, a solution  $y$ , which is a neighbour of the current solution  $x$ , is computed. If  $y$  has a better objective-function value than  $x$ , it becomes the new solution, i.e., the current solution  $x$  is replaced by  $y$ . If, on the other hand,  $y$  has a worse objective-function value than  $x$ , it is only accepted with a certain probability depending on two factors:

- the difference of the objective-function values of  $x$  and  $y$ , and
- the temperature parameter.

Intensification in SA is provided by the local-search nature of the algorithm, while diversification is produced with non-zero temperatures, by “uphill” moves. The pseudo-code of the SA metaheuristic can be represented as in Algorithm 2.1. In this pseudo-code,  $T_1, T_2, \dots$  represent an usually decreasing sequence of values for the temperature parameter, and  $\text{uniform}[0, 1]$  is a function for selecting a uniformly-distributed random number from the given interval. whereas the meaning of the rest of the symbols is as defined above.

SA has two parameters to control the search process: the initial temperature ( $t_{initial}$ ) and the total number of iterations or evaluations ( $it$ ). SA also allows to define the way the temperature is lowered during the annealing process.

SA has proved to be a solid optimization algorithm, capable of giving high-quality solutions to a wide scope of optimization problems [155]. Additionally, several works employ the SA metaheuristic for solving optimization problems in the area of radio network [19, 47, 54, 105, 141, 182].

## 2.4 Black-box optimization

The complexity of a problem is equivalent to the complexity of the best known algorithm solving that problem [160]. If there exists a polynomial-time algorithm to solve a problem,

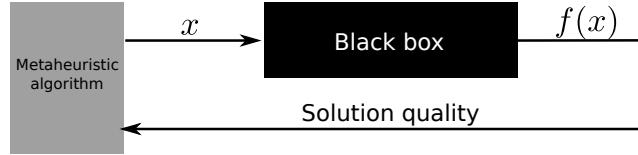


Figure 2.4: A metaheuristic algorithm process using a black box for the objective-function evaluation,  $f(x)$ , of a solution,  $x$ .

the problem is said to be easy or tractable. Similarly, if a problem is difficult or intractable, there is no known polynomial-time algorithm to solve it.

Many optimization problems cannot be formulated with a clear analytical mathematical notation. In such cases, the objective function may become a black box [87]. This is one of the main advantages when using metaheuristics, i.e., there is no need of a complete knowledge of the targeted model. Indeed, in a black-box optimization, no analytical formulation of the objective exists [160], as Figure 2.4 shows.

More specifically, a function  $f(\vec{x})$ ,  $\vec{x} \in \mathbb{R}^n$ , is a black-box function if and only if [160]:

- the domain  $\vec{x}$  is known,
- it is possible to get the value of  $f$  for each  $\vec{x}$  based on simulation, and
- there is no other information available for function  $f$ .

Typically, the experiments associated with these kind of problems are very expensive in terms of time and cost, since a simulation must be forced to evaluate the solution. Generally speaking, the most time-consuming part of a metaheuristic optimization process is the evaluation of the objective function [159]. This is especially true when dealing with real-world problems of areas such as structural design [14], molecular docking [161] and, the field on which this thesis focuses, radio-network design [18, 19, 20, 21, 22]. A possible substitution for lengthy evaluations is to reduce their complexity by approximating the objective function, thus replacing it with an approximation during the optimization process. This approach is known as meta-modeling [159]. However, when dealing with approximations, some degree of solution quality is inevitably sacrificed. As it will be shown in the following chapters, there is a very fine balance between the number of evaluations and the quality of the achieved solutions. Consequently, reducing the time spent in objective-function evaluation should favorably influence the solution quality achieved by a preferred metaheuristic algorithm. A major portion of this thesis is dedicated to improve this specific aspect in the area of radio-network optimization, starting with a high-performance, unified framework for radio-network planning, which is presented in Chapter 6.

In practice, however, the black-box evaluation of the objective function presents a problem. In the context of research about radio-network optimization there is an inherent difficulty of providing the black-box used to evaluate a given approach. A quick review of the state-of-the-art in radio-network optimization indicates that this fact has become increasingly regular in several published works [8, 29, 32, 66, 143, 146]. Clearly, this fact creates a barrier to one of the most important phases of scientific methodology: experimental reproducibility [57].

There are several reasons behind this situation. For example, it is a known fact that proprietary software, providing good computational models for radio-network simulation, is a very expensive tool for science. Even neglecting the economical aspect, but considering the great variety of software packages and license combinations, it is practically impossible

for a research laboratory to have the whole palette of commercially-available solutions at its disposal. Moreover, genuine users of these applications are generally not allowed to mention the formats, protocols, or algorithms used by the proprietary software, since their disclosure is often explicitly forbidden by restrictive licenses.

Some non-commercial and open-source packages for radio-network simulation [106, 85, 81, 123, 124, 179] present two main drawbacks: poor documentation and/or low scalability. The scalability issues shown by some projects restrain these packages to be used in larger, real-world environments, where big problem instances are the rule. Despite this, the merit and acknowledgement go to the authors of these frameworks, not only for providing them to the scientific community, but fundamentally because for providing an environment in which different kinds of simulations are completely reproducible. Additionally, the lack of documentation represents a big hurdle for extending the base code, which becomes a difficult task without the help of the original authors of the package. In such cases, author's knowledge is required for effectively expanding the functionality of an open-source tool. This is especially true when dealing with complex simulation frameworks, as the ones used for radio networks.

### 3 Principles of mobile radio networks

A cellular mobile radio network is a collection of individual cells that are served by several transmitters, called Base Stations (BSs). Each BS gives radio coverage to a small geographical area. The integration of the coverage of various BSs provides radio coverage over a much larger geographical area, thus defining a cellular radio network. The two basic functions of a radio-network system are:

- locate and track both active and idle mobile devices, called User Equipments (UEs); and
- attempt to connect each UE to the best available BS.

The first task involves a location-update procedure, which allows an UE to inform the network about its movement from one location area to the next. This process is called mobility management. The second task requires the constant evaluation of the radio-link quality with the serving BS, and the radio-link qualities of alternate BSs. This process is called radio-resource management, and is performed by the network using knowledge about the link-quality evaluations of the reference channels, e.g., the pilot channels.

The radio communications between an UE and a grid of BSs use low power. However, the movement of the UE causes highly irregular radio-link conditions, thus consistent monitoring and precise control are required to keep the radio-link quality at an acceptable level. At the core of the evaluation of radio-link quality is a statistical-measurement process based on a previous knowledge of the expected characteristics of the pilot channel. On the one hand, the link quality, and the size and distribution of the cells of a modern radio cellular system are limited by the speed of the link-quality measurement and network control. On the other hand, the spectral efficiency of a radio network is bounded by the cell sizes, the ability of radio links to withstand interference, and the ability of the system to react to variations in traffic [154].

Cellular radio systems partition the available spectrum among the BSs, and a given frequency is reused at the closest possible distance that the radio link will allow. Consequently, smaller cells have a shorter distance between reused frequencies, and this results in an increased spectral efficiency and traffic-carrying capacity. The radio links of a high-capacity radio network interfere with each other due to frequency reuse. For this reason, it is always desirable to use the lowest possible transmit power while maintaining each radio link above a given Quality-of-Service (QoS) threshold. Therefore, radio links should not significantly exceed their target QoS, since doing so will cause unnecessary interference to other radio links [154]. This particular situation is further discussed in Chapter 7, where an optimization approach, that minimizes the total transmit power used in a radio network, is presented.

#### 3.1 Handover

In radio networks, handover (or handoff) is one of the main features that allows the mobility of UEs [78]. The concept behind the handover operation is simple: when a UE moves from

the coverage area of a cell to the coverage area of a neighboring cell, the system creates a new connection with the latter cell and disconnects the user from the former one, so that an acceptable link quality can be maintained. Otherwise, the increase in transmit power, that is required to compensate for path loss, results in excessive interference. The handover procedure consists of two processes [154]:

- the link-quality evaluation before handover initiation, and
- the allocation of radio and network resources.

Generally speaking, radio networks with smaller cell sizes require faster and more reliable handover algorithms. Indeed, it has been shown that the number of cell-boundary crossings is inversely proportional to the cell size [94]. Since there is certain probability of dropping a connection whenever a handover is attempted, it is clear that the role of handover configuration becomes more important as the cell sizes decrease. Therefore, if the radio network does not detect poor signal quality fast enough, or makes too many handovers, the capacity is diminished due to increased interference and/or excessive control traffic [154].

### 3.1.1 Hard handover

During a hard handover, an UE can connect to only one BS at a time. A unique decision initiates and executes a handover without making a number of simultaneous connections among candidate BSs. Based on link measurements, the target BS is selected prior to executing the handover, and the active connection is instantly transferred to it. Moreover, the connection even experiences a brief interruption during the actual transfer, because the UE can only connect to one BS at a time. In contrast to soft handover (see next section), hard handovers do not take advantage of the diversity gain, where the signals from two or more BSs arrive at comparable strengths. Hard handover is a simple and inexpensive way to support UE mobility. It is used in Time-Division, Multiple-Access (TDMA) cellular systems such as GSM [154].

### 3.1.2 Soft handover

Soft Handover (SHO) enhances handover functionality by allowing an UE to potentially operate on multiple radio links at a time. During SHO, the target BS is selected as the best candidate from among the available BSs. The UE performs the necessary link-quality measurements by monitoring the signals from the surrounding BSs. Simultaneously keeping multiple connections means that SHO enhances the system performance through diversity reception.

Despite the advantages it provides, SHO is complex and expensive to implement. Additionally, interference actually increases with SHO, since several BSs can connect to the same UE. This increase in forward interference can become a problem. If the handover region is large, such that there are many UEs in SHO mode, the increased interference due to several BSs connected to the same UE can become a problem [154].

The SHO procedure is important in systems using the channel-access method known as Code Division Multiple Access (CDMA), and especially in Wide CDMA (WCDMA), which is employed by the UMTS. CDMA systems are interference-limited meaning their capacities are closely related to the level of interference they can tolerate. Specifically, a CDMA system cell is affected by the interference within its own cell, and also interference from its neighboring cells. In order to mitigate the level of interference, and thus increase the capacity and quality, CDMA systems use power control. The main idea behind power control is to prevent the UEs and the BSs from transmitting more power than is strictly

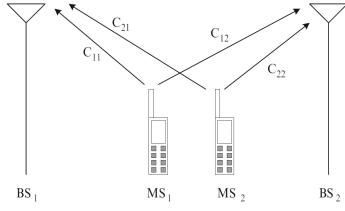


Fig. 13.2 Soft handoff example, from [219]

Figure 3.1: Example of two UEs in soft handover with two BSs. Adapted from [219???].

necessary to meet the target QoS level. For the power control to work properly, the system must ensure that each UE is connected to the BS having the least attenuation at all times. If this is not the case, a positive-feedback problem appears, and it can potentially destabilize the entire system [175]. The SHO procedure helps to prevent such situations by ensuring that each UE is served by the best BS most of the time, i.e., by allowing connections to multiple BSs.

The SHO condition depends on the relative, received-signal quality from different cells and the SHO window, which triggers the addition of a cell to the active set of the UE. Depending on radio propagation characteristics, the radio transmission can gain more than 3 dB out of a SHO situation [78]. From this point of view, SHO is a method to reduce interference and improve radio quality, particularly at the cell border where the radio coverage is of inferior quality. In UMTS Release 99 [163], SHO is specified to work from the BS towards the UE (downlink), and from the UE towards the BS (uplink).

With the introduction of the High Speed Packet Access (HSPA) as an improvement of the performance existing in WCDMA protocols, the role SHO plays in mobile network configuration and functioning slightly changed. The key difference is that the High Speed Downlink Packet Access (HSDPA) does not support SHO, whereas the High Speed Uplink Packet Access (HSUPA) does. This particular distinction is further discussed in Chapter 8, because it has some important implications in the balanced distribution of SHO areas, and thus in the quality and capacity of HSPA services [80].

### 3.2 Pilot signal and power

The CPICH transmit power is typically between 5% to 10% of the total downlink transmit power of the base station [95], but there is no standardized method to find a CPICH power setting.

The CPICH transmits in the downlink of a UMTS cell system. The transmit power is usually between 5% and 10% of the total power available at the base station [79]. The capacity of a cell is limited by the amount of available power at the base station and the interference level at the mobile terminal. The coverage area of any cell is controlled by changing its pilot power, which consequently modifies the service area of the network.

The CPICH transmit power is common to many different planning and optimization problems in UMTS networks [110].



## 4 Overview of radio-network optimization

Once a radio network is launched, an important part of its operation and maintenance deals with monitoring its quality characteristics. With the evolution of mobile communications, the complexity of network planning has grown along with its throughput capacity, thus making it practically impossible to plan modern radio networks with traditional methods. In this sense, an examination of coloured coverage maps in conjunction with some statistical analysis are no longer appropriate tools for troubleshooting a network. Moreover, since real-world radio networks are large and many of their configuration parameters are interdependent, an engineer is not able to cope with the level of complexity present in these systems. For this reason, the computer, along with specialized software, guides the engineer to the most appropriate configuration for the network. In the context of this thesis, this process is referred to as radio-network optimization.

Radio-network optimization may be divided into two fundamental phases: analysis and decision [111]. The analysis phase consists on the examination of the network performance, which mainly focuses on the definition and collection of Key-Performance Indicators (KPIs). KPIs are quantifiable measurements that reflect different network-quality factors. The second phase deals with the decision making, based on the analytical results collected in the previous phase, about changing a particular configuration or parameter setting. The process, a representation of which is depicted in Figure 4.1, is repeated until the achieved results are acceptable. Notice the similarity with the general, decision-making process that was presented in Chapter 1, Figure 1.1.

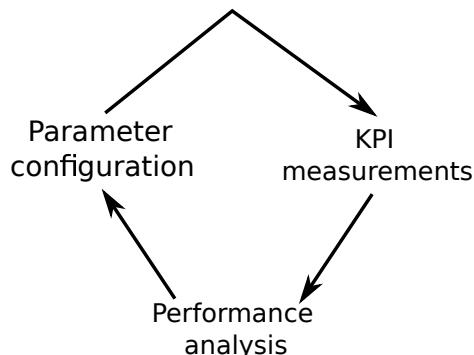


Figure 4.1: A typical optimization cycle for radio networks. This sequence is repeated until the achieved results are acceptable.

Since radio networks are increasingly more sophisticated, the need for optimization methods that are capable of coping with greater complexity is far from declining. Indeed, several radio-network optimization problems were shown to be NP-hard, since the computational time grows non-polynomially with the problem size [7, 9, 68, 73, 97, 129, 147]. As described in [111], there are other reasons directly related with the growth of already deployed networks that also increase the need for optimization methods:

**Network performance improvement** more users receive service coverage with the same physical infrastructure, making parameter optimization the less expensive and only viable short-term approach.

**Changes in user profile** the introduction of new, faster services puts additional stress on the infrastructure, requiring additional optimization efforts.

**Changes in the propagation conditions** the allocation of different frequency bands for different systems, e.g., GSM, UMTS or LTE, requires the deployment of new BSs, the radio propagations of which behave differently, especially in urban areas.

Depending on the optimization problem being addressed, network operators define an optimization target that is represented by an objective function that maps possible configurations into a real value. Unfortunately, there is no universal objective function in the field of radio-network optimization [111]. However, it is possible to optimize for a different target at a time, such as service coverage, SHO balance or signal propagation. Particular optimization algorithms for solving these problems are presented in Chapters 7, 8 and 9, respectively. In all three cases, the introduced optimization approaches are performed “offline”, meaning that the optimization software is not an active part of the target radio network. As the feedback information of each optimization target, the statistical data about the network functioning is used.

In the following, an overview of some well-known optimization problems for radio-mobile networks is given. Each section describes an optimization problem, and presents a short survey of recently proposed optimization methods for solving them.

## 4.1 Optimizing base-station locations

Some authors [74, 105] formulate the problem of locating BSs in terms of the minimum set-covering problem (see Figure 4.2). The set-covering problem is defined by considering the signal level in every test point from all BSs and requiring that at least one level is above a fixed threshold.



Figure 4.2: A graphical representation of the set-covering problem: (a) the problem input and (b) the solution.

A different formulation considers the BS-site location problem as a  $p$ -median problem [177], in which the BS location is the only decision variable considered. To each of the candidate solutions, an installation cost is also associated. The  $p$ -median problem constitutes seeking  $p$  different locations each time, regardless of how distant the sites are. The

problem involves selecting one installation-candidate site from each region such that the traffic capacity and the size of the covered area of the network are maximized with the lowest installation cost.

#### 4.1.1 Related work

Aydin et al. [13] proposed a solution to the  $p$ -median problem based on three metaheuristic algorithms: a genetic algorithm, SA, and tabu search. Their experimental study focused on the performance comparison between the three approaches.

In [177], the authors also used a simplified  $p$ -median problem as the model. They presented the results of extensive simulations to compare the performance of three different metaheuristic algorithms.

A solution to the set covering problem is proposed by Hao et al. [74]. An implementation of SA was developed to solve the formulated combinatorial problem. The presented results showed the feasibility of the proposed approach.

Mathar and Niessen [105] proposed a hybrid method that combines a linear-programming approach with SA. The SA algorithm substituted linear programming whenever an exact solution was out of reach because of the complexity of the problem instance.

Amaldi et al. [7] presented a discussion about the computational results of two different heuristics: greedy search and tabu search. The problem formulation was based on a set of candidate sites where the BSs could be installed, an estimation of the traffic distribution and a propagation description of the area to be covered. Some years later, the same authors [9] extended the problem formulation by adding the BS configuration and the hardware characteristics as additional constraints of the integer program. In both works, they proposed a mixed, integer-programming model to maximize the trade-off between covered area and installation costs.

Finally, Whitaker et al. [128] focused on providing a required service-coverage target at the lowest possible financial cost. Their work compared the performance of four different algorithms, namely SEAMO, SPEA2, NSGA-II and PESA, all targeting multiobjective optimization targets.

## 4.2 Optimizing antenna parameters

Since an antenna shapes the emitted energy, its configuration plays an important role in the coverage and interference of a radio network. The two most important parameters in this sense are the azimuth angle and the tilt (or elevation angle) of the antenna. The antenna azimuth, an example pattern of which is depicted in Figure 4.3, is the horizontal direction of the main antenna beam. The antenna tilt (see Figure 4.4) is defined as the angle of the main antenna beam relative to the horizontal plane. Both of these parameters have a great influence on network quality, although the antenna tilt usually requires less effort to implement, since most modern radio networks already support remote electrical tilt [12]. The adjustment of these two parameters optimize some important aspects of the network, e.g.:

- the path loss between the BS and the UE,
- the interference between neighboring cells, which leads to an overall capacity increase of the network.

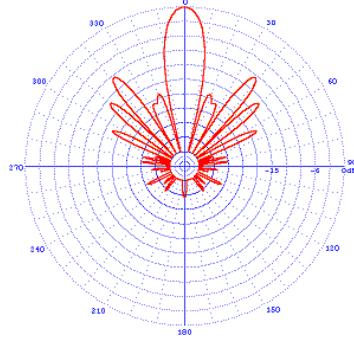


Figure 4.3: An example of an antenna-azimuth pattern.

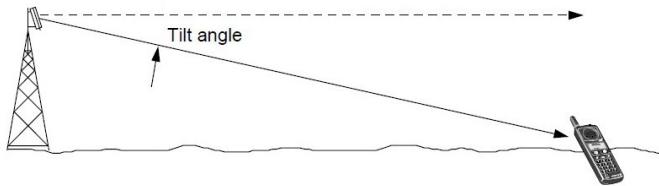


Figure 4.4: The antenna tilt, showing its angle with the horizontal plane.

#### 4.2.1 Related work

One of the approaches proposed by Gerdenitsch et al. [59] involves an “ad-hoc” strategy for adjusting antenna azimuth and downtilt by analyzing the structure of the network. The objective of this optimization was to improve the number of served users in the target area.

Siomina and Yuan [144] proposed an approach for automated optimization of antenna azimuth and tilt, including support for both mechanical and electrical tilt. The implementation introduced a SA-based algorithm that searches the solution space of feasible antenna configurations. The goal of the optimization was to improve power sharing among different cell channels and ultimately improve the throughput of the network.

In [67], the authors presented a compound optimization method containing two loops: the inner one and the outer one. The inner loop concentrated on frequency planning while the outer loop focused on finding the optimal settings of antenna azimuth and tilt for the current solution delivered by the inner loop. This approach is interesting because of its flexibility, e.g., the inner loop could be replaced with some other optimization objective, like service coverage.

In [48], the authors proposed an autonomous optimization approach for the antenna tilts. Based on the gradient-ascent method, the presented heuristic showed the fast convergence needed for an online-optimization method to be effective.

Combining the power setting of the pilot signal with the antenna configuration is also a common practice. For example, Siomina et al. [141] present an optimization approach for maximizing the service coverage that combines both antenna parameters with the power of the pilot signal. Their SA-based algorithm searches the solution space of possible configurations in order to improve the performance of the target radio network by reducing the total interference. The simulation results show the algorithm is capable of tackling large network instances.

Two optimization algorithms for finding an optimal setting of antenna tilt and pilot-signal power were also introduced in the previously cited work by Gerdenitsch et al. [59].

The first algorithm is based on a rule-based approach, while the second one extends it by incorporating SA. The evaluation of both techniques showed that the second algorithm achieves better solutions.

In a different work, Gerdenitsch et al. [60] proposed a genetic algorithm to tackle the optimization of antenna tilt and pilot-signal power, the goal of which is to increase network capacity. The implementation involves a deterministic fitness selection scheme, a problem-specific recombination operator and an improved mutation operator. After the initial identification of the best promising individuals, a local-optimization technique is applied to improve their fitness.

### 4.3 Optimizing coverage

Coverage is arguably the most common optimization objective considered in radio-network optimization. A general objective function for coverage optimization can be defined as follows:

$$f_{\text{cov}} = \frac{A_{\text{covered}}}{A_{\text{total}}},$$

where  $A_{\text{covered}}$  represents the area covered by the network and  $A_{\text{total}}$  represents the total area under optimization. Thus, the expression  $f_{\text{cov}}$  represents the proportion of the total area that is actually under network coverage, the value of which ranges from 0 (no coverage) to 1 (total coverage).

The area under optimization is usually divided into squares (or pixels) of a certain size, creating a Regular Square Grid (RSG) of a certain resolution. A pixel is considered covered if a given QoS measure is above a defined threshold. It is also common to use a binary function, e.g.,  $\text{cov}(x, y)$ , to check the coverage of a given pixel. The function returns 1 if the pixel located at  $(x, y)$  is covered, and 0 otherwise.

#### 4.3.1 Related work

Siomina and Yuan [148] considered the problem of minimizing the pilot-signal power subject to a coverage constraint. Their approach consists of an iterative mathematical program, based on a linear-integer formulation of the problem. The simulation results showed the trade-off between service coverage and power consumption for different test networks.

Almaldi et al. [9] investigated several mathematical-programming models for supporting decisions on where to install new BSs and how to select their configuration in order to find a trade-off between coverage and costs. The overall model takes into account signal-quality constraints, in both uplink and downlink directions, as well as the pilot-signal power.

Connecting UMTS and LTE-network optimization from the coverage point of view, Salient et al. [133] presented a framework to automatically identify a cell with sub-optimal coverage. The framework input is a collection of KPIs, including usage statistics and UE field measurements. The authors used the experimental results to provide a projection for the LTE self-optimization functionality.

In [162], the authors introduced a sparse-sampling algorithm to optimize the coverage of LTE networks by means of antenna tilt. By applying a reinforcement-learning technique, their approach optimizes coverage without prior knowledge about the target network.

Parkkinen et al. [167] presented a gradient-descent method to minimize an objective function that combines some KPIs with the service coverage. The power of the pilot signal of the cells is periodically updated in order to improve the afore-mentioned function.

## 4.4 Discussion

Regarding the optimization methods presented in the previous sections, three distinctive groups emerge: genetic algorithms, linear programming and other search methods.

**Genetic algorithms** These algorithms work on a population of solutions, allowing a more comprehensive search for optimal solutions. As a direct consequence, an increase in running time is commonly observed. The implementation effort of genetic algorithms is to some degree higher than for simpler search methods, e.g., local search, but their inherent structure makes parallel implementations rather simple. Additionally, genetic algorithms are less likely of being trapped in local optima, since they are a population-based metaheuristic.

**Linear programming** Linear-integer or mixed-integer programming are widely used in different optimization areas and there are many good software packages to solve such problems. Consequently, if a problem can be modeled as a continuous linear problem, there is usually no difficulty in finding optimality. In the context of this survey, linear programming has proven useful for BS-location optimization in early network planning stages.

**Other search methods** Other search methods, e.g., local search, SA, tabu search or gradient descent, usually represent a compromise between running time and quality of results. Their effectiveness relies on evaluating a great number of alternative solutions. The number of parameters taken into account, as well as the evaluation precision, directly influence their running time. These methods don't excel in full simulation scenarios. Moreover, some of them are easily trapped in local minima.

## 4.5 Summary

The variety of optimization problems described in the previous sections differ in many aspect, like implementation, running time and solution quality. Picking the right method for a given situation depends on the optimization task and the desired results. Since computational time is usually an important restriction, simpler and faster methods may be preferable.

Beside the convenience of a literature overview as presented here, it is very important to develop a feeling for the properties, advantages and drawbacks of the respective methods. Moreover, the recommendations of experts regarding the interpretation of the solutions and the feedback from everyday network operation are an essential input for establishing high-quality optimization methods.

Additionally, as noted for some of the cited works, hybrid methods, i.e., approaches that combine different optimization and/or search techniques, usually give competitive solutions to complex optimization problems. Therefore, a simple optimization algorithm can quickly find a subset of reasonable solutions, whereas a more complex method could be applied afterwards, to refine the search. Sometimes it may also be useful to apply a local search method at the end to find better solutions in the proximity of a current one.

## 5 Principles of GPU programming

During the past few years, the computing industry has been delivering extra processing power in the form of parallel computing, i.e., more cores instead of higher-frequency rates. Concurrently to this situation, new hardware architectures have appeared. Among them is the programmable Graphics Processing Unit (GPU), the computing capacity of which is already showing a faster progress compared to CPUs. The peak performance of the latest GPU is several times greater than the latest CPU, but even more important is the trend [40].

The GPU market was almost exclusively driven by the games industry. This fact makes this commodity hardware cheaper than other alternatives for High-Performance Computing (HPC), e.g., the classic ‘mainframe’ servers. Moreover, the performance-per-watt of GPUs is an extra benefit over their raw performance [40].

In February 2007, the first real opportunity for using GPUs for general programming and scientific computing came about with the release of the Compute Unified Device Architecture (CUDA).

In the following, the focus on the CUDA architecture. In particular, similar hardware produced by other vendors, e.g., AMD and Intel, as well as other programming frameworks like OpenCL, are based on the same technology paradigm of massively-parallel processors. For this reason, the principles presented in this section can be applied to current GPU technologies, in a vendor-independent manner.

### 5.1 CUDA

A GPU is effectively a large set of processor cores with the ability to directly address into a global memory. This makes it easier for developers to implement data-parallel kernels. The CUDA programming model [117] was created for developing applications for the GPU platform. A system within this model consists of a host, i.e., a traditional CPU, and one or more compute devices that are massively data-parallel coprocessors, i.e., a GPU. The host code transfers data to and from the global memory of the GPU using CUDA-function calls. Each processor of a CUDA device supports the Single-Program Multiple-Data (SPMD) model [11], in which all concurrent threads are based on the same code, although they may not follow exactly the same path of execution.

A CUDA program consists of multiple parts that are executed on either the CPU or the GPU. For this reason, the CUDA programming model may be viewed as an environment to isolate program parts that are rich in data parallelism and thus should be executed on the GPU. The parallel parts of a CUDA program are implemented as kernels, i.e., functions written in the C language that describe the work of a single thread of execution. Several restrictions apply on kernel functions: there must be no recursion, no static variable declarations, and a non-variable number of arguments [131]. At run time, the kernel invocation is typically done on thousands of threads, which are organized within developer-defined groups called thread blocks. The threads can share data and synchronize their execution only within the block they belong to. Thread blocks are, in turn, organized in a higher-hierarchy level called grid. All threads within a given grid execute the same kernel function.

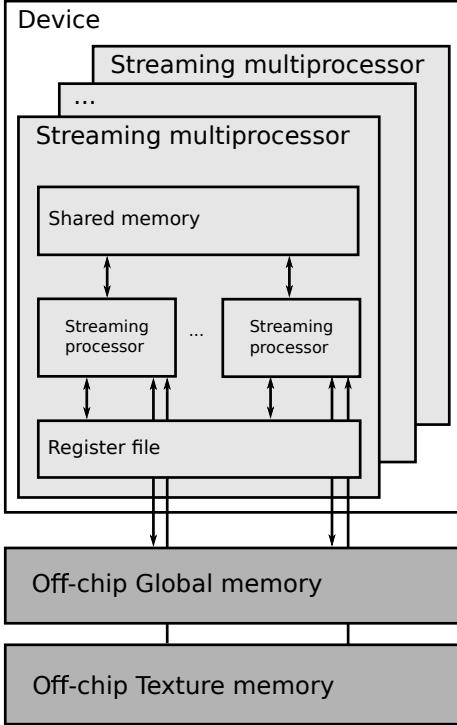


Figure 5.1: A simplified diagram of the architecture of a modern GPU. Adapted from [131].

During execution, continuous sections of threads within a block are grouped into warps of 32 parallel threads, which represent the multi-threading scheduling unit [131]. A Streaming Multiprocessor (SM) executes one instruction at a time for all the 32 threads in the warp. Consequently, if a thread block is not evenly divisible by the warp size, any remaining thread slots are wasted. Another undesirable situation appears when threads in a warp take different control paths, since the execution performance of the entire warp is reduced.

The architecture of a typical GPU is depicted in Figure 5.1. It consists of  $n$  SMs, each containing  $m$  Streaming Processors (SPs). Each SP executes a single instruction of a thread in a Single-Instruction, Multiple-Thread (SIMT) manner [40]. The registers of each SM are dynamically partitioned among the threads running on it.

Table 5.1 lists the different types of memory on a GPU, along with some of their properties. Specifically, the location of the memory, its hit-latency in terms of clock cycles, whether it is read-only or not, and the program scope it may be accessed from. Due to the different memory system a GPU has, the actual throughput an application can achieve depends on issues related to the access to memory, in particular the slow accesses to global memory from the GPU chip, and the use of shared memory in the SPs to mitigate the high-latency effects [40]. Since variables in the source code can be declared to reside in global, shared, local, or constant memory, a programmer has the means to organize the code in such a way that the application throughput can be maximized.

Clearly, there are hard limits to the memories, threads, and total bandwidth available to an application running on a GPU. Managing these limits is critical when optimizing applications, but applying strategies for avoiding one limit can easily cause other limits to be hit. Additionally, managing the behavior of threads so that those in the same warp follow the same control paths and load contiguous values from global memory can improve the execution performance [131].

A detailed discussion of the CUDA programming model can be found in [51].

Table 5.1: Properties of different memory types on a GeForce 8800 GPU. Adapted from [131].

Memory	Location	Latency	Read-only	Program scope
Global	off-chip	200-300 cycles	no	global
Texture	on-chip cache	~100 cycles	yes	global
Shared	on-chip	~register	no	function
Register	on-chip	~1 cycle	no	function

Table 5.2: Naming-convention translation between OpenCL and CUDA. Adapted from [90].

OpenCL	CUDA
Grid	Grid
Work group	Block
Work item	Thread
__kernel	__global__
__global	__device__
__local	__shared__
__private	__local__
imagend_t	texture<type,n,...>
barrier(L M F)	__syncthreads( )
get_local_id(0 1 2)	threadIdx.x y z
get_group_id(0 1 2)	blockIdx.x y z
get_global_id(0 1 2)	(not implemented)

## 5.2 OpenCL

The Open computing language (OpenCL) [151] is an open parallel computing API designed to enable GPUs and other co-processors to work together with the CPU, providing additional computing power. As a standard, OpenCL 1.0 was released in 2008, by The Khronos Group, an independent standards consortium [109].

The main advantage of OpenCL over CUDA relies on the fact that its source code can be compiled to run on a variety of hardware, including multicore CPUs and GPUs from different vendors. This provides a complete framework that is capable of exploiting the parallel features of different hardware without the need of changing the implementation. Moreover, being an open standard, its users are not tight to the decision of only one vendor. As it was mentioned before, the details described in the previous sections may be equally applied on CUDA and OpenCL.

One unfortunate consequence of the vendor variety is that CUDA and OpenCL have each introduced its own naming conventions. For the sake of consistency, in Table 5.2, a short “translation dictionary” between both platforms is presented.

In the remaining of this thesis, the naming convention introduced by the CUDA platform is used.

## 5.3 Summary

This chapter gave an overview of the basic concepts, the potential and the limitations of the GPUs when used as parallel processors for general programming. In particular, emphasis has been given to the details that differentiate this platform from the traditional serial architecture on CPUs. In this sense, it is important to recognize which applications can

benefit from using a GPU and which not, also taking into consideration the effort required at the implementation time.

## 6 A parallel framework for radio-network planning and optimization

In this chapter, a parallel framework for radio-coverage simulation is presented. The objective of the framework is to provide an environment for the radio-coverage prediction of large radio networks. Due to its high performance, the framework also enables the evaluation of more complex optimization problems for radio networks.

The framework is implemented as a module of a geographical information system, since the prediction calculation employs digital elevation models and land-usage data. Following a master-worker parallel paradigm over a message-passing communication model proved to be a bottleneck for the performance of the parallel module. A new approach, that overcomes this performance constraint, is introduced in this chapter. The efficiency improvement is based on overlapping process execution and communication. This minimizes the idle time of the worker processes and thus improves the overall efficiency of the system. To this end, the intermediate calculation results are saved into an external database (DB) instead of sending them back to the master process. This approach is implemented as part of a parallel radio-prediction tool (PRATO) for the open-source Geographic Resources Analysis Support System (GRASS) [114]. An extended analysis of the experimental results is provided, which are based on real data from an LTE network currently deployed in Slovenia. Based on these experiments, which were performed on a computer cluster, the new technique exhibits better scalability than the traditional master-worker approach. Some real-world data sets are presented, the coverage predictions of which are calculated in a shorter time while saturating the hardware utilization.

The content of this chapter extends the research work published by the author in [17]. The rest of this chapter is organized as follows. Section 6.2 gives an overview of the relevant publications, describing how they relate to the presented work. Section 6.3 gives a description of the radio-coverage prediction problem, including the radio-propagation model. Section 6.4 concentrates on the design and implementation of the radio-propagation tool, for both the serial and parallel versions. Section 6.5 discusses the experimental results and their analysis.

## 6.1 Motivation

Although Gordon Moore's well-known and often cited prediction still holds [108], the fact is that for the past few years, CPU speeds have hardly been improving. Instead, the number of cores within a single CPU is increasing. This situation poses a challenge for software development in general and research in particular: a hardware upgrade will, most of the time, fail to double the serial-execution speed of its predecessor. However, since this commodity hardware is present in practically all modern desktop computers, it creates an opportunity for the parallel exploitation of these computing resources in order to enhance the performance of complex algorithms over large data sets. The challenge is thus to deliver the computing power of multi-core systems in order to tackle a computationally time-consuming problem, the completion of which is unfeasible using traditional serial approaches. Indeed, improving the performance and data-set sizes a well-known approach may handle opens new possibilities for models and implementations of enhanced analytical methods.

A traditional approach when dealing with computationally expensive problem solving is to simplify the models, thus reducing the time needed for their calculation. Clearly, this method increases the introduced error level, which is not an option for a certain group of simulations, e.g., those dealing with disaster-contingency planning and decision support [84, 180]. The conducted simulations during the planning phase of a radio network also belong to this group. Their results are the basis for the decision making prior to physically installing the BSs and antennas that will cover a certain geographical area. A larger deviation of these results increases the probability of making the wrong decisions at installation time, which may considerably increase the costs or even cause mobile-network operators to incur losses.

Various researchers have successfully deployed high-performance computing (HPC) systems and techniques to solve different problems dealing with spatial data [6, 10, 71, 84, 99, 180, 119, 156, 157, 158, 174]. Their work confirms that a parallel paradigm such as master-worker, and techniques like work pool (or task farming) and spatial-block partitioning are applicable when dealing with parallel implementations over large spatial data sets. However, it is well known that parallel programming and HPC often call for area experts in order to integrate them into a given environment [36]. Moreover, the wide range of options currently available creates even more barriers for general users wanting to benefit from HPC.

In the following, a high-performance, radio-propagation prediction tool for GSM (2G), UMTS (3G) and LTE (4G) networks is presented. It can be used for planning the different phases of a new radio-network installation, as well as a support tool for maintenance activities related to network troubleshooting in general and optimization in particular. Specifically, automatic radio-coverage optimization requires the evaluation of millions of radio-propagation predictions in order to find a good solution set. This is often unfeasible using other serial implementations of academic or commercial tools [81, 106, 124].

As a reference implementation, the publicly available radio-coverage prediction tool, presented in [81], was used. The authors developed a modular radio-coverage tool that performs separate calculations for radio-signal path loss and antenna-radiation patterns, also taking into account different configuration parameters, such as antenna tilting, azimuth and height. The output result, saved as a raster map, is the maximum signal level over the target area, in which each point represents the received signal from the best-serving transmitter (or cell). This work implements some well-known radio-propagation models, e.g., Okumura-Hata [75], the description of which is latter presented in Section 6.3.2. Regarding the accuracy of the predicted values, the authors reported comparable results to those of an industrial tool [81]. To ensure that the presented implementation is completely compliant with this reference, a comparison test was designed, that consists of running both tools with the same set of input

parameters. The test results from PRATO and the reference implementation were identical in all the tested cases.

## 6.2 Related work

The task-parallelization problem within the GRASS environment has been addressed by several authors in a variety of studies. For example, in [27], the authors presented a collection of GRASS modules for a watershed analysis. Their work concentrates on different ways of slicing raster maps to take advantage of a Message Passing Interface (MPI) implementation.

In the field of HPC, the authors of [6] presented MPI and Ninf-G implementation examples of a GRASS raster module, that processes vegetation indexes from satellite images. The authors acknowledge a limitation in the performance of their MPI implementation for big processing jobs. The restriction appears due to the computing nodes being fixed to a specific spatial range, since the input data are equally distributed among worker processes, creating an obstacle for load balancing in heterogeneous environments.

Using a master-worker technique, the work presented in [82] abstracts the GRASS data types into its own *struct* and MPI data types, thus not requiring the GRASS in the worker nodes. The data are evenly distributed by row among the workers, with each one receiving an exclusive column extent to work on. Their test cluster contained heterogeneous hardware configurations. The authors noted that data-set size is bounded by the amount of memory on each of the nodes, since they allocate the memory for the whole map as part of the set-up stage, before starting the calculation. Regarding the data sets during the simulations, the largest one contained 3,265,110 points. They concluded that the data-set size should be large enough for the communication overhead to be hidden by the calculation time, so that the parallelization pays off.

In [156], the authors employed a master-worker approach, using one worker process per worker node. The complete exploitation of the computing resources of a single computing node is achieved with OpenMP. The experimental environment featured one host. The horizon-composition algorithm presents no calculation dependency among the spatial blocks. Consequently, the digital elevation model (DEM) was divided into separate blocks to be independently calculated by each worker process. The authors presented an improved algorithm that can also be used to accelerate other applications like visibility maps. The tasks are dynamically assigned to idle processes using a task-farming paradigm over the MPI.

Similar to [156], in [157] there was no calculation dependency among the spatial blocks. The experimental evaluation was made over multiple cores of one CPU and a GPU, and a master-worker setup was used for process communication.

In [180], the authors presented a parallel framework for GIS integration. Based on the principle of spatial dependency, they lowered the calculation-processing time using a knowledge database, delivering the heavy calculation load to the parallel back-end if a specific problem instance is not found in the database. There was an additional effort to achieve the presented goals, since the implementation of a fully functional GIS (or “thick GIS” as the authors call it) was required on both the desktop client and in the parallel environment.

An agent-based approach for simulating spatial interactions was presented in [65]. This technique decomposes the entire landscape into equally-sized regions, i.e., a spatial-block division as in [156], which are in turn processed by a different core of a multi-core CPU. This work used multi-core CPUs instead of a computing cluster.

Some years ago, grid computing received a lot of attention from the research community. It appeared to be a good alternative for accessing the extra computational power needed for the spatial analysis of large data sets [10, 171, 173]. However, several obstacles are still preventing this technology from being widely used. In particular, its adoption requires

not only hardware and software compromises with respect to the involved parts, but also a behavioral change at the human level [10].

## 6.3 Radio-coverage prediction for mobile networks

### 6.3.1 Background

As it was mentioned in Chapter 3, radio communications in a mobile network take place between a BS (or fixed transmitter) and a number of UEs (or mobile receivers). The effects of signal propagation limit the performance of a mobile-radio system. For this reason, the characterization and modeling of radio propagation is considered a fundamental aspect in radio-network planning [4]. Consequently, understanding the mathematical modeling of a frequency channel is necessary to accurately predict the system performance and to provide a mechanism to analyze the effects caused by signal propagation [122].

The coverage planning of radio networks is a key problem that all mobile operators have to deal with. Moreover, it has proven to be a fundamental issue, not only in LTE networks, but also in other standards for mobile communications [132, 137, 148, 166]. One of the primary objectives of mobile-network planning is to efficiently use the allocated frequency band to ensure that some geographical area of interest can be satisfactorily reached with the BSs of the network. To this end, radio-coverage prediction tools are of great importance, as they allow network engineers to test different network configurations before physically implementing the changes. Radio-coverage prediction is a complex task, mainly due to the several combinations of hardware and configuration parameters that have to be analyzed in the context of different environments. The complexity of the problem means that radio-coverage predictions are computationally-intensive and time-consuming, hence the importance of using fast and accurate tools (see Section 6.4.4 for a complexity analysis of the algorithm). Additionally, since the number of deployed transmitters keeps growing with the adoption of modern standards [132], there is a clear need for a radio-propagation tool that is able to cope with larger work loads in a feasible amount of time (see Section 6.4.4 for the running time of the serial version).

### 6.3.2 Radio-propagation model

PRATO uses a modified version of the well-known Okumura-Hata model for radio-propagation predictions [75]. Other accurate methods exist, like the ones based on ray tracing [38, 170]. However, these methods are more sensible to deviations in input data, like DEMs and buildings, and are still inefficient in terms of the computational effort required to achieve satisfying results. Empirical methods for radio-propagation predictions, like Okumura-Hata, give acceptable results within a feasible amount of time. For this reason, they have become the industry standard for non-deterministic signal-propagation calculations [16, 35, 76, 137, 149].

The Okumura-Hata model has been largely studied and shown to be suitable for predicting the radio propagation in LTE networks [4, 137]. In its primary form, the model distinguishes the distance from the receiver to the transmitter, the frequency used and the effective antenna height, i.e., the height of the antenna above the receiver's level. These variables are taken into account in order to calculate the path loss in line-of-sight (LOS) conditions. Additionally, for distinguishing non-line-of-sight (NLOS) conditions, the terrain profile and Earth shape are added to the original formula. In this context, a NLOS situation appears when the first Fresnel zone is obscured by at least one obstacle [176]. Equation (6.1) describes the path loss when there is LOS between the transmitter and the receiver.

$$\begin{aligned} pl_{\text{LOS}}(d_{(x,y)}, \beta) = & a_0 + a_1 \log(d_{(x,y)}) + a_2 \log(H_A) + \\ & a_3 \log(d_{(x,y)}) \log(H_A) - 3.2 [\log(11.75 \cdot H_R)]^2 + \\ & 44.49 \log(F) - 4.78 [\log(F)]^2, \quad (6.1) \end{aligned}$$

where  $\beta = (a_0, a_1, a_2, a_3)$  is the vector containing the control parameters of the model,  $d_{(x,y)}$  is the distance (in kilometers) from the transmitter to the topography point with coordinates  $(x, y)$ ,  $H_A$  is the effective antenna height (in meters) of the transmitter,  $H_R$  is the antenna height (in meters) of the receiver, and  $F$  is the frequency, expressed in MHz. In NLOS conditions, the path loss is calculated as in Equation (6.2):

$$pl_{\text{NLOS}}(d_{(x,y)}) = \sqrt{[\alpha K(d_{(x,y)})]^2 + E(d_{(x,y)})^2}, \quad (6.2)$$

where  $\alpha$  is the knife-edge diffraction control parameter, the value of which is calculated based on the level of obstruction of a Fresnel zone,  $K(d_{(x,y)})$  is the knife-edge diffraction loss (in dB), and  $E(d_{(x,y)})$  is the correction due to the Earth sphere (in dB). All three values depend on the characteristics of the topography point with coordinates  $(x, y)$ .

In this work, as well as in [52], the terrain profile is used for LOS determination, i.e., an obstacle obstruction in the first Fresnel zone of the transmitter. In order to adequately predict signal-loss effects due to foliage, buildings and other fabricated structures, additional loss factors based on the land usage (clutter data), are included. This technique is adopted by several propagation models for radio networks [3, 16, 113]. Consequently, an extra term is introduced for signal loss due to clutter, thus defining the model-predicted path loss as

$$pl(d_{(x,y)}, \beta) = pl_{\text{LOS}}(d_{(x,y)}, \beta) + pl_{\text{NLOS}}(d_{(x,y)}) + pl_{\text{CLUT}}(d_{(x,y)}), \quad (6.3)$$

where  $pl_{\text{CLUT}}(d_{(x,y)})$  represents the clutter loss at the topography point with coordinates  $(x, y)$ , expressed in dB.

## 6.4 Design and implementation

### 6.4.1 Geographic Resources Analysis Support System

GRASS [114], a free and open-source software project that implements a Geographical Information System (GIS), is used as the software environment for PRATO. This GIS software was originally developed at the US Army Construction Engineering Research Laboratories and is a full-featured system with a wide range of analytical, data-management, and visualization capabilities. Currently, the development of GRASS GIS is supported by a growing community of volunteer developers.

The use of GRASS GIS as an environment for PRATO presents many advantages. First, the current development of GRASS is primarily Linux-based. Since the field of HPC is dominated by Linux and UNIX systems, an environment with Linux support is critical for this work. Software licensing is another important consideration for choosing GRASS, since it is licensed under the GNU Public License [150] that imposes the availability of the source code. This allows to make potential modifications to the system, thus adapting it for the parallel computation environment. Moreover, GRASS provides a great deal of built-in functionality, capable of operating with raster and vector topological data that can be stored in an internal format or a DB.

### 6.4.2 Multi-paradigm parallel programming

The implementation methodology adopted for PRATO follows a multi-paradigm, parallel programming approach in order to fully exploit the resources of each node in a computing cluster. This approach combines a master-worker paradigm with an external DB. To efficiently use a shared memory multi-processor on the worker side, and to effectively overlap the calculation and communication, PRATO uses POSIX threads [26].

To use the computing resources of a distributed memory system, such as a cluster of processors, PRATO uses the MPI [70]. The MPI is a message-passing standard that defines the syntax and semantics designed to function on a wide variety of hardware. The MPI enables multiple processes, running on different processors of a computer cluster, to communicate with each other. It was designed for high performance on both massively parallel machines and on workstation clusters.

PRATO also supports the execution of the most computationally-intensive parts of the radio-propagation algorithm on a GPU. Moreover, the GPU hardware is used if it is available on the computing nodes that host the worker processes (see Section 6.4.5.3 later in this chapter for a discussion about the GPU implementation).

In order to make the text clearer and to differentiate between the programming paradigms used from here on, a POSIX thread will be referred to simply as a ‘thread’ and a MPI process as a ‘process’.

### 6.4.3 Design of the serial version

This section describes the different functions contained in the serial version of PRATO, which is implemented as a GRASS module. Their connections and data flow are depicted in Figure 6.1, where the parallelograms of the flow diagram represent the input/output (I/O) operations.

The design follows a similar internal organization as the radio-planning tool presented in [81], but with some important differences. Specifically, the design presented here employs a direct connection to an external database server for intermediate result saving, instead of the slow, built-in GRASS database drivers. To explicitly avoid tight coupling with a specific database vendor, the generated output is formatted in plain text, which is then forwarded to the DB. Any further processing is achieved by issuing a query over the database tables that contain the path-loss results for each of the processed transmitters.

#### 6.4.3.1 Input parameters

All input data are read in the first step (see “Read input data” in Figure 6.1). Their formats differ based on the data they contain, i.e.,

- GRASS raster files are used for the DEM and clutter data, whereas
- a text file is used for the transmitter configurations and other service-dependent options.

Since the module accepts a considerable amount of input parameters, they are read from a text-based initialization (INI) file. This is far more practical than passing them as command-line parameters, which would make them error-prone and difficult to read. Besides, the INI file may contain configuration parameters for many transmitters. The user selects which one(s) to process at run-time by passing a command-line option.

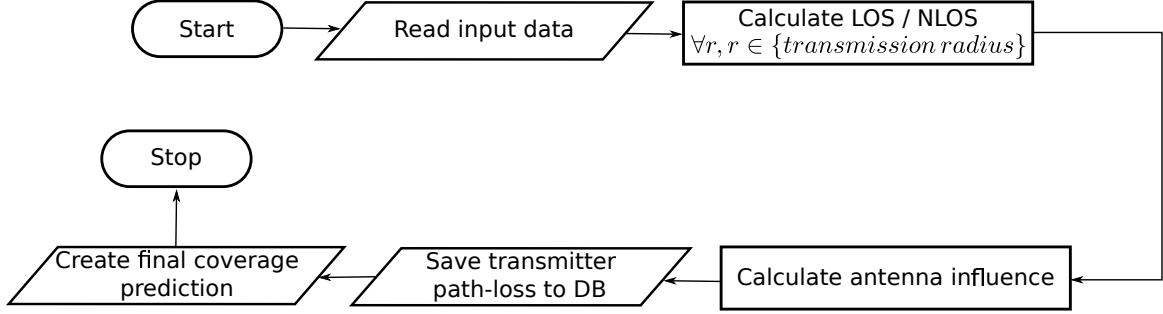


Figure 6.1: Flow diagram of the serial version.

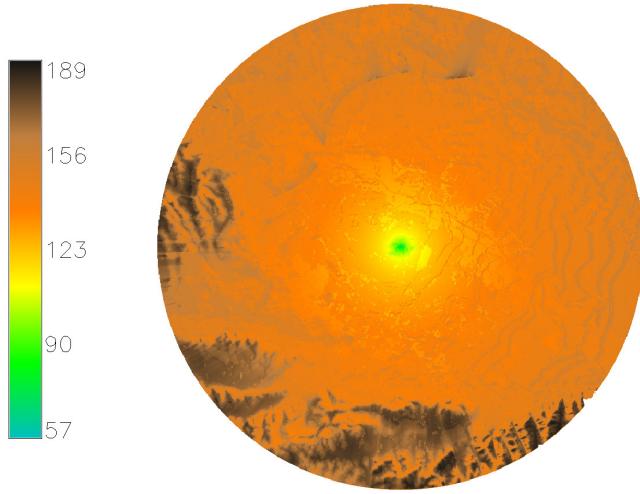


Figure 6.2: Example of a raster map, showing the result of a path-loss calculation from an isotropic source.

#### 6.4.3.2 Isotropic path-loss calculation

This step starts by calculating which receiver points,  $r$ , are within the specified transmission radius (see “*transmission radius*” in Figure 6.1). The transmission radius is defined around each transmitter in order to limit the radio-propagation calculation to a reasonable distance. For these points, the LOS and NLOS conditions are calculated with respect to the transmitter (see “Calculate LOS/NLOS” in Figure 6.1). The following step consists of calculating the path loss for an isotropic source (or omni antenna). This calculation is performed by applying the radio-propagation model, which was previously defined in Equation (6.3), to each of the points within the transmission radius around the transmitter.

Figure 6.2 shows an example of the isotropic path-loss calculation, only including the map area within the transmission radius. The color scale is given in dB, indicating the signal loss from the isotropic source of the transmitter, located at the center. Notice the hilly terrain is clearly distinguished due to LOS and NLOS conditions from the signal source.

#### 6.4.3.3 Antenna diagram influence

This step considers the antenna radiation diagram of the current transmitter and its influence over the isotropic path-loss calculation (see “Calculate antenna influence” in Figure 6.1). Working on the in-memory results generated by the previous step, the radiation diagram of the antenna is taken into account, including the beam direction, the electrical and the mechanical tilt. Figure 6.3 shows the map area within the transmission radius, where this

calculation step was applied to the results from Figure 6.2. Notice the distortion of the signal propagation that the antenna has introduced.

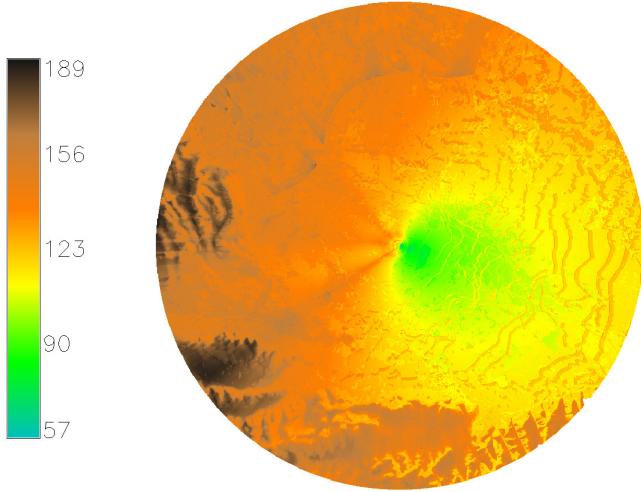


Figure 6.3: Example of a raster map, showing the antenna influence over the isotropic path-loss result, as depicted in Figure 6.2.

#### 6.4.3.4 Transmitter path-loss prediction

In this step, the path-loss prediction of the transmitter is saved in its own database table (see “Save transmitter path-loss to DB” in Figure 6.1). This is accomplished by connecting the standard output of the GRASS module with the standard input of a database client. Naturally, the generated plain text should be understood by the DB itself.

#### 6.4.3.5 Coverage prediction

The final radio-coverage prediction, containing the aggregation of the partial path-loss results of the involved transmitters, is created in this step (see “Create final coverage prediction” in Figure 6.1). The received signal strength from each of the transmitters is calculated as the difference between its transmit power and the path loss for the receiver’s corresponding position. This is done by executing an SQL query over the tables containing the path-loss predictions of each of the processed transmitters. Finally, the output is generated using the GRASS built-in modules *v.in.ascii* and *v.to.rast*, which create a raster map using the query results as the input. The resulting raster map contains the maximum received signal strength for each individual point, as shown in Figure 6.4. In this case, the color scale is given in dBm, indicating the strongest received signal strength from the transmitters.

#### 6.4.4 Computational complexity

In this section, the time complexity of the radio-coverage prediction algorithm is presented, the pseudo-code of which is listed in Algorithm 6.1.

The algorithm starts by loading the input DEM and clutter data. Both RSGs should account for the same area and resolution, consequently containing the same number of pixels,  $M$ . The transmitter data is then loaded into set  $T_x$ , the cardinality of which is denoted as  $n = |T_x|$ . For each transmitter  $t_x \in T_x$ , a smaller subarea of the DEM and clutter data, denoted as  $DEM_{t_x}$  and  $Clut_{t_x}$ , respectively, is delimited around  $t_x$ , based on a given transmission radius. The number of pixels within this sub-area is denoted as  $m$ , and its

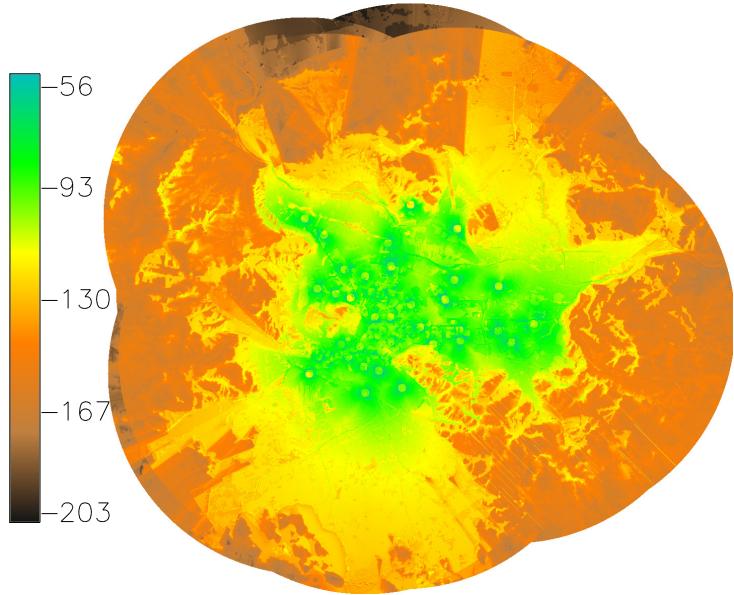


Figure 6.4: Example of a raster map, displaying the final coverage prediction of 136 transmitters over a geographical area. The color scale is given in dBm, indicating the received signal strength. Darker colors denote areas with a reduced signal due to the fading effect of the hilly terrain and clutter.

---

**Algorithm 6.1** Pseudo-code of the radio-coverage prediction algorithm. The time complexity is given per line.

---

```

 $DEM \leftarrow$  DEM of the whole area.                                 $\triangleright O(M)$ 
 $Clutter \leftarrow$  signal losses due to clutter of the whole area.  $\triangleright O(M)$ 
 $T_x \leftarrow$  transmitter configuration data.                       $\triangleright O(n)$ 
for all  $t_x \in T_x$  do                                      $\triangleright O(n \cdot m^2)$ 
     $DEM_{t_x} \leftarrow$  DEM area within transmission radius of  $t_x$        $\triangleright O(m)$ 
     $Clut_{t_x} \leftarrow$  Clutter area within transmission radius  $t_x$         $\triangleright O(m)$ 
     $LoS_{t_x} \leftarrow$  LineOfSight ( $DEM_{t_x}$ )                                $\triangleright O(m^2)$ 
     $PL_{t_x} \leftarrow$  PathLoss ( $DEM_{t_x}, Clut_{t_x}, LoS_{t_x}$ )            $\triangleright O(m^2)$ 
     $Diag_{t_x} \leftarrow$  Antenna diagram of  $t_x$                             $\triangleright O(1)$ 
     $PL_{t_x} \leftarrow$  AntennaInfluence ( $Diag_{t_x}, PL_{t_x}$ )             $\triangleright O(m)$ 
end for
for all  $t_x \in T_x$  do                                      $\triangleright O(n \cdot m)$ 
     $CoveragePrediction \leftarrow$  PathLossAggregation ( $t_x, PL_{t_x}$ )         $\triangleright O(m)$ 
end for
return  $CoveragePrediction$ 

```

---

value is the same for all  $t_x \in T_x$ . The visibility for an RSG pixel is computed using the *LineOfSight* function, by walking from the antenna of the transmitter to the given pixel element, along the elements intersected by a LOS, until either the visibility is blocked, or the target is reached [45]. Regarding the *PathLoss* function, whenever a receiver point is in NLOS, the walking path from the transmitter has to be inspected for obstacles, calculating the diffraction losses for each of them, i.e.,  $\alpha$  and  $K(d_{(x,y)})$  from Equation (6.2). Hence, its quadratic complexity, which dominates the complexity of the algorithm, together with *LineOfSight*, resulting in an algorithmic complexity denoted by:

$$O(M + n \cdot m^2). \quad (6.4)$$

Although  $n$  will generally be many orders of magnitude smaller than  $m^2$ , its computational-time complexity is relevant for practical use. For example, assuming the radio-coverage prediction for one transmitter completes in around 15 seconds using a serial implementation, the prediction for a mobile network comprising 10,240 transmitters would have an execution time of almost two days.

#### 6.4.5 Design of the parallel version

The focus here is on the practical usability and performance of PRATO. The parallel implementation aims at overcoming the computational-time constraints that prevent a serial implementation from tackling bigger problem instances in a feasible amount of time.

A major drawback of the GRASS as a parallelization environment is that it is not thread-safe, meaning that concurrent changes to the same data set have an undefined behavior [24]. One technique to overcome this problem is to abstract the spatial data from the GRASS. For example, in [82], the authors achieved the GRASS abstraction by introducing a *Point* structure with four *double* attributes, where each pixel of the RSG is mapped to an instance of this structure. Another possibility is for one of the processes, e.g., the master, to read entire rows or columns of data before dispatching them for processing to the workers [6, 82]. In this case, an independence between row/column calculations is required, which is a problem-specific property. Here, abstraction from the GRASS is achieved by loading each spatial-data set into a separate 2D matrix of basic data-type elements, e.g., *float* or *double* depending on the desired accuracy. Each matrix is then assigned the geographical location of the closest corner to the origin of the map-projection system used, e.g., the lower-left corner for the transverse Mercator map projection over central Europe. It follows that the geographical location of any element within the matrix is calculated as the combination of the geographical location of the matrix and the offset of the target element (see Figure 6.5). The advantage of this technique is having the geographical location of a pixel readily available with a minimum memory footprint. Moreover, a convenient consequence of this abstraction schema is that worker processes are completely independent of the GRASS, thus significantly simplifying the deployment of the parallel implementation over multiple computing hosts.

In the area of geographical-information science, the master-worker paradigm has been successfully applied by several authors [5, 6, 27, 71, 82, 156, 157]. However, this technique presents certain issues that prevent the full exploitation of the available computing resources when deployed over several networked computers. Additionally, such issues are difficult to measure when the parallelization involves only one computing node [156, 157] (because no network communication is required), or only a few processes deployed over a handful of nodes [82]. Specifically, the problem refers to network saturation and idle processes within the master-worker model. Generally speaking, a single communicating process, e.g., the master, is usually not able to saturate the network connection of a node. Using more than one MPI process per node might solve this problem, but possible rank-ordering problems

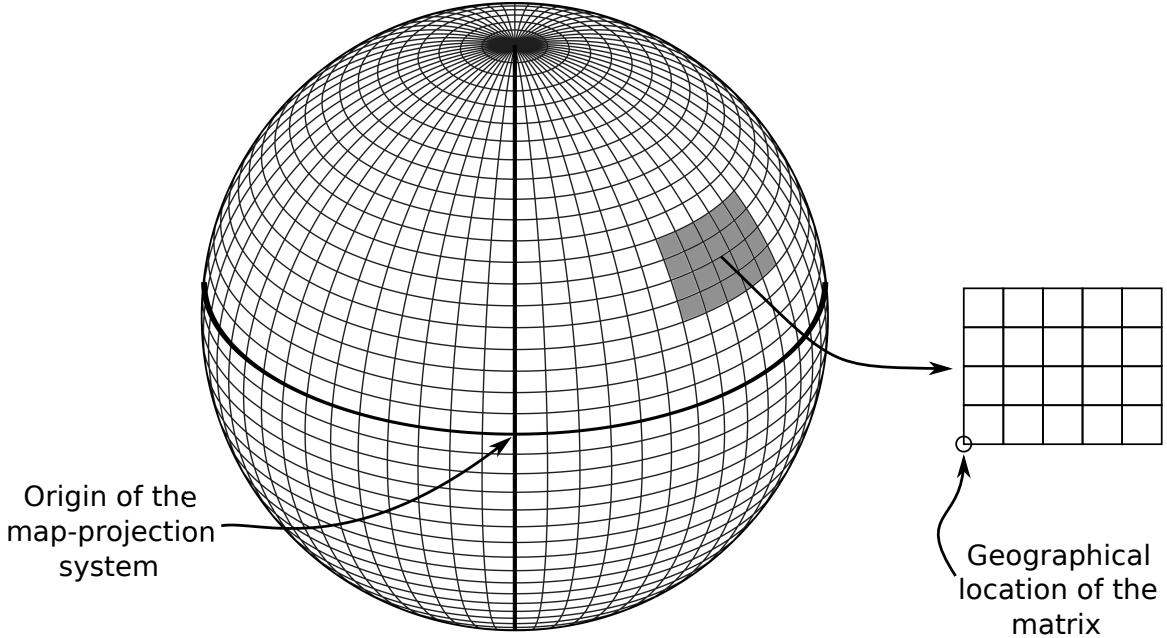


Figure 6.5: Example of a geographical-location mapping of the input-spatial data into a 2D matrix, the lower-left corner of which indicates the nearest point to the origin of the map-projection system.

may appear, thus restricting the full utilization of the network [127]. Another issue appears when the master process executes the MPI code, in which case other processes sleep, making a serial use of the communication component of the system. Consequently, the master process becomes the bottleneck of the parallel implementation as the number of worker processes grows. This situation is also common when dealing with the metadata of a spatial region, which may relate to several elements of a RSG, making it a frequent cause of load imbalance [65, 77, 174]. In PRATO, the transmitter configuration and its antenna diagram represent metadata that are complementary to the sub-region that a transmitter covers.

Hybrid MPI-OpenMP implementations [156, 157], in which no MPI calls are issued inside the OpenMP-parallel regions, also fail to saturate the network [127]. A possible solution to this problem is to improve the communication overlap among the processes. To this end, PRATO features non-blocking point-to-point MPI operations, and an independent thread in the worker process that saves the intermediate results into a DB. One such database system per computer cluster is used, which also serves the input data to the GRASS, in order to aggregate the partial results of the path-loss predictions or to visualize them. It is important to note that any kind of DB may be used, e.g., relational, distributed [121] or even those of the NoSQL type [152]. Nevertheless, a central, relational-database system is used here, since they are the most popular and widely available ones. Additionally, the non-blocking message-passing technique used to distribute the work-load among the nodes provides support for heterogeneous environments. As a result, computing nodes featuring more capable hardware receive more work than those with weaker configurations, thus ensuring a better utilization of the available computing resources despite hardware diversity, and improved load balancing.

#### 6.4.5.1 Master process

The master process, the flow diagram of which is given in Figure 6.6, is the only component that runs within the GRASS environment. As soon as the master process starts, the input parameters are read. This step corresponds to “Read input data” in Figure 6.6, and it is

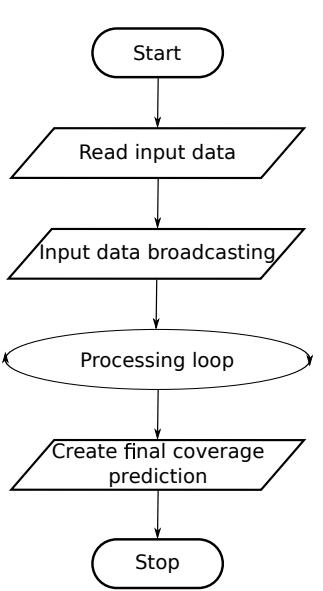


Figure 6.6: Flow diagram of the master process.

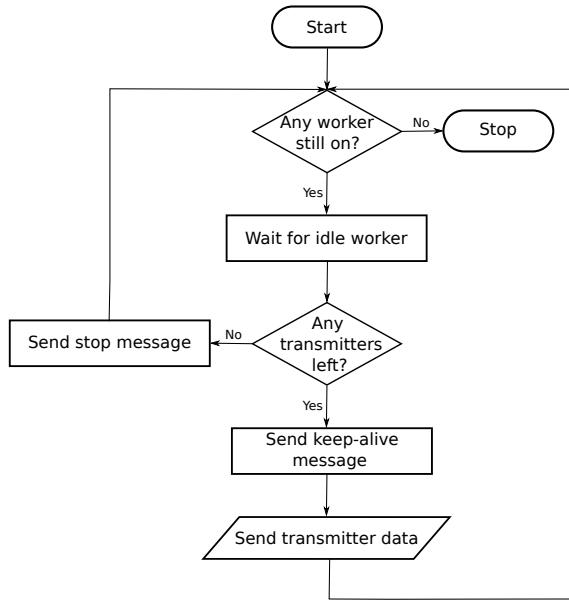


Figure 6.7: Flow diagram of the “Processing loop” step of the master process.

carried out in a similar way as in the serial version. The next step delivers the metadata that is common to all the transmitters to all the processes (see “Metadata broadcasting” in Figure 6.6). Before distributing the work among the worker processes, the master process proceeds to decompose the loaded raster data into 2D matrices of basic-data-type elements, e.g., *float* or *double*, before dispatching them to the multiple worker processes. In this case, the decomposition applies to the DEM and the clutter data only, but it could be applied to any point-based data set. In the next step, the master process starts an asynchronous message-driven processing loop (see “Processing loop” in Figure 6.6), the main task of which is to assign and distribute the sub-region and configuration data of different transmitters among the idle worker processes.

The flow diagram shown in Figure 6.7 illustrates the “Processing loop” step of the master process. In the processing loop, the master process starts by checking the available worker processes, that might calculate the radio-coverage prediction for the next transmitter. It is worth pointing out that this step also serves as a stopping condition for the processing loop itself (see “Any worker still on?” in Figure 6.7). The active worker processes inform the master process that they are ready to compute by sending an idle message (see “Wait for idle worker” in Figure 6.7). The master process then announces to the idle worker process that it is about to receive new data for the next calculation, and it dispatches the complete configuration of the transmitter to be processed (see “Send keep-alive message” and “Send transmitter data” steps, respectively, in Figure 6.7). This is only done in the case that there are transmitters for which the coverage prediction has yet to be calculated (see “Any transmitters left?” in Figure 6.7). The processing loop of the master process continues distributing the transmitter data among the worker processes, which asynchronously become idle as they finish the radio-prediction calculations they have been assigned by the master process. When there are no more transmitters left, all the worker processes announcing they are idle will receive a shutdown message from the master process, indicating to them that they should stop running (see “Send stop message” in Figure 6.7). The master process will keep doing this until all the worker processes have finished (see “Any worker still on?” in Figure 6.7), thus fulfilling the stopping condition of the processing loop.

Finally, the last step of the master process is devoted to creating the final output of the calculation, e.g., a raster map (see “Create final coverage prediction” in Figure 6.6). The final coverage prediction of all the transmitters is an aggregation from the individual path-loss results created by each of the worker processes during the “Processing loop” phase in Figure 6.6, which provides the source data for the final raster map. The aggregation of the individual path-loss results is accomplished by issuing an SQL query over the database tables containing them, in a similar way as in the serial version.

#### 6.4.5.2 Worker processes on CPU

An essential characteristic of the worker processes is that they are completely independent of the GRASS, i.e., they do not have to run within the GRASS environment nor use any of the GRASS libraries to work. This aspect significantly simplifies the deployment phase to run PRATO on a computer cluster, since no GRASS installation is needed on the computing nodes hosting the worker processes.

One possibility to overcome the thread-safety limitation of the GRASS is to save the transmitter path-loss predictions through the master process, thus avoiding concurrent access. However, for the workers to send intermediate results back to the master process, e.g., as in [5, 82], is a major bottleneck for the scalability of a parallel implementation. In such case, the scalability is limited by the master process, because it must serially process the received results in order to avoid inconsistencies due to concurrent access. Instead, PRATO allows each of the worker processes to output its intermediate results into a DB, i.e., each path-loss prediction in its own table. Additionally, worker processes do this from an independent thread, which runs concurrently with the calculation of the next transmitter received from the master process. In this way, the overlap between the calculation and communication significantly hides the latency created by the result-dumping task, thus making better use of the available system resources.

The computations of the worker processes, the flow diagram of which is given in Figure 6.8, begin by receiving metadata about the transmitters and the geographical area from the master process during the initialization time (see “Receive broadcasted metadata” in Figure 6.8).

After the broadcasted metadata are received by all the worker processes, each one proceeds to inform the master process that it is ready (i.e., in an idle state) to receive the transmitter-configuration data that define which transmitter path-loss prediction to perform (see “Send idle message” in Figure 6.8). If the master process does not give the instruction to stop processing (see “Has stop message arrived?” in Figure 6.8), the worker process collects the sub-region spatial data and the transmitter configuration (see “Receive transmitter data” in Figure 6.8). In the event that a stop message is received, the worker process will wait for any result-dumping thread to finish (see “Wait for result-dump thread” in Figure 6.8) before shutting down. The coverage calculation itself follows a similar design as the serial version (see “Coverage calculation” in Figure 6.8).

As mentioned before, the worker process launches an independent thread to save the path-loss prediction of the target transmitter into a DB table (see “Threaded save path-loss to DB” in Figure 6.8). It is important to note that there is no possibility of data inconsistency due to the saving task being executed inside a thread, since path-loss data from different workers belong to different transmitters and are, at this point of the process, mutually exclusive.

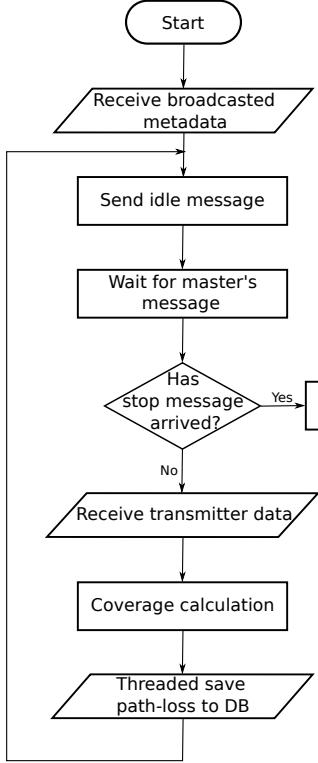


Figure 6.8: Flow diagram of a worker process.

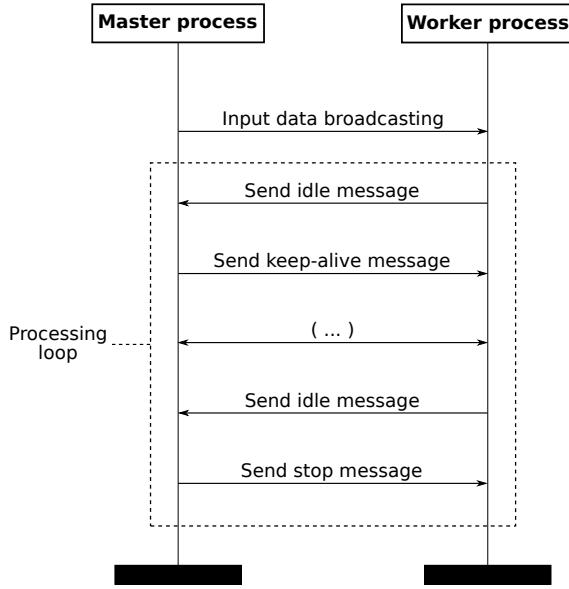


Figure 6.9: Communication diagram, showing the message passing between the master and a worker process.

#### 6.4.5.3 Worker processes on GPU

PRATO provides multi-GPU support for improving the execution performance on the computing nodes hosting the worker processes. The algorithmic adaptation from a CPU to a GPU is not a trivial task. This section focuses on the main modifications made to the radio-propagation algorithm in order for it to work on GPU hardware.

It is well known that the bandwidth of the PCI Express bus can cause a throughput bottleneck when a significant amount of data is transferred between a CPU and a GPU in a heterogeneous system [69]. Some researchers acknowledged that unless a full working set of data can fit into the memory on a GPU, the PCI Express will become the bottleneck of the system [120, 136]. For this reason, it is imperative to have as much data as possible allocated on the GPU itself.

In order to minimize the CPU-to-GPU memory transfers, the spatial data used by the radio-propagation algorithm was organized as explained in Section 6.4.5, i.e., using geographically-located, offset-based 2D matrices. However, the internal representation of the matrix elements was changed to use less memory. To this end, the clutter-matrix elements were represented as *unsigned char*, since they express signal loss in dB. It follows that for the radio-propagation prediction of one transmitter, the following matrices should be allocated on the GPU:

- one 2D matrix containing DEM data for the target subregion, the elements of which are *float* or *double*;
- one 2D matrix containing clutter data for the target subregion, the elements of which are *unsigned char*; and

- one 2D matrix containing the resulting path-loss prediction.

The dimension of all matrices is based on the transmission radius, within which the radio-propagation prediction should be calculated. The contents of the DEM and clutter matrices is constant throughout the calculation process. For this reason they were saved into read-only texture memory to take advantage of the faster access time (see Section 5.1, Chapter 5). Regarding the resulting path-loss matrix, each step of the radio-prediction algorithm is applied over the results of the previous step (see Figure 6.1 for a flow diagram of the steps involved), thus avoiding extra allocation on global memory or a data-transfer from/to the CPU.

#### 6.4.5.4 Master-worker communication

Similar to [156, 157], the message-passing technique used in this work enables a better use of the available computing resources, both in terms of scalability and load balancing, while introducing a negligible overhead. This last point is supported by the experimental results, introduced in Section 6.5.3.

The first reason to implement the message-passing technique is to support heterogeneous computing environments. In particular, our approach focuses on taking full advantage of the hardware of each computing node, thus explicitly avoiding the bottlenecks introduced by the slowest computing node in the cluster. This problem appears when evenly distributing the data among the worker processes on disparate hardware, e.g., as in [6, 82], being more noticeable with a larger number of computing nodes and processes. In other words, computing nodes that deliver better performance have more calculations assigned to them. Moreover, in real-world scenarios, it is often the case that a large number of dedicated computing nodes featuring exactly the same configuration is difficult to find, i.e., not every organization owns a computer cluster.

A second reason for selecting a message-passing technique is related to the flexibility it provides for load balancing, which is of greater importance when dealing with extra data or information besides spatial data [77]. This can be seen in Figure 6.7, where the master process, before delivering the spatial subset and transmitter-configuration data, sends a message to the worker process, indicating that it is about to receive more work. This a priori meaningless message plays a key role in correctly supporting the asynchronous process communication. Notice that the subset of spatial data that a worker process receives is directly related to the transmitter for which the prediction will be calculated. Similar to [156, 157], this problem-specific property enables the use of a data-decomposition technique based on a block partition of spatial data, e.g., the DEM and clutter data.

In general, there are many different ways a parallel program can be executed, because the steps from the different processes can be interleaved in various ways and a process can make non-deterministic choices [139], which may lead to situations such as race conditions [37] and deadlocks. A deadlock occurs whenever two or more running processes are waiting for each other to finish, and thus neither ever does. To prevent PRATO from deadlocking, message sending and receiving should be paired, i.e., an equal number of send and receive messages on the master and worker sides [139]. Figure 6.9 depicts the master-worker message passing, from which the transmitter-data transmission has been excluded for clarity. Notice how each idle message sent from the worker process is paired with an answer from the master process, whether it is a keep-alive or a stop message.

## 6.5 Simulations

Considering the large computational power needed for predicting the radio-coverage of a real radio network, the use of a computer cluster is recommended. A computer cluster is a group of interconnected computers that work together as a single system. Computer clusters typically consist of several commodity PCs connected through a high-speed local-area network (LAN) with a distributed file system, like NFS [138]. One such system is the DEGIMA cluster [72] at the Nagasaki Advanced Computing Center (NACC) of the Nagasaki University in Japan. This system ranked in the TOP 500 list of supercomputers until June 2012<sup>1</sup>, and in June 2011 it held the third place in the Green 500 list<sup>2</sup> as one of the most energy-efficient supercomputers in the world.

This section presents the simulations, and an exhaustive analysis of the performance and scalability of the parallel implementation of PRATO. The most common usage case for PRATO is to perform a radio-coverage prediction for multiple transmitters. Therefore, a straight-forward parallel decomposition is to divide a given problem instance by transmitter, for which each coverage prediction is calculated by a separate worker process.

The following simulations were carried out on 34 computing nodes of the DEGIMA cluster. The computing nodes are connected by a LAN, over a Gigabit Ethernet interconnect. As mentioned before, the reason for using a high-end computer cluster such as DEGIMA is to explore, by experimentation, the advantages and drawbacks of the considered methods. However, this does not imply any loss of generality if applying these principles over a different group of networked computers that do not operate as a computer cluster. Moreover, PRATO also supports parallel calculation of radio-propagation predictions for multiple cells by distributing the processes among the individual cores of a single CPU.

Each computing node of DEGIMA features one of two possible configurations, namely:

- Intel Core i5-2500T quad-core processor CPU, clocked at 2.30 GHz, with 16 GB of RAM; and
- Intel Core i7-2600K quad-core processor CPU, clocked at 3.40 GHz, also with 16 GB of RAM.

During the simulation runs, the nodes equipped with the Intel i5 CPU hosted the worker processes, whereas the master process and the PostgreSQL database server (version 9.1.4) each run on a different computing node, featuring an Intel i7 CPU. The DB server performed all its I/O operations on the local file system, which was mounted on an 8 GB RAM disk. During the simulations, the path-loss predictions of 5,120 transmitters occupied less than 4 GB of this partition. No GPU hardware was used for the following simulation sets.

A 64-bit Linux operating system (Fedora distribution) was the operating system used. The message-passing implementation OpenMPI, version 1.6.1, was manually compiled with the distribution-supplied gcc compiler, version 4.4.4.

### 6.5.1 Test networks

The parallel performance of PRATO was tested with real radio networks of different sizes. In order to create the synthetic test data sets with an arbitrary number of transmitters, a group of 2,000 transmitters of a real network was used. The configuration parameters of these transmitters resembled those of the LTE network deployed in Slovenia by Telekom Slovenije, d.d., which were, in turn, randomly replicated and distributed over the whole

---

<sup>1</sup><http://www.top500.org>

<sup>2</sup><http://www.green500.org>

target area. The path-loss predictions were calculated using the radio-propagation model introduced in Section 6.3.2. The DEM area, as well as the clutter data, covered 20,270 km<sup>2</sup> with a pixel resolution of 25 m<sup>2</sup>. The clutter data contained different levels of signal loss due to land usage. For all the points within a radius of 20 km around each transmitter, a receiver positioned 1.5 m above the ground was assumed, and the frequency was set to 1,843 MHz.

### 6.5.2 Weak scalability

The weak-scalability experiments are meant to analyze the scalability of the parallel implementation in cases where the workload assigned to each process (one MPI process per processor core) remains constant as the number of processor cores is increased. It follows that the number of transmitters deployed over the target area is directly proportional to the number of processor cores and worker processes. This was accomplished by assigning a constant number of transmitters per core, while increasing the number of cores hosting the worker processes. Here, the following numbers of transmitters per worker/core were tested {5,10,20,40,80}, by progressively doubling the number of worker processes from 1 to 64.

Problems that are particularly well-suited for parallel computing exhibit computational costs that are linearly dependent on the size of the problem. This property, also referred to as algorithmic scalability, means that proportionally increasing both the problem size and the number of cores results in a roughly constant time to solution.

The master-worker (MW) configuration performs result aggregation continuously, i.e., while receiving the intermediate results from the worker processes. In contrast, the master-worker-DB (MWD) setup performs the result aggregation as the final step. This set of experiments is meant to investigate how the proposed MWD technique compares with the classic MW approach in terms of scalability when dealing with a constant computational load per core.

An important fact about the presented simulations when using multi-threaded implementations is to avoid oversubscribing a computing node. For example, if deploying four worker processes over a quad-core CPU, the extra threads will have a counter effect on the parallel efficiency, since the CPU resources would be exhausted, which slows the whole process down. For this reason, we have deployed three worker processes per computing node, leaving one core free for executing the extra threads.

#### 6.5.2.1 Results

The results represent the best running time out of a set of 20 independent simulation runs, for which the transmitters and the rank ordering of the worker processes were randomly selected. The collected running times for the weak-scalability experiments are shown in Figure 6.10. All the measurements express wall-clock times in seconds for each setup and problem instance, defined as the number of transmitters per process (Tx/process). The wall-clock time represents the real time that elapses from the start of the master process to its end, including the time that passes while waiting for the resources to become available. The running-time improvements of the MWD versus the MW setup are shown in Table 6.1.

The time measurements observed from the weak-scalability results show that the classic MW approach performs well for up to four worker processes. When using eight worker processes, the MW setup is practically equivalent to the MWD approach, indicating that the master process is being fully exploited. When increasing the problem size and the number of worker processes to 16, the running-time gain is already clear, favoring the MWD configuration. This gain keeps growing, although slower, as we increase the number of worker processes to 32 and 64, confirming the hypothesis that in a classic MW approach, the parallel efficiency is bounded by the capacity of the master process to serve an increasing number of

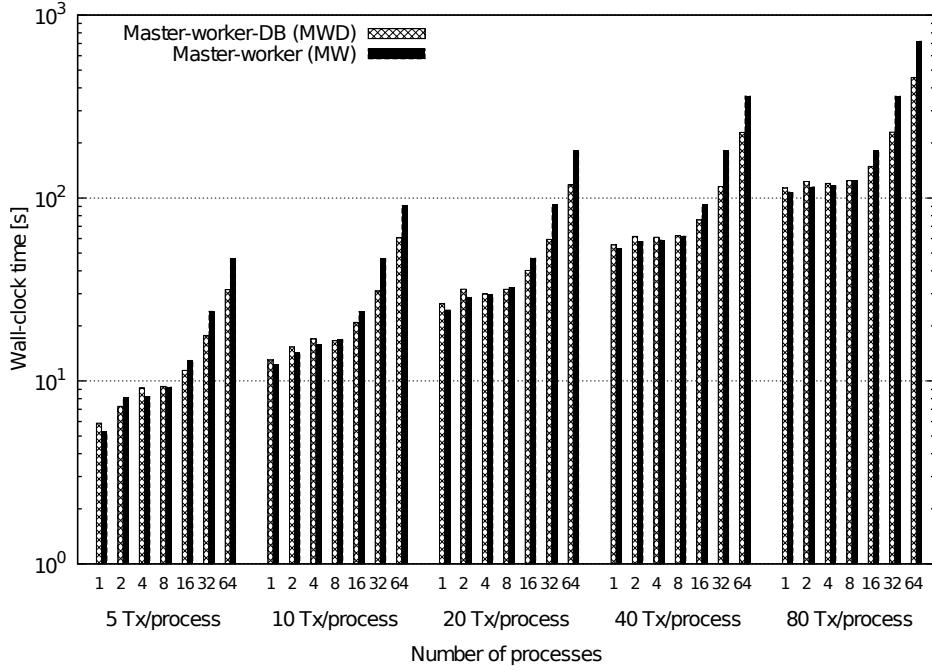


Figure 6.10: Measured wall-clock time for weak-scalability experiments, featuring MW and MWD setups. Experiments allocated one MPI worker process per core. The wall-clock time axis is expressed in a base-10 logarithmic scale, whereas the axis representing the number of cores is expressed in a base-2 logarithmic scale.

Table 6.1: Running-time gain (in percent) of the simulations for the weak-scalability of the MWD setup relative to the classic MW approach.

TX/core	Number of cores						
	1	2	4	8	16	32	64
5	-11.39	-10.42	-11.14	-0.95	11.75	26.15	32.53
10	-5.84	-7.78	-7.67	0.91	12.81	33.28	33.55
20	-8.59	-10.88	-1.04	1.95	14.29	35.23	35.27
40	-5.26	-6.90	-3.68	-0.67	17.27	36.23	36.65
80	-5.29	-7.11	-3.20	-0.31	17.94	36.32	36.57

worker processes. Interestingly, the gain when using 32 and 64 worker processes is almost the same. After further investigation, the reason for this behavior was found: the new bottleneck was the LAN being completely saturated by the worker processes. Consequently, they have to wait for the network resources to become available before sending or receiving data, which is not the case when running the MW setup. Therefore, using the MWD approach a hardware constraint is hit, meaning that the bottleneck is no longer at the implementation level. Moreover, since the master process is far from overloaded when serving 64 worker processes, it can be expected that the MWD approach will keep scaling if a faster network infrastructure is used, e.g., 10-gigabit Ethernet or InfiniBand.

Certainly, the parallel version of PRATO scales better using the MWD approach, when challenged with a large number of transmitters (5,120 for the biggest instance) over 64 cores. This fact shows PRATO would be able to calculate the radio-coverage prediction for real networks in a feasible amount of time, since many operational radio networks have already deployed a comparable number of transmitters, e.g., the 3G network within the Greater London Authority area, in the UK [118]. For a more in-depth discussion and experimentation about real-world planning scenarios, see Chapter 10.

Not being able to achieve perfect weak scalability using the MWD setup is due to a number of factors. Specifically, the overhead time of the serial sections of the parallel process grows proportionally with the number of cores, e.g., aggregation of the intermediate results, although the total contribution of this overhead remains low for large problem sizes. Moreover, the communication overhead grows linearly with the number of cores used. Consequently, the findings of Huang et al. [82] can be confirmed, who concluded that the data-set size should be large enough for the communication overhead to be hidden by the calculation time. This ensures profitable parallelization in terms of running-time reduction.

### 6.5.3 Strong scalability

This set of simulations is meant to analyze the impact of increasing the number of computing cores for a given problem size, i.e., the number of transmitters deployed over the target area does not change, but only the number of worker processes used is increased. Here, the following number of transmitters were tested  $\{1,280, 2,560, 5,120\}$ , by gradually doubling the number of workers from 1 to 64 for each problem size.

#### 6.5.3.1 Results

Similar to the weak-scalability experiments, these time measurements show that when applying a classic MW approach, the running-time reduction starts flattening when more than eight worker processes were used. Moreover, the running times for 16, 32 and 64 worker processes are the same, i.e., they do not improve due to the master process being saturated. In contrast, when using the proposed MWD technique, the running-time reduction improves for up to 32 worker processes, after which there is no further improvement since the network was being fully exploited. These results clearly show that when applying parallelization using a larger number of worker processes, the master process becomes the bottleneck of the MW approach. When using the MWD configuration, a steady running-time reduction is observed, until a hardware constraint is hit, e.g., the network infrastructure.

The overhead of sending/receiving asynchronous messages in order to support heterogeneous systems was also measured. It was found that this overhead never exceeds 0.02% of the total running time for the MW experiments, and 0.01% for the MWD experimental set.

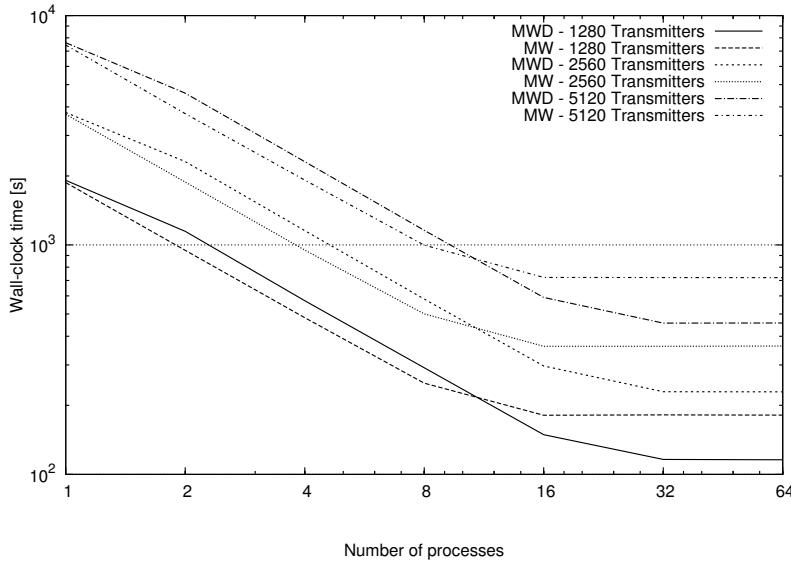


Figure 6.11: Measured wall-clock time for strong-scalability experiments, featuring MW and MWD setups. Experiments assigned one MPI worker process per core. The wall-clock time axis is expressed in a base-10 logarithmic scale, whereas the axis representing the number of cores is expressed in a base-2 logarithmic scale.

### 6.5.3.2 Speedup

In order to further analyze how well the PRATO scales using the MW and MWD approaches, the performance of the parallel implementation in terms of its speedup was measured, which is defined as:

$$S(NP) = \frac{\text{execution time for base case}}{\text{execution time for } NP \text{ cores}}, \quad (6.5)$$

where  $NP$  is the number of cores executing the worker processes. The parallel implementation running on only one core was the base case for comparisons. The serial implementation is not a good base comparison for the parallel results as it does not reuse the resources between each transmitter-coverage calculation and it does not overlap the I/O operations with the transmitter computations. In practice, this means that several concatenated runs of the serial version would be considerably slower than the single-worker configuration.

Using the speedup metric, linear scaling is achieved when the obtained speedup is equal to the total number of processors used. However, it should be noted that a perfect speedup is almost never achieved, due to the existence of serial stages within an algorithm and the communication overhead of the parallel implementation.

Figure 6.12 shows the average speedup of the parallel implementation for up to 64 worker processes, using the standard MW method and the proposed MWD approach. The average speedup was calculated for the three different problem instances, i.e., 1,280, 2,560, and 5,120 transmitters deployed over the target area. The number of transmitters used in these problem sizes is comparable to several real-world radio networks that were already deployed in England, e.g., Hampshire County with 227 BSs, West Midlands with 414 BSs, and Greater London Authority with 1,086 BSs [118]. Note that it is common for a single base station to host multiple transmitters.

The plotted average speedup clearly shows the minimal overhead of the MWD approach when using a small number of worker processes. This overhead accounts for the final aggregation of the intermediate results at the DB, which in the MW configuration is performed

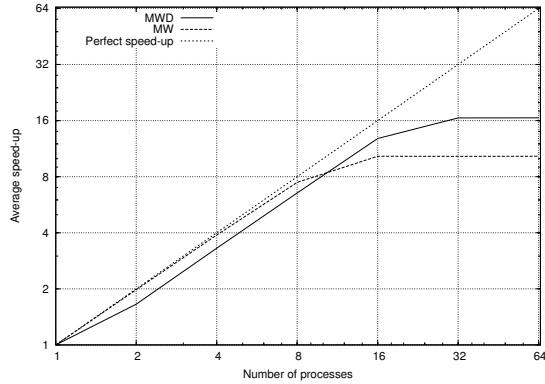


Figure 6.12: Average speedup for strong-scalability experiments. Both axes are expressed in a base-2 logarithmic scale.

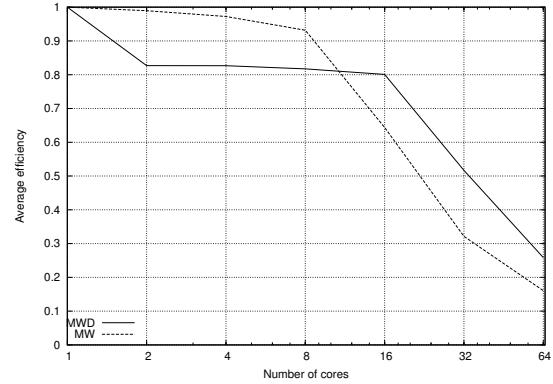


Figure 6.13: Average parallel efficiency for strong-scalability experiments. The parallel-efficiency axis is expressed in a linear scale, whereas the axis representing the number of cores is expressed in a base-2 logarithmic scale.

along worker processing. Like before, the DB component allows the parallel implementation to fully exploit the available computing resources when deploying a larger number of worker processes, until the network-capacity limit is met. Of course, these results are directly correlated with the wall-clock times shown in Figure 6.11.

#### 6.5.3.3 Efficiency

Another measure to study how well PRATO utilizes the available computing resources considers the parallel efficiency of the implementation. The definition of parallel efficiency is as follows:

$$E(NP) = \frac{S(NP)}{NP}, \quad (6.6)$$

where  $S(NP)$  is the speedup as defined in Equation (6.5), and  $NP$  is the number of cores executing worker processes. Figure 6.13 shows the average parallel efficiency of the parallel implementation for different problem sizes, as the number of processing cores was increased. Like for the speedup measure, the average parallel efficiency from the same problem instances was calculated.

The ideal case for a parallel application would be to utilize all the available computing resources, in which case the parallel efficiency would always be equal to one as the core count increases. From the plot in Figure 6.13, it can be observed that the efficiency of the MWD approach is better than in the MW case for larger number of processes, and as long as there was available capacity at the LAN level. In accordance to the previous analysis, the under utilization of the computing resources is more significant when the master process is overloaded (in the MW case) than when the network infrastructure is saturated (in the MWD case). The lower efficiency is directly proportional to the number of idle worker processes that wait either for the master process (MW case) or for network access (MWD case).

Overall, the experimental results confirm that the objective of fully exploiting the available hardware resources is accomplished when applying the presented MWD approach, thus improving the scalability and efficiency of PRATO when compared to a traditional MW technique.

#### 6.5.4 GPU performance

<Experiments not yet finished ...???>

## 6.6 Summary

PRATO, a parallel radio-coverage prediction tool for radio networks, has been presented in this chapter. The tool is intended to be used for radio-network planning analysis and decision support. Its high-performance capabilities make it ideal for automatic-optimization tasks that require a large number of evaluations.

The parallel implementation of PRATO includes a novel parallel technique for master-worker configurations. The introduced MWD technique, which combines the use of a DB system with a work-pool approach, delivers improved performance when compared with a traditional MW setup. Moreover, the presented system provides parallel and asynchronous computation that is completely independent of the GIS used, in this case the GRASS environment. Consequently, a GIS installation is only needed on the master node, thus simplifying the required system setup and greatly enhancing the applicability of this methodology in different environments.

The extensive simulations, performed on the DEGIMA cluster of the Nagasaki Advanced Computing Center, were analyzed to determine the level of scalability of the implementation, as well as the impact of the presented methods for parallel-algorithm design aimed at spatial-data processing. The conducted analyses show that when using the MWD approach, PRATO is able to calculate the radio-coverage prediction of real-world radio networks in a reduced amount of time. Moreover, the experimental results show that PRATO has a better scalability when using the MWD approach than the standard MW setup, since it is able to completely saturate the network infrastructure of the computer cluster. These promising results also show the great potential of the MWD approach for parallelizing different time-consuming tasks dealing with spatial data, where DBs form an intrinsic part of almost all GIS. Furthermore, the automatic optimization of radio networks, where millions of radio-propagation predictions take part in the evaluation step of the optimization process, is also an excellent use case for PRATO. Indeed, this last point will be further discussed and validated in the following chapters.

The performance of the worker processes has been additionally improved by including the implementation of the radio-propagation algorithm on GPU. The use of GPU hardware is optional, i.e., it is exploited only if it is available on the computing nodes that host the worker processes. The experimental simulations showed a significant speedup due to the GPU implementation, when compared to the CPU-only version.

To the best of the author's knowledge, neither the MWD parallel technique nor the parallel implementation of the radio-prediction algorithm as presented in this chapter, have yet been described in the related literature.



## 7 The service-coverage problem

The high-performance of PRATO, the radio-coverage simulation framework presented in Chapter 6, allows dealing with big problem instances in a reduced amount of time. Additionally, it enables tackling optimization problems that, because of their size, are out-of-reach of traditional approaches, mainly due to the computational-time complexity of their objective-function evaluation.

In this chapter, the challenge is to exploit PRATO for solving one of the classic optimization problems of radio networks: the service-coverage problem. Considering the minimization of the total amount of pilot power subject to a full coverage constraint, a novel optimization approach is introduced. The presented method, based on parallel autonomous agents, gives very good solutions to the problem in an acceptable amount of time. The parallel implementation takes full advantage of GPU hardware in order to achieve considerable speed-up. The analysis of the experimental results, considering six real-world, radio networks of different sizes, studies solution-quality and performance aspects.

The content of this chapter extends the research work published by the author in [18] and [20]. The rest of this chapter is organized as follows. Section 7.1 gives a description of the coverage problem and its motivation from the mobile operator's perspective. In Section 7.2, a short overview of related research works is given, before formally introducing the key elements of the service-coverage problem in Section 7.3. The parallel-agent approach, as well as the strategies used for result comparison, are presented in Section 7.5, followed by the simulations and their analyses in Section 7.6.

## 7.1 Motivation

Solving the service-coverage problem for radio networks has received a great deal of attention in the past years. Its complexity demands the confluence of different skills in areas such as propagation of radio signals, telecommunications and information systems, among others.

Even several decades after the launch of the first commercial GSM network, service-coverage planning remains a key problem that all mobile operators have to deal with. Its intricacy arises from the wide range of different combinations of configuration parameters and their evaluation-time complexity. One crucial parameter, which is mainly subject of adjustment, is the transmit power of the pilot signal (see Section 3.2, Chapter 3).

Regardless of the mobile technology used, e.g., GSM, UMTS or LTE, a lower transmit power generates less interference, which, in turn, translates into more capacity of the radio link (see Chapter 3). Moreover, reducing pilot-power usage is also related to issues regarding the human exposure to the electromagnetic fields generated by BS antennas [50]. During the past few years, public opinion has been extremely sensitive regarding this issue, and thus many countries have already imposed safety standards to limit the electromagnetic field levels produced by antennas in a given range.

From the UMTS perspective, minimizing pilot-power usage leaves more power available for increased network capacity. This is especially important if the traffic and other channels are configured relative to the pilot channel [79]. Moreover, as the demand for internet access and data services increases [41], so does the pressure on existing network infrastructure, making parameter optimization the only viable solution in the short-term [110].

The idea of using autonomous agents for optimization is not new. It has proven to be a solid optimization approach for solving different types of problems, not only within the area of radio networks [34, 50], but also in other fields [166, 169]. The increased computational-time complexity when dealing with big problem instances is tackled using a parallel, agent-based algorithm on GPU. This minimizes the overhead when deploying a larger number of agents working in parallel over the service area, only limited by the amount of memory available.

## 7.2 Related work

There are several approaches in the literature that address the service-coverage problem in radio networks [9, 110, 142]. Some of them even claim to achieve near-optimal solutions [148]. As a matter of fact, most formulations are only useful for small network instances and often fail when challenged with larger, real-world networks.

A genetic-algorithm approach for solving the service-coverage problem for GSM networks was presented in [100]. The proposed solution is based on the physical distribution of BSs in order to maximize coverage. The simulations were performed on a test network with 40 candidate sites for BS antennas.

In [148], Siomina and Yuan considered the problem of minimizing the total amount of pilot power for UMTS networks, subject to a full coverage constraint. They tackled the problem with an iterative linear-programming approach, reporting very good results for some test networks, containing from 15 to 65 base stations. The authors noted that bigger problem instances could not be solved because of hardware constraints on the target platform.

As for LTE networks, the service-coverage problem was addressed in [162]. The authors presented an algorithm, based on reinforcement learning, to tackle three aspects of the coverage problem, i.e., coverage holes, weak coverage and pilot pollution. The experimental simulations, performed on 3 BSs, used different antenna-tilt configurations as the proposed solutions.

The service-coverage problem, as presented in this chapter, corresponds to achieving full coverage of the target area, without coverage holes.

## 7.3 Radio-network model

Extending the representation of a radio-network model from [112], this section presents the definitions of all the elements included in the mathematical model used for the simulations.

The goal here is to analyze the state of the network in a given situation, i.e., a ‘snapshot’ at an arbitrary instance. A snapshot consists of a set of UEs having individual properties, such as location, and equipment type. The static approach inherently ignores dynamic effects that influence the system, like fast power control.

For additional information regarding mathematical models of comparable problems, see [110].

### 7.3.1 Basic elements

Consider a UMTS network with a set of antenna installations (cells),  $C$ . A RSG of a given resolution represents a geographical area,  $A_{\text{total}}$ , within which a set of UEs,  $M$ , is spatially distributed over the pixels of  $A_{\text{total}}$ . Further,  $L_{cm}^{\downarrow}$  is defined as the downlink attenuation factor between cell  $c \in C$  and UE  $m \in M$ . Similarly,  $L_{mc}^{\uparrow}$  represents the uplink attenuation factor between UE  $m$  and cell  $c$ . The attenuation factor values are calculated by performing signal-propagation predictions for every pair  $(c, m)$ ,  $c \in C$ ,  $m \in M$ , using the radio-propagation model introduced in Section 6.3.2, Chapter 6. These predictions already include losses and gains from cabling, hardware, and user equipment.

The amount of power allocated to the pilot signal of cell  $c$  is denoted as  $p_c$ , and it can adopt any value from the sorted set of available pilot power levels,  $P_c = \{p_c^1, p_c^2, \dots, p_c^k\}$ , where  $p_c^k$  is the maximum power.

Based on the introduced elements, the received pilot power from cell  $c$  to UE  $m$  is  $L_{cm}^{\downarrow} p_c$ .

### 7.3.2 Coverage

An UE  $m$  within the area  $A_{\text{covered}}$  is under service coverage if at least one cell  $c$  covers it. Cell coverage is provided to an UE  $m$  from a cell  $c$  if its signal-to-interference ratio,  $sir(c, m)$ , at the RSG pixel where  $m$  is located, is not lower than a given threshold,  $\gamma^{\text{cov}}$ :

$$sir(c, m) = \frac{L_{cm}^{\downarrow} p_c}{\tau_0 + \sum_{i \in C} L_{im}^{\downarrow} p_i} \geq \gamma^{\text{cov}}, \quad (7.1)$$

where  $\tau_0$  is the thermal noise. For convenience, a binary function is defined to determine the coverage of a UE  $m$  by a cell  $c$ . So, for any pair  $(c, m)$ ,  $c \in C$ ,  $m \in M$ , the coverage of UE  $m$  by cell  $c$  is defined as:

$$cov(c, m) = \begin{cases} 1 & \text{if } sir(c, m) \geq \gamma^{\text{cov}} \\ 0 & \text{otherwise} \end{cases}. \quad (7.2)$$

A set, denoted as  $C_m$ ,  $C_m \subset C$ , contains all the cells covering an UE  $m$ . From this set, the cell with the highest  $sir(c, m)$  is referred to as the best server, and denoted as  $c_m^*$ .

Notice that the described radio-network model is easily adaptable for different mobile technologies, e.g., GSM, UMTS and LTE. For example, if solving the service-coverage problem for UMTS, it would be reasonable to assume that all cells in the network operate at maximum power, and adapt Equation (7.1) accordingly. This is, from the interference point of view,

the worst-case scenario [30, 148]. This assumption guarantees, that even under heavy user traffic, full coverage of the service area is maintained, due to the cell-breathing principle [79].

## 7.4 Problem definition

In the problem of optimization of pilot powers for service coverage, the objective is to find a set of pilot-power settings for all cells in the network, such that the total pilot power used is minimized, and a given service coverage criteria is fulfilled. In other words, solving the service-coverage problem corresponds to finding the pilot power levels  $p_c$ , for all cells  $c \in C$ , such that coverage of at least  $b$  UEs is guaranteed, while the total amount of pilot power used is minimized. Here, full coverage of the service area is being considered, thus  $b = |M|$ . Consequently, the optimization objective is defined as follows:

$$P^* = \min \sum_{c \in C} p_c, \quad (7.3)$$

subject to

$$\frac{\sum_{m \in M} cov(c_m^*, m)}{b} = 1. \quad (7.4)$$

It has been proved that the problem of pilot-power optimization for full coverage of the service area is *NP*-hard, since it can be reduced to the set-covering problem [168]. Consequently, as long as  $P \neq NP$ , it is unfeasible that a polynomial-time algorithm exists, which is able to find an exact solution to this problem.

## 7.5 Optimization approaches

Since some of the analyzed problem instances are part of a real mobile network deployed in Slovenia by Telekom Slovenije, d.d., there are no references in the literature of other optimization techniques dealing with exactly the same data set. For this reason, two different strategies for setting the pilot power are being presented. They should provide a basis for the comparison of the experimental results. The first strategy is the attenuation-based pilot power, presented in [142], in which a pixel of the service area is always covered by the cell with the maximum attenuation-factor value, i.e., the minimum path loss. The second strategy is the presented parallel-agent approach, based on ideas inspired by two-dimensional cellular automata [135] and metaheuristics [159]. A detailed description is given in Section 7.5.2.

Similar criteria for result comparison have also been used in [142, 148].

### 7.5.1 Attenuation-based approach

The first heuristic for setting the pilot power of all cells in the network is known as attenuation-based, since it relies on the downlink-attenuation factor,  $L_{cm}^\downarrow$ . An UE located on some pixel of the service area is always covered by the cell with the maximum  $L_{cm}^\downarrow$ . Whenever the maximum available power,  $p_c^k$ , is the same for all the cells in the network, this is equivalent to selecting the cell with the minimum required pilot power to cover a UE  $m$ . Hence, under this assumption, the cell  $c$  covering UE  $m$  is identified as:

$$p_{cm}^{\text{att}} = \min p_c \forall c \in C \iff cov(c, m) = 1 \quad (7.5)$$

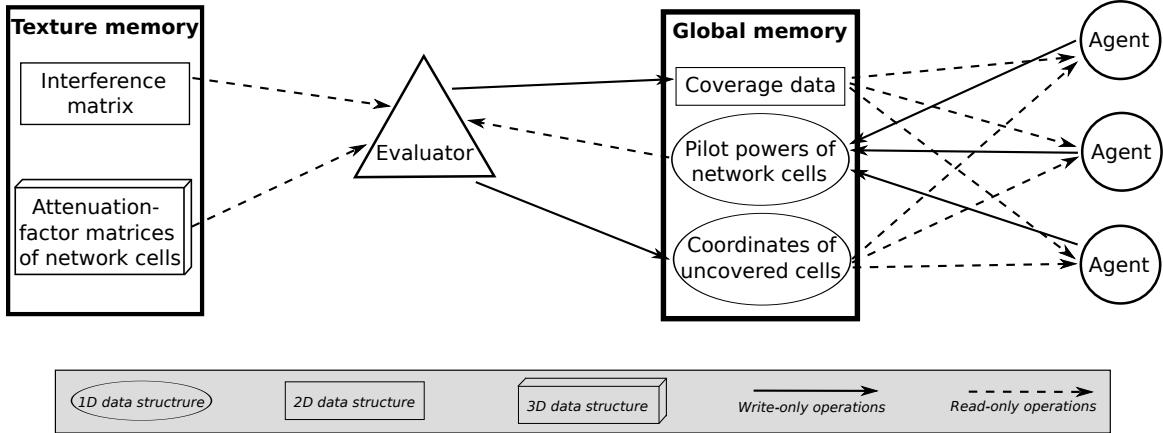


Figure 7.1: Architecture of the parallel, agent-based optimization system on GPU.

Picking the cells conforming to Equation (7.5) and setting the pilot powers accordingly, full coverage of the service area is achieved. The solution exhibits a total pilot power defined as:

$$P^{\text{att}} = \sum_{c \in C} \max p_{cm}^{\text{att}}. \quad (7.6)$$

The procedure to find a cell  $c$  for every UE  $m$  in the service area consists in sorting, in descending order, all UEs by their attenuation-factor values,  $L_{cm}^{\downarrow}$ . The solution is thus established by the first  $b$  UEs of the sorted sequence, taking the maximum pilot-power setting for a cell into account, i.e.,  $p_{cm}^{\text{att}}$ .

### 7.5.2 Parallel-agent approach

In the parallel-agent approach, a set of autonomous worker agents explore the target geographical area,  $A_{\text{total}}$ , in order to optimize the pilot-power consumption. Each agent randomly moves over the  $A_{\text{total}}$  as it dictates different changes to the pilot power of the cells. PRATO, the radio-coverage framework presented in Chapter 6, performs the objective-function evaluations and radio-propagation predictions based on the proposed changes of the agents.

The moving process during the optimization is strictly random. However, several physical properties that are exclusive to the service-coverage problem are being exploited during the exploration of the search space. Additionally, whenever the current solution breaks any of the given constraints, the optimization process is guided back to the space of valid solutions, providing a mechanism for improving exploration and escaping from local optima.

Because the behaviour of the agents is independent between each other, a parallel implementation is fairly straight-forward to achieve. Figure 7.1 shows the architecture of the agent-optimization system. In this GPU-only architecture, agents work in a parallel and autonomous manner, while the evaluator reacts to their changes.

#### 7.5.2.1 Objective-function evaluation

The evaluator represents a central component of the optimization system. It reacts to the pilot-power changes by recalculating the objective-function value. Recall that the objective-function evaluation involves the radio-coverage prediction of the service area and the calculation of the total pilot power used by the cells in the target network.

After a short initialization, during which the attenuation-factor matrices of all the cells and the interference matrix are calculated, the evaluator computes the coverage of the service area based on the pilot powers supplied as the initial solution. Initial solutions are randomly generated from valid pilot-power settings that conform to the full coverage constraint.

The evaluator also maintains a special part of the memory (see “Coordinates of uncovered cells” in Figure 7.1) that is intended for registering uncovered areas, i.e.,  $\overline{A}_{\text{covered}}$ . If Equation (7.4) does not hold, the “special” agents randomly select a location from this portion of memory so that a valid solution may be reached again.

It is worth mentioning that the evaluator itself has no influence in the optimization process from a quality point-of-view. Its task is to provide feedback and updated information to the agents that move through the service area. From a performance point-of-view, the importance of the evaluator is significant, as it will be shown in the following sections.

The evaluation of the objective function was completely implemented on the GPU using OpenCL (see Section 5.2, Chapter 5). The reason behind this decision is the impact objective-function evaluation has on the performance of the optimization system as a whole, as discussed in Section 2.4, Chapter 2. The implementation of the agents is also based on the GPU, which drastically reduces the number of data transfers between CPU and GPU, since all problem elements are available on the GPU during the optimization process. Consequently, careful memory utilization and organization are critical to successfully accommodate all involved problem elements on the GPU, the memory of which is significantly smaller than the RAM memory available in desktop computers.

### 7.5.2.2 Autonomous agents

The agents apply the pilot-power changes only considering local information. Each of them encapsulates a set of steps that is consistently applied as it randomly moves through the service area of the network. Whenever an agent arrives at a new location, the set of covering cells is calculated, i.e.,  $C_m$ .

The step set an agent applies following this point is directly related to  $|C_m|$ , whereas its movement is determined by  $|\overline{A}_{\text{covered}}|$ , i.e., the area without service coverage.

The behavior of an agent is dictated by the pseudo-code shown in Algorithm 7.1. The first four steps are responsible for guiding its movements. The coordinates are randomly selected from two sets,  $A_{\text{total}}$  and  $\overline{A}_{\text{covered}}$ . Only “special” agents may move to a location without service coverage, and they apply the step set  $SS_0$  for as long as the solution is not valid. The portion of “special” agents used for correcting a solution is a parameter of the optimization process. During the following steps of Algorithm 7.1, the agent applies step sets  $SS_0$  and  $SS_1$  based on the number of cells in  $C_m$ .

If the current location of the agent is not covered by any cell, i.e.,  $|C_m| = 0$ , the step set  $SS_0$  is applied (see Algorithm 7.2). At the beginning, the cell with the highest attenuation factor that may cover a UE at this location,  $c'$ , is selected. If several cells have the same  $L_{cm}^{\downarrow}$  value, one of them is randomly chosen. Once  $c'$  is uniquely identified, the agent changes its pilot power by  $inc\_rate$  dB.

The step set  $SS_1$ , the pseudo-code of which is listed in Algorithm 7.3, is applied if the location of the agent is under the coverage of one or more cells, i.e.,  $|C_m| \geq 1$ . The first step randomly selects a cell from the set  $C_m$ , followed by a decrease of the pilot power of cell  $c'$ . This practice keeps the coverage constraint valid over  $A_{\text{total}}$ , although it might potentially break it on other areas. Ideally, every pixel of the geographical area has to be covered by exactly one network cell, although this is just a representation of a perfect solution that is unreachable because of irregularities in the network topology and the terrain.

---

**Algorithm 7.1** Pseudo-code representing the behaviour of an agent.

---

```

repeat
    if is_special_agent() and  $\overline{A_{\text{covered}}} > 0$  then
         $l \leftarrow \text{pick\_random\_location}(\overline{A_{\text{covered}}})$ 
    else
         $l \leftarrow \text{pick\_random\_location}(A_{\text{total}})$ 
    end if
    move(l)
    if  $|C_m| = 0$  then
        apply(SS0)
    else
        if  $|C_m| \geq 1$  then
            apply(SS1)
        end if
    end if
until stopping_criterion()

```

---

**Algorithm 7.2** Pseudo-code representing the step set  $SS_0$ , which is applied by the agents in areas with no service coverage.

---

```

repeat
     $c' \leftarrow \text{next\_cell\_with\_max\_att}(m)$ 
     $p_{c'} \leftarrow \text{adjust\_power}(c', \text{inc\_rate})$ 
until  $p_{c'} \in P_{c'}$ 

```

---

**Algorithm 7.3** Pseudo-code representing the step set  $SS_1$ , which is applied by the agents in areas with service coverage.

---

```

repeat
     $c' \leftarrow \text{pick\_random\_cell}(C_m)$ 
     $p_{c'} \leftarrow \text{adjust\_power}(c', \text{dec\_rate})$ 
until  $p_{c'} \in P_{c'}$ 

```

---

In both step sets,  $SS_0$  and  $SS_1$ , the agent makes sure that the new pilot power setting,  $p_{c'}$ , is an element of  $P_{c'}$ . If this is not the case, cell  $c'$  is discarded and another cell is repeatedly selected at the beginning of both step sets, until this condition is satisfied.

The values *inc\_rate* and *dec\_rate* are configurable parameters that should be set before starting the optimization process. They indicate the relative adjustment (expressed in dB) of the pilot power of cell  $c'$ . On the one hand, lowering the pilot power of a cell decreases the interference it creates within its coverage area and those of their neighbours. Since the  $sir(c, m)$  value increases with lower interference, the coverage of  $m$  may be achieved by a neighbor cell with the same or lower pilot power. On the other hand, increasing the pilot power of the cell with the maximum attenuation factor improves coverage by evenly distributing the power among different network cells. This cell is, on average, the nearest one to the location of a UE  $m$ .

With the objective-function evaluation running on the GPU, a new performance bottleneck appeared. The limitation factor in this case was the CPU-to-GPU data transfers that occurred in each iteration of the optimization process (see Section 5.1, Chapter 5).

The GPU kernel of the agents is launched as one thread block that contains one thread per deployed agent. The thread block is organized in a one-dimensional grid. The initial location of each agent is randomly generated using the current system time as a random seed. Since OpenCL provides no function for random-number generation, a simplified version of Marsaglia's generator [104] was implemented.

The analysis each agent does about the received signals at the current location is saved into the shared memory of the thread block. It contains the network cell and its pilot-power setting. Since both numbers are of type *short*, there is enough space in a 16 KB shared-memory block to allocate 4,096 agents. The last step involves saving the new pilot powers into global memory. This step is performed by only one of the threads within the thread block in order to avoid memory-access conflicts. Updated pilot powers are saved in negative form to indicate that coverage re-calculation is needed for these cells. In case there are several updated pilot powers for one network cell, the median is calculated and applied as the new pilot power.

Even though coalesced access is not achieved by the GPU kernel of the agents, its sole implementation provided enhanced performance. This performance gain appears because of the lower number of data transfers between the CPU and the GPU, since most data are available in global memory. Moreover, the GPU kernel also produces the truly parallel behavior of the agents, as they all apply the pilot-power changes at the same time.

## 7.6 Simulations

### 7.6.1 Test networks

The test networks,  $\text{Net}_1$ ,  $\text{Net}_2$  and  $\text{Net}_3$  are subsets of the real radio network deployed by Telekom Slovenije, d.d. The path-loss predictions were calculated using the radio-propagation model presented in Section 6.3.2, Chapter 6. A DEM with a  $25 \text{ m}^2$  resolution was used as the terrain-profile data. The requirements for the coverage threshold,  $\gamma^{\text{cov}}$ , were provided by experts of the Radio Network department of Telekom Slovenije, d.d.

$\text{Net}_1$  is deployed over a densely populated urban area. For this reason, the value of  $\gamma^{\text{cov}}$  is lower here, since network capacity is the dominating factor, whereas coverage is flexible because of a larger cell density, i.e., more BSs per surface unit.  $\text{Net}_2$  represents a network deployed over a rural area, meaning that the network capacity can be reduced at the cost of a better coverage, since the user density is lower. The last network,  $\text{Net}_3$ , represents a

Table 7.1: Sizes of the test networks used for experimentation of the service-coverage problem, in terms of equipment and geographical area.

	Number of base stations	Number of cells	Surface (km <sup>2</sup> )	Resolution (m <sup>2</sup> )
Net <sub>1</sub>	26	77	100.00	25
Net <sub>2</sub>	8	23	306.25	25
Net <sub>3</sub>	45	129	405.00	25
Net <sub>4</sub>	65	193	56.25	50
Net <sub>5</sub>	12	36	16.00	50
Net <sub>6</sub>	50	148	56.25	50

Table 7.2: Network parameters of the test networks used for the service-coverage problem.

	$p_c^k$	$\tau_0$	$\gamma^{\text{cov}}$
Net <sub>1</sub>	15.00 W	$1.55 \cdot 10^{-14}$ W	0.010
Net <sub>2</sub>	19.95 W	$1.55 \cdot 10^{-14}$ W	0.020
Net <sub>3</sub>	15.00 W	$1.55 \cdot 10^{-14}$ W	0.015
Net <sub>4</sub>	19.95 W	$1.55 \cdot 10^{-14}$ W	0.010
Net <sub>5</sub>	19.95 W	$1.55 \cdot 10^{-14}$ W	0.010
Net <sub>6</sub>	19.95 W	$1.55 \cdot 10^{-14}$ W	0.010

suburban area with a densely populated, but relatively small, downtown center, where a compromise between the network capacity and the coverage has to be achieved.

The second group of test networks, including Net<sub>4</sub>, Net<sub>5</sub> and Net<sub>6</sub>, is part of the publicly available MOMENTUM project [85]. Test network Net<sub>4</sub> represents the city of Berlin (Germany), Net<sub>5</sub> represents the city of The Hague (Netherlands), and Net<sub>6</sub> is the largest network optimized in [148], representing a reduced version of Net<sub>4</sub>. All networks include information about BS locations, path-loss predictions and realistic antennas, which are part of the scenarios provided by the MOMENTUM project.

Network configurations, that represent what could be an initial-network setup by common planning standards [79], were produced using the attenuation-based approach. Such configurations can be easily calculated by a network planner. Table 7.1 lists the number of BSs and cells per test network, as well as the size of the geographical area. Different network-parameter values used during the simulations are shown in Table 7.2.

### 7.6.2 Parameter settings of the parallel-agent approach

The parameter settings for the optimization algorithm were determined after some experimentation with the test networks. The parameter settings for each test networks are listed in Table 7.3.

Using a higher *inc\_rate* than *dec\_rate* reflects the behavior of the agents when full coverage of the service area is not guaranteed. In practice, areas without service coverage usually appear as irregular islands. The stopping criteria were set by limiting the total number of pilot-power changes an agent is allowed to make. The value was set to 10,000, even though for some of the test networks the best solutions were found in the first quarter of the experiment.

Table 7.3: Parameter settings of the parallel-agent approach for each test network.

	Agents	<i>inc_rate</i> (dB)	<i>dec_rate</i> (dB)	Pilot-power changes
Net <sub>1</sub>	16	0.2	-0.1	10,000
Net <sub>2</sub>	16	0.2	-0.1	10,000
Net <sub>3</sub>	16	0.2	-0.1	10,000
Net <sub>4</sub>	6	1.0	-0.1	10,000
Net <sub>5</sub>	2	1.0	-0.1	10,000
Net <sub>6</sub>	6	1.0	-0.1	10,000

Table 7.4: Optimization results after applying two different approaches for solving the service-coverage problem. All values are expressed in Watts.

	Attenuation-based		Parallel agents	
	Total power	Average pilot power	Total power	Average pilot power
Net <sub>1</sub>	419.292	5.445	137.064	1.780
Net <sub>2</sub>	78.297	3.404	33.344	1.450
Net <sub>3</sub>	1,014.113	7.861	582.954	4.519
Net <sub>4</sub>	179.876	0.932	145.715	0.755
Net <sub>5</sub>	73.872	2.052	34.884	0.969
Net <sub>6</sub>	147.014	0.993	112.332	0.759

### 7.6.3 Experimental environment

All experiments were performed on a multi-core Intel i7 2.67 GHz desktop computer with 6 GB of RAM running a 64-bit Linux operating system. The GPU hardware used was an ATI HD5570 with 1 GB of DDR3 RAM. The implementation language used was C, combined with OpenCL and OpenMPI extensions.

### 7.6.4 Results

The results achieved by the parallel-agent approach, listed in Table 7.4, improved the optimization objective significantly. They show that the pilot-power usage was reduced in all networks while the service area was kept under full coverage. Moreover, the parallel-agent solution for Net<sub>1</sub> improved the attenuation-based setting by more than 300%. As for Net<sub>2</sub>, the observed improvement is around 232%, while the improvement for Net<sub>3</sub> is more than 170%.

As mentioned in Section 7.1, the practical interpretation of these results is directly related to a capacity increase due to reduced interference. Specifically, the capacity of the network has been significantly increased in all three problem instances. Therefore, a greater number of users should be able to access services provided by the mobile network, since coverage is assured. Moreover, an increased speed in high-speed data services should also be observed [79].

The last test network, Net<sub>6</sub>, is the same as in [148]. When comparing these results to those of [148], an improvement of almost 3% can be observed in the solution provided by the parallel-agent approach.

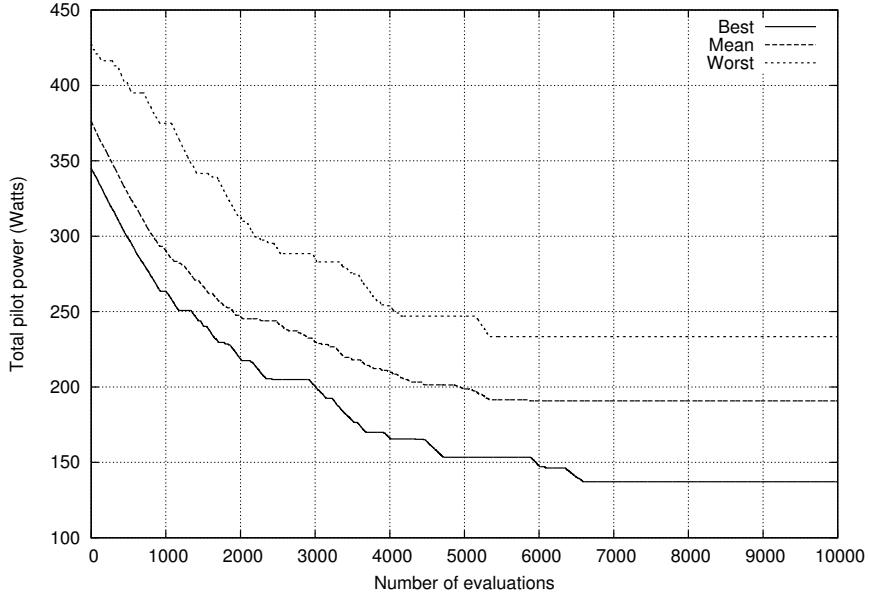


Figure 7.2: Convergence profile of the parallel-agent approach for the test network  $\text{Net}_1$ .

#### 7.6.4.1 Performance analysis

The graphs shown in Figures 7.2, 7.3 and 7.4 depict the convergence of the parallel-agent approach after ten independent runs for test networks  $\text{Net}_1$ ,  $\text{Net}_2$  and  $\text{Net}_3$ , respectively. Only feasible solutions were plotted, i.e., the solutions that meet the full-coverage constraint. Unfeasible solutions were marked with a value of inferior quality than the worst solution found: 428 for  $\text{Net}_1$ , 129 for  $\text{Net}_2$ , and 1,435 for  $\text{Net}_3$ .

From the graphs of  $\text{Net}_1$  (see Figure 7.2) and  $\text{Net}_2$  (see Figure 7.3), a good initial convergence can be observed. This is followed by a steady improvement of the intermediate solutions. In  $\text{Net}_1$ , no additional solution improvement is noticed towards the end of the optimization process. This fact suggests that the stopping criteria is suitable for this problem instance. A similar situation is observed for  $\text{Net}_2$ , that also shows a flat profile towards the end. From the graph of  $\text{Net}_3$  (see Figure 7.4), a slower initial convergence, followed by a steady improvement of intermediate solutions and no significant solution enhancement towards the end, can be observed. This convergence profile suggests that this problem instance presents a more difficult optimization case than for  $\text{Net}_1$  and  $\text{Net}_2$ . Indeed, this is the largest test network in terms of surface area. However, further investigation is needed to confirm this hypothesis. Nevertheless, the parallel-agent approach improved the pilot-power usage of this test network by almost 75%.

In the following, the speed-performance analysis of the experimental simulations is presented. This analysis covers the running times during the optimization of the first three test networks, i.e.,  $\text{Net}_1$ ,  $\text{Net}_2$  and  $\text{Net}_3$ . The running times were measured for each implementation, and the best times, calculated after ten independent runs, are given. The number of pilot-power changes per agent was limited to 100, while all other algorithm parameters were kept at the same values as in Section 7.6.2.

Table 7.5 lists the wall-clock times in seconds for the different implementations and test networks. The implementations include: the CPU-MPI implementation that consists of objective-function evaluation on CPU and parallel agents over MPI, the GPU-MPI implementation that consists of objective-function evaluation on GPU and parallel agents over MPI, and the GPU-GPU implementation that consists of objective-function evaluation and

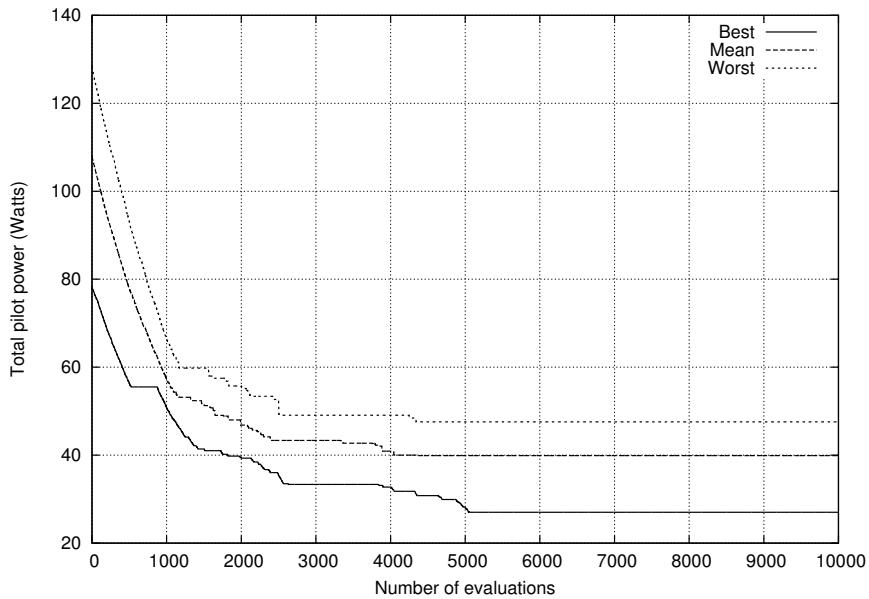


Figure 7.3: Convergence profile of the parallel-agent approach for the test network Net<sub>2</sub>.

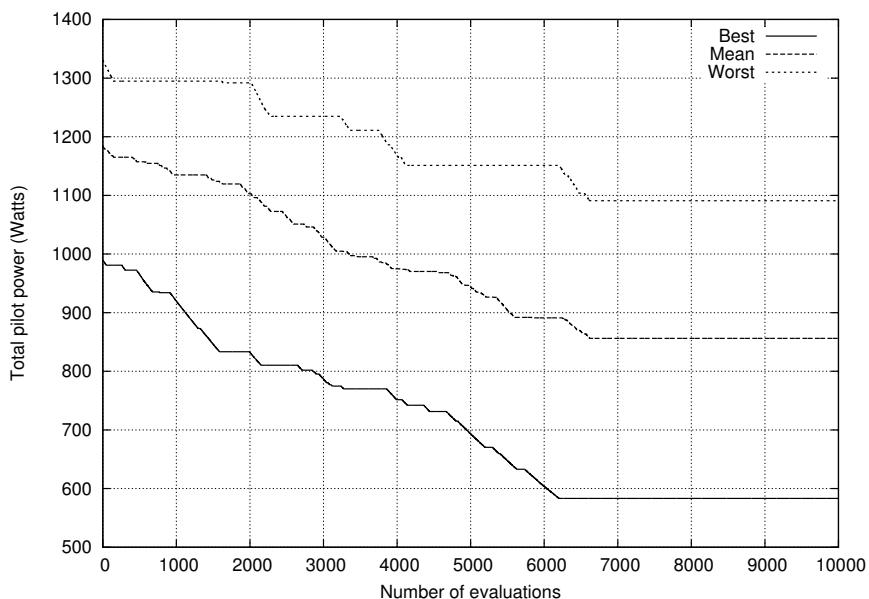


Figure 7.4: Convergence profile of the parallel-agent approach for the test network Net<sub>3</sub>.

Table 7.5: Wall-clock times (in seconds) and speed-up factors for the different implementations of the objective-function evaluation and the parallel agents, during the experimentation of the service-coverage problem.

<This times are not final??>						
	CPU-MPI		GPU-MPI		GPU-GPU	
	Best time		Best time	Speed-up	Best time	Speed-up
<i>Net</i> <sub>1</sub>	105,455		346	305x	67	1574x
<i>Net</i> <sub>2</sub>	33,700		195	173x	46	733x
<i>Net</i> <sub>3</sub>	191,900		506	379x	117	927x

parallel agents on the same GPU. The CPU-MPI implementation is the basis for the speed-up calculation of the other two implementations.

The function evaluation on the GPU, that communicates with the agents over MPI, provides the second measured setup. The evaluator implementation takes advantage of shared memory for thread collaboration within a thread block and texture memory for constant elements, as is it was explained in Section 6.4.5.3, Chapter 6. Still, the speed-up is considerable but improvable, since numerous data transfers between CPU and GPU are needed for the agents to access optimization-related information.

The last result set presents measurements for the complete GPU implementation, including objective-function evaluation and agents on the same device. The substantial speed-up delivered by this combination highlights the great impact that CPU-to-GPU memory transfers have on the overall system performance. This fact is supported by the speed-up between the second and third measured setups, which exhibit, on average, an improvement of more than ??%.

## 7.7 Summary

This chapter presented a novel optimization approach for solving the well-known service-coverage problem in radio networks. The problem addressed the full coverage of a geographical area using a minimum amount of pilot power. The newly introduced parallel-agent approach was successfully tested in six networks that represent real-world scenarios. The experimental results show that the parallel-agent approach is able to find better solutions than other common radio-planning methods [79]. Moreover, the algorithm successfully tackled larger networks, thus overcoming the obstacles of other state-of-the-art optimization methods regarding problem-instance size [142, 148].

Compared to a different optimization approach in the literature [147], the solution-quality of the parallel-agent approach showed a quality improvement. The proposed solutions, calculated for the same problem instance as in [148], were improved at the cost of longer running time. It is worth mentioning that it is feasible for the optimization algorithm to take a longer time to reach the solution, since design problems, as the service-coverage one, are usually solved offline. A comparison and analysis of the performance of the radio-coverage prediction for real-world, radio-network planning is latter provided in Chapter 10.

Different implementations of the parallel-agent approach, combining a serial version on CPU, parallel processes over MPI and GPU kernels, were presented. In particular, GPU architectures enable the implementation of parallel heuristics in a natural way while substantially improving the computational-time performance. To the best of the author's knowledge, the parallel-agent approach as presented in this chapter, has not yet been described in the related literature.

## 8 The soft-handover balancing problem

In Chapter 7, an application exploiting the advantages of faster evaluation methods has been presented. Solving the service-coverage problem for real-world networks capitalizes on the ability to tackle bigger problem instances. Because of their size, such problems were previously unsolvable in a feasible amount of time. This improved performance also allows solving optimization problems with a higher degree of complexity, usually represented by the evaluation of multi-dimensional, non-convex objective functions.

This chapter focuses on solving a new optimization problem for 3G networks, that deals with downlink and uplink SHO areas (see Section 3.1, Chapter 3). By introducing a penalty-based objective function and some hard constraints, the formal definition of the SHO-balancing problem in UMTS networks is given. The state-of-the-art mathematical model used and the penalty scores of the objective function are set according to the configuration and layout of a real mobile network, deployed in Slovenia by Telekom Slovenije, d.d. The balancing problem is then tackled by three optimization algorithms, each of them belonging to a different category of metaheuristics.

To the best of the author's knowledge, there is no reference in the literature of a simulation-based approach to find active downlink and uplink SHO areas. Additionally, there are no formal optimization methods known to the author that tackle the SHO balancing problem as described here. The approach described in this chapter extends the research work published by the author in [19].

The remainder of this chapter is organized as follows. Section 8.1 describes the motivation behind the SHO-balancing problem, whereas Section 8.2 gives an overview of other works related to pilot-power and SHO optimization in UMTS. The static network model is presented in Section 8.3, where all the elements of the mathematical model and the objective function are defined. In Section 8.4, the problem is formally defined, followed by a short description of the optimization algorithms used in Section 8.5. The simulations, including their environment and parameter setup, are introduced in Section 8.6, before their result analysis in Section 8.6.4.

## 8.1 Motivation

Despite several built-in mechanisms, that allow a radio network to overcome different problems due to the lack of SHO during a HSDPA connection, some abnormal cases do arise, especially in those areas where there is SHO capability in the uplink, but none in the downlink. An example of such a case is depicted in Figure 8.1, which shows the interference behavior during a HSPA connection in: (a) normal SHO conditions, and (b) unbalanced SHO conditions. The plotted data are actual radio network statistics, taken from the mobile network deployed in Slovenia by Telekom Slovenije, d.d. The graph on the left, (a), shows a normal HSUPA-enabled service situation, in which the measured interference is proportional to the traffic being served. Note how the noise rises with the increased traffic on cell 1, while its neighbor (cell 2) has almost no interference nor traffic. Moreover, the graph profile for both traffic and noise of cell 1 are almost identical. The graph on the right, (b), depicts a problematic situation, where the noise level does not only rise on the cell serving the HSUPA services (cell 1), but also on the neighboring one. Notice how the interference level rises on the cell that has almost no traffic (cell 2). It is clear that the source of this noise rise is generated by the active connection on cell 1, which shows an increase in HSUPA traffic. However, the noise-level profile on cell 2 does not follow its traffic, as it did in the normal situation (a). This is due to cell 2 not being part of the active set. Such situations appear when the UL coverage is larger than the DL coverage. Interestingly enough, this seems to be an exceptional case, as Holma and Toskala write in [80], when describing the SHO in chapter 5:

"... There is no obvious reason why the serving E-DCH cell would not be the same as the serving HSDPA cell, and this is also required to be the case in the specifications."

Given the described context, the challenge is to achieve the correct balance or distribution of downlink and uplink SHO areas within a working UMTS network. Therefore, the network has to be fine-tuned to improve the SHO-area balancing, thus to avoid the exceptional appearance of problematic situations, as shown in Figure 8.1. This clearly implies that the mobile network configuration should not be excessively altered, since other aspects of the network are working well before starting the optimization process. Hence, the objective of the optimization problem is to find a pilot-power configuration for all the cells in the target network, such that the balance of downlink and uplink SHO areas is improved and other network aspects are preserved. The optimization process takes into account different kinds of hardware, e.g., amplifiers, cables, and antennas, adjusting the pilot powers of the cells.

PRATO, as defined in Chapter 6, is used as the evaluation framework of the SHO-balancing problem. A state-of-the-art mathematical model [112] describes the downlink and uplink SHO areas. By introducing a penalty-based objective function and some hard constraints, a formal definition of the SHO-balancing problem in UMTS networks is given. The mathematical model and the penalty scores of the objective function are set according to the configuration and layout of a real mobile network, deployed in Slovenia by Telekom Slovenije, d.d. The SHO settings are also taken from the actual network configuration, still they were adapted to closely model interference and other dynamic aspects of the network.

## 8.2 Related work

The SHO optimization has received quite some attention from the scientific community during the past years. This mainly relates to the importance it has within the deployed networks that provide high-speed services, such as video telephony [28] and data services by means of HSPA [33, 31].

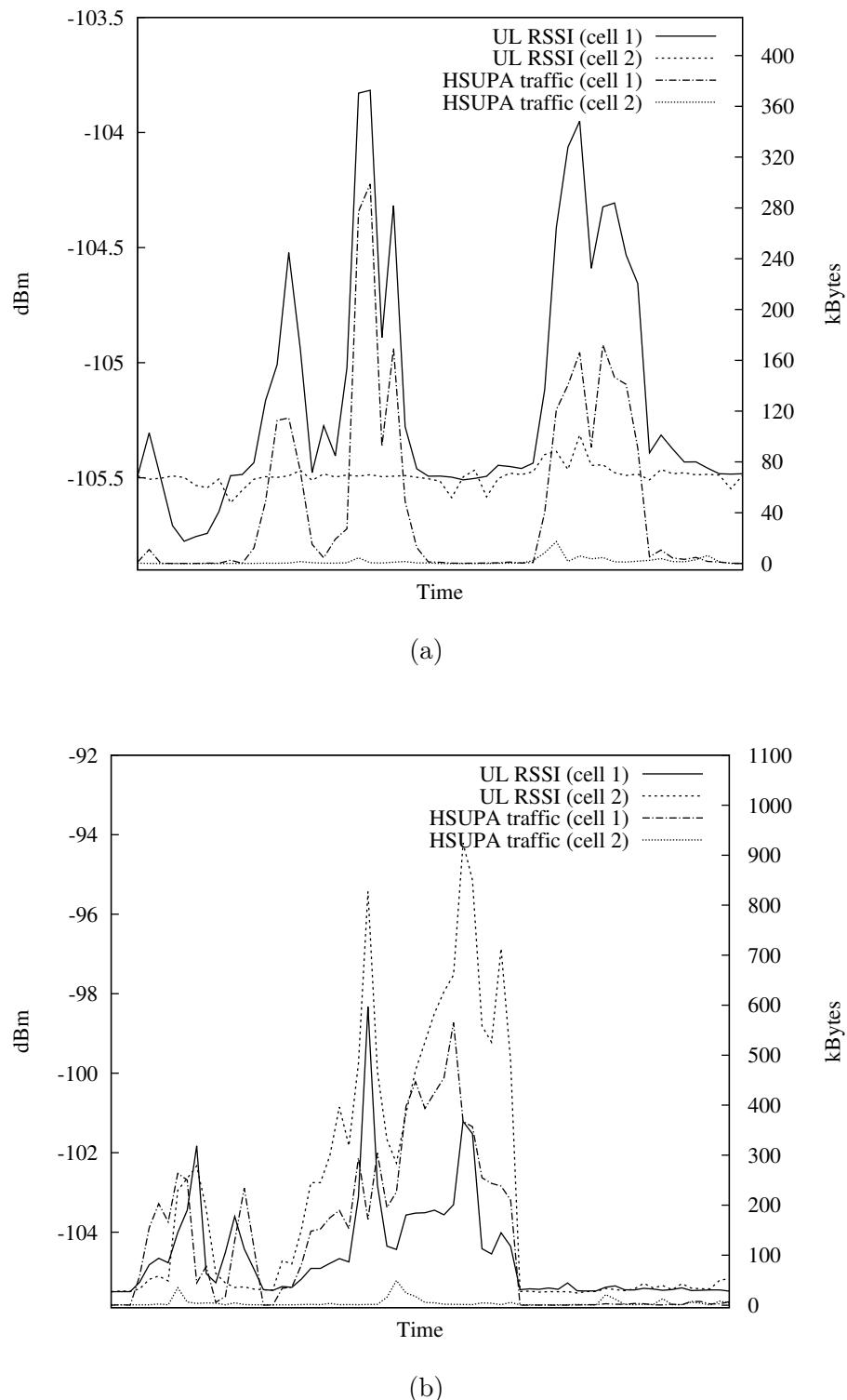


Figure 8.1: HSUPA traffic and uplink interference with: (a) balanced downlink and uplink SHO conditions, and (b) unbalanced downlink and uplink SHO conditions.

Some authors tackled optimization problems at the planning stage of the network [49, 62], considering, among other variables, BS locations and hardware. However, most mobile operators are unable to apply such methods to a live network, since the planning phase of the new installation has long been concluded. Moreover, the great majority of the BSs has already been deployed and their hardware also installed. Therefore, from the mobile operator's point of view, mainly parameter and software optimization are the only tools available, when it comes to QoS improvement (see Chapter 4) and network troubleshooting in the short term.

Optimizing SHO by means of pilot-power adjustment is an established way of enhancing network capacity, when high-speed services like HSDPA and HSUPA coexist with legacy technologies [31]. In the UMTS, the pilot power is typically between 5% to 10% of the total downlink transmit power of the BS [95], but there is no standardized method to find a pilot-power setting. A number of existing approaches to resolve this issue exist in the related literature [78, 140, 181], being those based on optimization methods the most effective ones [49, 55, 95, 98, 145]. Such a wide spectrum of available procedures is directly related to the diverse criteria taken into account when assigning the pilot power of a cell. The fundamental reason behind this fact is that the pilot power is a common adjustment variable of various optimization problems in radio networks. This is especially true for UMTS networks, due to their frequency-reuse factor of 1 [78].

## 8.3 Radio-network model

This section extends the representation of the radio-network model, previously introduced in Section 7.3, Chapter 7, in order to include the SHO functionality. In this context, the mathematical model links the SHO settings with the pilot-power level of each cell, the best-server pattern, and the network coverage.

By introducing a change step of 0.01 dB and bounding the pilot power of a cell  $c$ ,  $c \in C$ , to  $\pm 2$  dB (relative to the pilot-power setting the cell had before optimization), the number of elements in the set  $P_c$  is reduced. The purpose of this reduction is twofold. First, since the optimization targets a live network, there is no need for the algorithms to create complete new configurations, but just to fine-tune existing ones. Second, the problem complexity is lowered, because the size of the search space is smaller and discrete.

### 8.3.1 Soft-handover areas

To obtain a realistic outline of the areas where an UE may potentially maintain connections to more than one cell, a static version of the active set, as defined in [112], is used. To this end, a SHO window,  $\gamma^{\text{sho}}$ , and a maximum active-set size,  $as^{\max}$ , are introduced. Both parameters are taken from the working configuration of the real network. It follows that the cells to which an UE  $m$ ,  $m \in M$ , may maintain concurrent downlink connections are part of the set:

$$\begin{aligned} SHO_m^\downarrow &= \left\{ c \mid L_{c^*m}^\downarrow p_{c^*} - L_{cm}^\downarrow p_c \leq \gamma^{\text{sho}} \right\}, \\ |SHO_m^\downarrow| &\leq as^{\max}, \end{aligned} \quad (8.1)$$

where  $L_{c^*m}^\downarrow$  is the downlink attenuation factor of the best-serving cell, and  $p_{c^*}$  is its pilot power. Since the number of elements in  $SHO_m^\downarrow$  is at most  $as^{\max}$ , the weakest links are removed if there are several present. This method is well suited for configurations with no hysteresis, since dynamic effects are ignored in static models [112].

Additionally, in the uplink, the set of cells to which an UE can potentially be in SHO is defined as:

$$SHO_m^\uparrow = \left\{ c | L_{mc}^\uparrow p_m^\uparrow \geq 3.16227766 \cdot 10^{-12} mW \right\}, \quad (8.2)$$

where  $L_{mc}^\uparrow$  is the uplink attenuation factor from an UE  $m$  to a cell  $c$ , and  $p_m^\uparrow$  is the uplink transmit power of  $m$ .

The static nature of the model intentionally neglects mobility and dynamic interference by narrowing  $\gamma^{\text{sho}}$  down to 2 dB [112].

## 8.4 Problem definition

Using the elements defined in Section 8.3, an objective function was formulated in cooperation with a team of radio engineers of the Radio Network Department at Telekom Slovenije, d.d. The objective function is constructed as a weighted sum, containing different costs that penalize the occurrence of specific SHO conditions in downlink and uplink, which may potentially cause the afore-mentioned malfunctioning, introduced in Section 8.1.

A cost-based objective function is the most natural and straight-forward way of defining the optimization objective. Besides it is easily extendable to include other future situations, also defining the mutual importance of the different phenomena taken into account at the optimization phase.

Hence, the definition of the objective function for the SHO-balancing problem is the minimization of the sum of penalty scores given as:

$$\min f_{\text{sho}} = \sum_{c \in C} \sum_{m \in M} pf_{\text{cov}}(1 - cov_{cm}) + pf_{\text{sho}}^\uparrow sho_{cm}^\uparrow (1 - sho_{cm}^\downarrow) + pf_{\text{sho}}^\downarrow sho_{cm}^\downarrow (1 - sho_{cm}^\uparrow), \quad (8.3)$$

where

$$sho_{cm}^\downarrow = \begin{cases} 1 & c \in SHO_m^\downarrow \\ 0 & \text{otherwise} \end{cases}, \quad (8.4)$$

$$sho_{cm}^\uparrow = \begin{cases} 1 & c \in SHO_m^\uparrow \\ 0 & \text{otherwise} \end{cases}, \quad (8.5)$$

and

- $pf_{\text{cov}}$  represents the penalty factor for uncovered areas,
- $pf_{\text{sho}}^\uparrow$  represents the penalty factor for uplink SHO areas where SHO is not possible in the downlink, and
- $pf_{\text{sho}}^\downarrow$  represents the penalty factor for downlink SHO areas where SHO is not possible in the uplink.

After extensive experimentation, and working in cooperation with the radio engineers from the Radio Network Department at Telekom Slovenije, d.d., the penalty factors from Equation (8.3) are set to the following values:

- $pf_{\text{cov}} = 15$ ,
- $pf_{\text{sho}}^\uparrow = 13$ , and

- $pf_{\text{sho}}^{\downarrow} = 3$ .

It is clear that the coverage is the most important quality aspect from the network point of view (penalty factor  $pf_{\text{cov}}$ ). Moreover, it imposes the biggest constraint to the optimization process, since the balance between SHO areas should not sacrifice network coverage. Another important characteristic that emerges from these values is the preference for minimizing areas where SHO capability is available in the uplink, but not in the downlink (penalty factor  $pf_{\text{sho}}^{\uparrow}$ ). As it has been described in Section 8.1, the consequences of such SHO arrangement produce severe interference in neighboring cells (Figure 8.1), which may also result in service inaccessibility. The last factor  $pf_{\text{sho}}^{\downarrow}$  imposes a penalty value over areas where the SHO capability is available in the downlink, but not in the uplink. Recall that when accessing HSPA services, SHO is available only in the uplink. For this reason, the link throughput may benefit from the SHO in the uplink if it is available. The relative lower importance of the last penalty factor, when compared to the other ones, is directly related to the consequences of the unbalancing that such SHO areas may have on the network. In this case, only the HSPA throughput is affected, while the service accessibility should not be an issue, given there is enough uplink coverage [80].

## 8.5 Optimization approaches

The SHO-balancing problem has been tackled using three fundamentally different optimization algorithms, namely:

- DE (see Section 2.3.1, Chapter 2), from the family of evolutionary algorithms;
- DASA (see Section 2.3.2, Chapter 2), from the family of swarm-intelligence algorithms; and
- SA (see Section 2.3.3, Chapter 2), from the group of classic metaheuristic algorithms, targeted at combinatorial optimization problems.

Each of these algorithms shall minimize the objective function value by adopting essentially disparate approaches, hence the diversity of applying algorithms belonging to different families to solve the same optimization problem. Therefore, the result analysis shall establish which of the presented approaches is better suited for solving the SHO-balancing problem.

The following sections describe how the SHO-balancing problem is represented by the internal structure of each of the selected algorithms and their controlling parameters.

### 8.5.1 Differential evolution

The DE algorithm features a parallel direct search method, which utilizes a population of  $D$ -dimensional parameter vectors. The SHO-balancing problem is expressed in each component of a vector  $X$  of the population, which represents the pilot power of a target cell, i.e.:

$$X_{aG} = \{x_1, x_2, \dots, x_c, \dots, x_D\}, \quad (8.6)$$

where  $x_c \in P_c$  represents a candidate pilot-power setting of cell  $c$ , and  $G$  indicates the generation of an individual  $a$  in the population. Since there are  $|C|$  cells in a mobile network, it follows that the population size,  $D = |C|$ .

From the different variants of DE, the most popular one is used here, called *DE/rand/1/bin*. The nomenclature used to name this variant indicates the way the algorithm works:

- *DE* denotes the differential evolution algorithm,

---

**Algorithm 8.1** A move in the search space of SA for solving the SHO-balancing problem.

---

```

 $c' \leftarrow \text{pick\_random\_cell}(C)$ 
repeat
    if  $\text{uniform}[0, 1] < 0.5$  then
         $p_{c'}^{\text{new}} \leftarrow p_{c'} + 0.01$ 
    else
         $p_{c'}^{\text{new}} \leftarrow p_{c'} - 0.01$ 
    end if
until  $p_{c'}^{\text{new}} \in P_{c'}$ 
 $p_{c'} \leftarrow p_{c'}^{\text{new}}$ 

```

---

- *rand* indicates that the individuals selected to compute the mutation values are randomly chosen,
- 1 specifies the number of pairs of selected solutions used to calculate the crossover vector, and
- *bin* means that a binomial recombination operator is used.

### 8.5.2 Differential ant-stigmergy algorithm

The mapping between the balancing problem and DASA is similar to the one for DE:

$$X_a = \{x_1, x_2, \dots, x_i, \dots, x_D\} \quad (8.7)$$

In this case, each ant,  $a$ , creates its own solution vector,  $X_a$ , during the minimization process. At the end of every iteration, and after all the ants have created solutions, they are evaluated to establish if any of them is better than the best solution found so far.

### 8.5.3 Simulated annealing

From the SA perspective, the system under optimization is in a given *state* at each time step during the process. The objective function maps a system state to a value known as the *energy* of the system in that state. A *move* in the search space represents a change in the state of the system. After making a move, the system may exhibit lower or higher energy, depending on the results of the objective function.

Algorithm 8.1 shows the pseudo-code of a move in the search space of possible pilot-power settings, resulting in a new state of the system.

At the first step, a cell,  $c'$ , is randomly selected from the set of all cells in the network,  $C$ . In step 2, a change of +0.01 dB or -0.01 dB is applied with 50% probability to  $p_{c'}$ . The pilot power of cell  $c'$  is expressed in dBm. The randomly generated pilot-power setting,  $p_{c'}^{\text{new}}$ , is checked for validity in step 3, i.e., it must be an element of the set  $P_{c'}$ . If  $p_{c'}^{\text{new}}$  is not a valid pilot power, step 2 is executed again, generating another random pilot power. Finally, in step 4, the pilot power of cell  $c$  is replaced by  $p_{c'}^{\text{new}}$ .

It is important to note that, as long as  $|P_{c'}| > 1$ , the pseudo-code shown in Algorithm 8.1 shall never be trapped in an endless loop. On the other hand, if  $|P_{c'}| < 2$ , there are no candidate pilot powers for cell  $c'$ , and thus there is no possibility of optimization. Notice also that the acceptance of a move in the search space is left to SA and its stochastic components.

## 8.6 Simulations

The simulations were performed over the target geographical area, for which DEM and clutter data were available. The mobile users were assumed to be uniformly distributed. The SHO conditions were determined by the relative received-signal quality from different cells, and the SHO window, which triggers the addition of a cell to a user's active set [78].

### 8.6.1 Test network

The test network used for the simulations, Net<sub>7</sub>, is a subset of the real UMTS network deployed in Slovenia by Telekom Slovenije, d.d. It represents a network extending over a hilly terrain, combining both rural and middle-dense suburban areas, which contains 25 cells within an area of more than 150 km<sup>2</sup>. Table 8.1 shows some characteristics of the test network used, and Figure 8.2 shows the area under radio coverage,  $A_{\text{covered}}$ , within  $A_{\text{total}}$ .

Table 8.1: Technical characteristics of Net<sub>7</sub>, the test network used for the SHO-balancing problem.

Number of cells	25
Coverage threshold (RSCP)	-115 dBm
SHO window ( $\gamma^{\text{sho}}$ )	2 dB
User equipment ( $p_m^\uparrow$ )	21 dBm, power class 4
Pixel resolution	25 m <sup>2</sup>
Population density	398/km <sup>2</sup>

### 8.6.2 Algorithm parameters

In this section, the algorithm-parameter setup used during the simulations is given. In all three cases, the parameter names are given with their respective values and descriptions.

The parameters controlling the behavior of the DE algorithm were set as follows:

- $NP = 100$ , the population size;
- $G_{\max} = 1000$ , the maximum number of generations for the algorithm to run;
- $CR = 0.8$ , the crossover constant; and
- $F = 0.5$ , the mutation-scaling factor.

As for DASA, the parameters were set to the following values:

- $m = 10$ , the number of ants;
- $b = 10$ , the discrete base;
- $q = 0.2$ , the pheromone dispersion factor;
- $s_+ = 0.01$ , the global scale-increasing factor;
- $s_- = 0.01$ , the global scale-decreasing factor; and
- $e = 1.0^{-2}$ , the maximum parameter precision.

There are only two parameters controlling SA, namely:

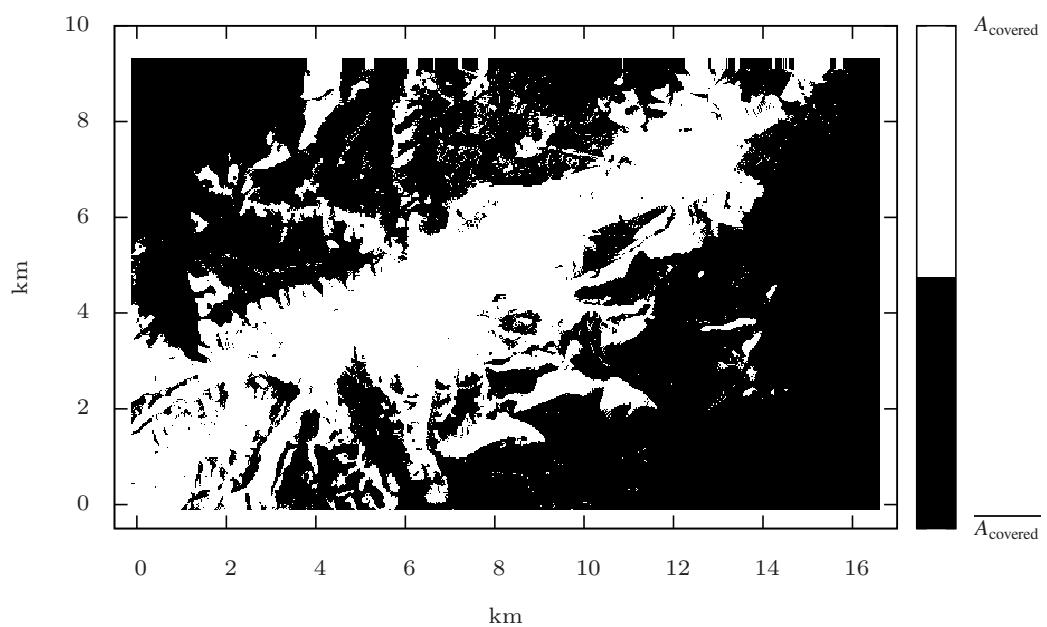


Figure 8.2: Area under radio coverage,  $A_{\text{covered}}$ , and without radio coverage,  $\overline{A}_{\text{covered}}$ , within the complete geographical area,  $A_{\text{total}}$ , of test network  $\text{Net}_7$ .

Table 8.2: Solution-quality performance of the three algorithms, after 30 independent runs.

	Best	Worst	Mean	Std. deviation
DE	2,286,292.00	2,286,541.00	2,286,517.09	62.06
DASA	2,286,446.00	2,286,633.00	2,286,592.00	26.19
SA	2,293,350.00	2,295,570.00	2,294,626.50	663.75

- $t_{initial} = 125$ , the initial temperature; and
- $it = 100,000$ , the total number of iterations.

In this case, the exponential-lowering schema was chosen as the way the temperature was lowered during the SA searches.

### 8.6.3 Experimental environment

All experiments were carried out on a 4-core Intel i7 2.67 GHz desktop computer with 6 GB of RAM running a 64-bit Linux operating system. The implementation languages used were C and Python, with the latter mostly used as ‘glue’ to hold the different implementation parts together, as well as for I/O operations. To lower the time needed to run one optimization round, the entire objective-function evaluation was implemented using OpenCL and executed on a nVidia GeForce GTX 260. This individual change exhibited more than 15-fold, execution-time speedup, when compared to the original CPU-only version.

### 8.6.4 Results

In this section, the performance of the selected algorithms is presented. The analysis includes aspects related to solution quality and convergence speed. All experimental results were obtained after 30 independent runs, each of them limited to a maximum of 100,000 evaluations. The gathered results are shown in Table 8.2.

It may be observed that DE reached the lowest objective-function value, closely followed by DASA. Likewise, both algorithms arrived at very similar results for the worst, mean and standard-deviation values. SA, on the other hand, did not achieve comparable values, since its results are behind those of DE and DASA. Notice that even the best SA solution is no better than the worst solution of DASA. Moreover, the standard deviation exhibited by SA is one order of magnitude bigger than those of DASA and DE.

The convergence of the best-recorded run of each of the three algorithms is shown in Figure 8.3. It is worth mentioning that every optimization run starts from a different solution, randomly constructed by picking a pilot-power setting,  $p_c^k$ , from every  $P_c$ ,  $1 \leq k \leq |P_c|$ ,  $\forall c \in C$ . Notice how fast DASA converged to a good solution. After a number of evaluations without improvement, DASA resets itself and continues searching from a new random point within the search space [93]. DE also converged considerably fast, although not as fast as DASA did. In this case, DE does not reset itself if the current solution cannot be improved. Despite this, and based on the flat profile the graph exhibits towards the end of the optimization run, it is clear that 100,000 evaluations is an adequate stopping criterion for all algorithms. The third algorithm, SA, slowly converged towards the best solution found, even though it was not as good as the solutions found by DE and DASA.

The three convergence profiles shown in Figure 8.3 give a clearer notion about the way these algorithms explore the search space of the SHO-balancing problem.

The simulation-running times have been intentionally omitted, since the algorithm implementations are fundamentally different and therefore not comparable with each other.

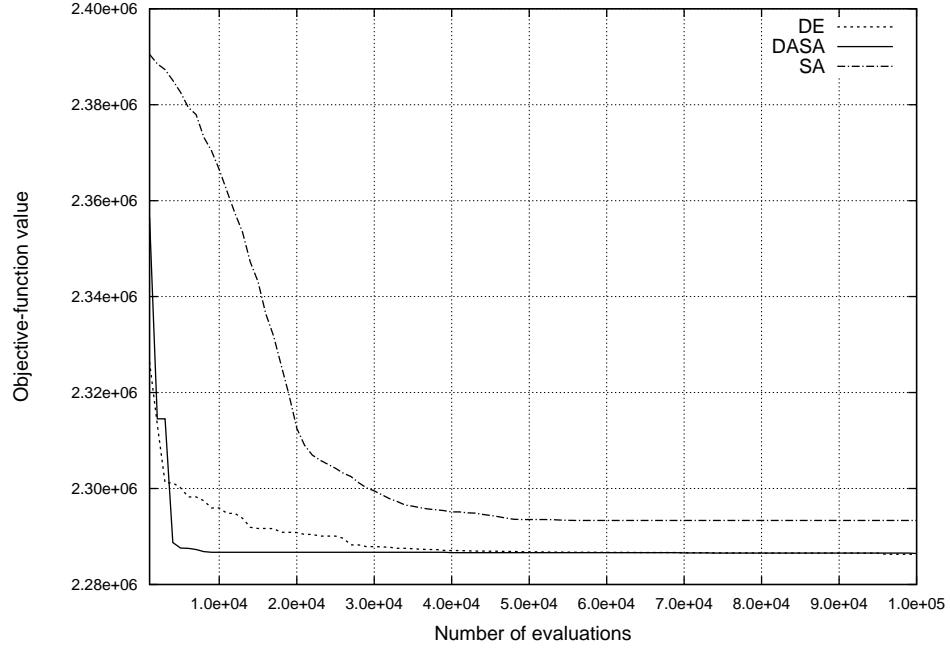


Figure 8.3: Convergence analysis for each of the three algorithms, i.e., DE, DASA and SA, showing the best results obtained for the SHO-balancing problem.

#### 8.6.4.1 Performance analysis

Table 8.3 presents the analysis of the obtained results from the network point of view. After 30 independent runs of each of the three algorithms, the best results obtained were evaluated for the improvement and the decline of each of the measured network-performance aspects. The results are shown in Table 8.3, where '+' indicates improvement and '-' indicates a decline of a given criteria. Overall, it may be observed that the measured criteria have been significantly improved. The only exception is the measure for downlink SHO, without SHO in the uplink (labeled as 'SHO $\downarrow$ , no SHO $\uparrow$ '), which shows an expected decline, since it is the optimization aspect with the lowest penalty-factor value.

The coverage has been improved with an average of 4.29%, whereas the coverage area where there is no SHO capability, has been increased 7.74% in average. Areas where SHO is available in both downlink and uplink have also been improved, i.e., 3.75% in average. This particular improvement is interesting from the optimization point of view, because it had no explicit penalty factor set. Therefore, this may be understood as a consequence of the correct representation of the different network aspects in the objective function.

Table 8.3: Improvement analysis of the best solution that each algorithm achieved for the SHO-balancing problem.

	Uncovered	Covered, no SHO	SHO	no SHO <sup>†</sup> , SHO <sup>†</sup>	SHO <sup>†</sup> , no SHO <sup>†</sup>	Total
Before opt.	63.00 %	15.11 %	15.73 %	1.80 %	4.36 %	100.00 %
DE sol.	60.23 %	16.13 %	16.09 %	1.47 %	6.08 %	100.00 %
DASA sol.	60.24 %	16.16 %	16.90 %	1.46 %	5.24 %	100.00 %
SA sol.	60.42 %	16.55 %	15.97 %	1.56 %	5.50 %	100.00 %
DE impr.	+4.40 %	+6.75 %	+2.29 %	+18.33 %	-39.45 %	—
DASA impr.	+4.38 %	+6.95 %	+7.44 %	+18.88 %	-20.18 %	—
SA impr.	+4.09 %	+9.53 %	+1.52 %	+13.33 %	-26.15 %	—
Avg. impr.	+4.29 %	+7.74 %	+3.75 %	+16.85 %	-28.59 %	—

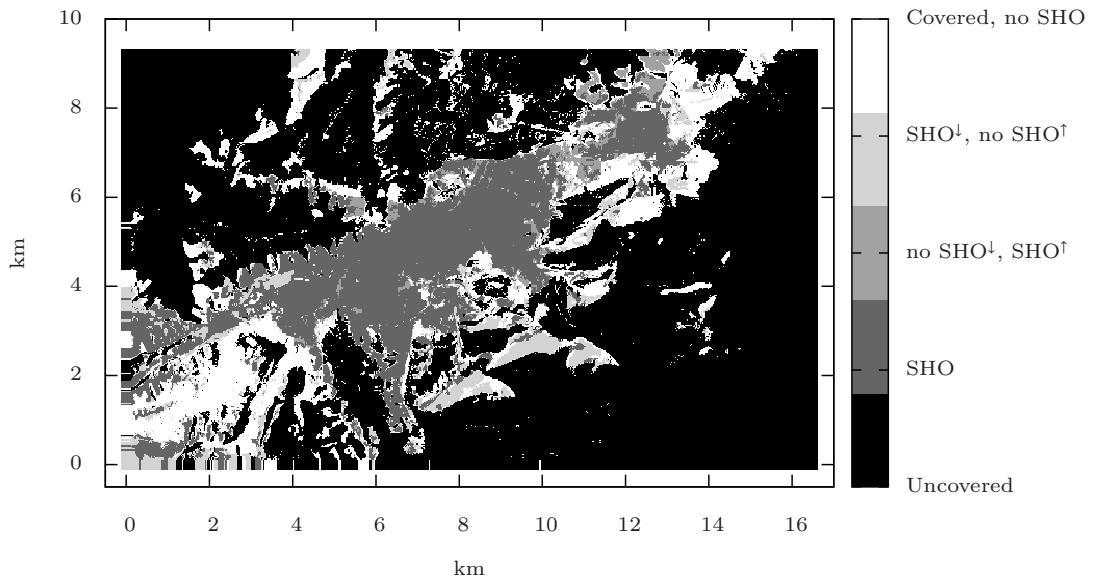


Figure 8.4: Spatial distribution of the SHO areas before the optimization.

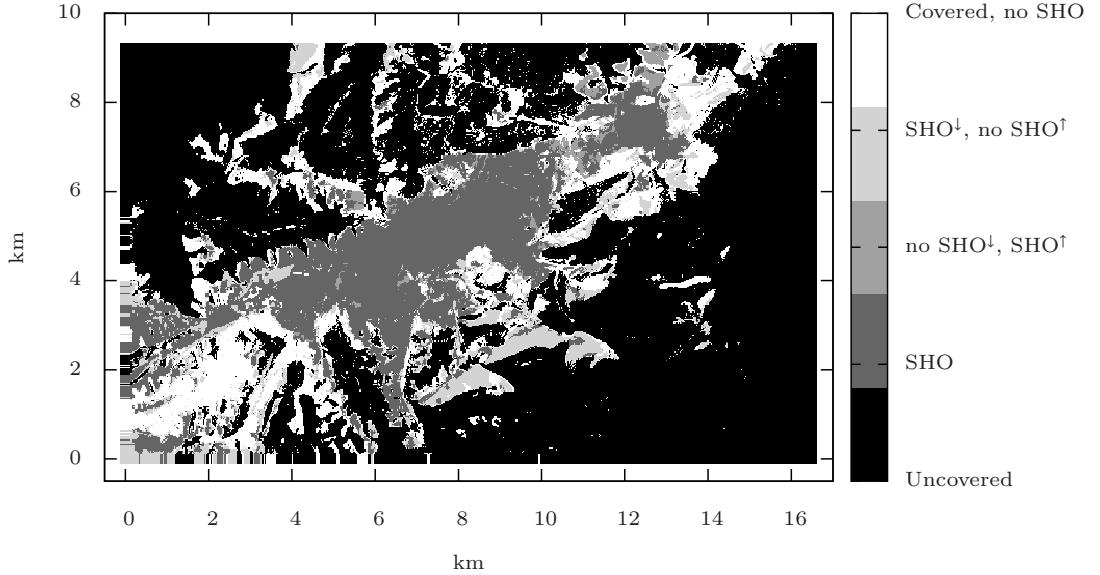


Figure 8.5: Spatial distribution of the SHO areas after the optimization.

The second most important optimized aspect in the SHO-balancing problem is the proportion of areas with uplink SHO and no SHO in the downlink (labeled as ‘no  $\text{SHO}^\downarrow$ ,  $\text{SHO}^\uparrow$ ’ in Table 8.3). This particular condition has been improved by almost 17% in average, greatly reducing the possibility of interference in neighboring cells when serving HSPA traffic. The last measured aspect takes into account areas with downlink SHO and no SHO in the uplink (labeled as ‘ $\text{SHO}^\downarrow$ , no  $\text{SHO}^\uparrow$ ’ in Table 8.3). This condition, although it hasn’t improved, does not expose the mobile network to malfunctioning, only to reduced throughput within these specific areas. However, the reduced throughput is relative, since there are many cells capable of serving HSDPA data access, as the downlink SHO condition confirms. For this reason, the serving cell should not only deliver HSDPA, but also take care of the user signaling and power control, received in the uplink. Clearly, this is only feasible in areas where uplink coverage is guaranteed.

It is worth mentioning that the simulation results were obtained for a real radio network with actual configuration data. Moreover, the hard constraints imposed to the optimization process (the pilot power limited to the  $\pm 2$  dB interval) ensure that the resulting configuration may be immediately applied to a mobile network. This fact can be contrasted with the spatial distribution of each of the optimized aspects, before and after applying the optimization results, as it is shown in Figures 8.4 and 8.5. The lack of any prominent visual change in Figures 8.4 and 8.5 is a desired consequence of the fine-tuning procedure the network has been exposed to. Still, the improvements are present precisely over the areas that are most exposed to malfunctioning due to unbalanced SHO, e.g., the cell-coverage borders.

## 8.7 Summary

This chapter formally introduced a new optimization problem for 3G networks: the SHO-balancing problem. A characterization of the consequences that unbalanced SHO areas have on the quality of HSPA services was also given. Particularly, tackling the SHO-balancing problem was possible due to the improved performance delivered by the evaluation framework PRATO (see Chapter 6).

Using a extension of the radio-network model presented in Section 7.3, Chapter 7, the penalty scores of the objective function were set according to the configuration and layout of a real mobile network, deployed in Slovenia by Telekom Slovenije, d.d.

The balancing problem has been tackled by three optimization algorithms, namely DE, DASA and SA. All three algorithms were able to improve the given network configuration, being DE the most successful one. The presented results confirm that a great proportion of the SHO areas, that were not balanced before the optimization, were corrected, therefore significantly reducing the possibility of HSPA-service failures. Additionally, radio coverage was improved, while all other essential network services were not altered.

One of the key advantages of the presented method is that it targets the optimization of a deployed network, for which the focus is to fine-tune the existing configuration instead of creating complete new solutions. Furthermore, a deployed network has a great number of hard-constraints that should be taken into account at the optimization stage. Yet, the presented approach is simple and versatile enough to be used in practically any working UMTS network. Moreover, the introduced model is applicable for mobile networks in heterogeneous environments, because it imposes no restrictions regarding cell layout or radio-propagation characteristics, which are completely adaptable through PRATO.

It is important to note that some methods proposed in this chapter have been particularly designed for problems that emerge during the planning of 3G radio networks. Despite this, they may be adapted to other standards, e.g., 2G and 4G, without lose of generality.

## 9 Framework automated tuning

The models and algorithms from Chapter 6 form the basis for the radio-coverage framework used throughout this thesis. Until here, the framework has been studied in the context of optimization-problem solving for radio networks.

In this chapter, the focus is shifted towards radio-network planning activities, and how the framework can aid a network-planning engineer in his or her everyday tasks. The objective is to facilitate refined network planning, the complexity of which is generally beyond the scope of any manual approach.

A central part of a radio-planning tool is its radio-propagation model. But the calculated signal-propagation predictions will be as good as the input data used for their estimation. Moreover, acquiring the necessary information to support the decision making in this context is a challenging task. In practical situations, an emitted signal propagates by interacting with the surrounding environments. Consequently, the ability of a propagation model to adapt to the environment where it is used improves the accuracy of the calculated signal-propagation predictions.

Within this context, this chapter presents two automated-tuning capabilities for PRATO. The first one involves the parameter tuning of the empirical radio-propagation model using a snapshot of field measurements. The second one involves the optimization of clutter losses over different regions of the country, therefore adapting the losses due to land usage to the local conditions of each region. The results of the experimental simulations, performed over three regions of the real LTE network deployed by Telekom Slovenije, d.d., show the suitability of the presented methods to improve the accuracy of the calculated radio-propagation predictions.

The content of this chapter extends the research work published by the author in [22]. The rest of this chapter is organized as follows. Section ?? introduces principles of radio-propagation prediction, and the mathematical model used. The parameter-tuning problem and the analytical approach for tackling it are presented in Section ??, including the simulations performed on the framework and their results. Section ?? concentrates on describing the optimization problem involving the regional adaptation of signal losses due to clutter, including the performed simulations the achieved results. Finally, Section ?? gives an overview of relevant publications, describing how they relate to our work, before drawing some conclusions.

### 9.1 Motivation

With the advent of long-term evolution (LTE) as the fourth generation (4G) in cellular technology, mobile operators are facing the challenges of deploying a new network. LTE follows the well established universal mobile telecommunication system/high-speed packet access (UMTS/HSPA) combo, targeting higher peak data rates, higher spectral efficiency and lower latency [149].

The deployment of a new mobile network is always a challenge for mobile operators, who constantly struggle to find the optimal investment in order to provide a competitive

network in terms of coverage and quality of service. Indeed, coverage planning remains a key problem that all operators have to deal with. It has proven to be a fundamental issue since the deployment of the first GSM networks, more than 20 year ago.

One of the primary objectives of radio-coverage planning is to efficiently use the allocated frequency band for a geographic area to be satisfactorily reached with the radio stations of the network. To this end, radio-coverage prediction tools are of great importance as they allow network engineers to test different configurations before physically implementing the changes. However, to accurately predict the radio coverage of a mobile network is a very complex task, mainly due to the wide range of various combinations of hardware and configuration parameters which have to be analyzed in the context of different environments. The complexity of the problem means that radio-coverage prediction is generally a computationally-intensive and time-consuming task, hence the importance of fast and accurate prediction tools.

Although different mathematical models have been proposed for radio-propagation modeling, none of them excels in a network-wide scenario [137]. Empirical propagation models usually give good results with a limited computational effort. However, for improved accuracy, the model parameters have to be adapted to better fit a specific network or region within it, mainly because of inaccuracies in input data and environmental changes in the network region, e.g. foliage of trees or snow. Consequently, a combination of different parameters is generally needed in order to reliably calculate radio-propagation predictions for particular environments. Moreover, since the number of deployed cells (transmitters) keeps growing with the adoption of modern standards [132], there is a clear need for a radio propagation tool that is able to cope with larger work loads in a feasible amount of time.

To address the afore-mentioned issues, we adapt the parameters of an empirical propagation model to a set of field measurements. The parameter tuning is analytically calculated per cell, in order to increase the accuracy of the calculated predictions. Moreover, we fine tune the signal losses due to land usage (clutter) in a regional basis, using an optimization approach. As a working framework to tackle the presented problems, we use a parallel radio-prediction tool [21], thus showing the suitability of the presented framework for real-world planning and optimization of LTE radio networks. Particularly, we show the tool capabilities to handle several parallel radio-prediction runs using a metaheuristic algorithm and distributed objective-function evaluation, while resolving the clutter optimization problem.

## 9.2 Related work

There are few examples of radio-network simulators in the literature [81, 106, 123, 124, 134, 179]. Most of these tools were developed for academic research, thus not targeting industrial-sized environments.

A Matlab-based LTE simulator was proposed in [106]. It implements a standard LTE downlink physical layer, including Adaptive Modulation and Coding (AMC), multiple users, MIMO transmission and scheduler. Despite being open source and freely available, the fact of being implemented in Matlab makes it restrictive in terms of tackling big problem instances of real networks.

A promising tool from the performance point of view was presented in [124], where the authors implemented a full-stack LTE system in C++. Although the tool has no graphical capabilities for displaying the simulation progress or outcome, they might be included, since the source code is available. In our opinion, the main drawback of this tool is the lack of documentation, which makes it very difficult to continue extending this work without the direct help of some of the original authors.

As an extension to the well-known NS-2 network simulator, Filiposka and Trajanov [52] introduced a module for radio-propagation predictions, which takes the terrain profile into

account. In this case, the authors focus on the relief, leaving out signal loss due to land-usage, which is an important factor when targeting realistic radio-propagation scenarios.

Following an optimization-oriented approach, the authors of [3] study the effects on location accuracy while performing semi-automated optimization of the parameters of a radio-propagation model. While their optimization component improves the accuracy of the radio predictions, it does so requiring human intervention, hence the term semi-automated optimization. In terms of the effects of location accuracy, they conclude that locations with a median accuracy of around 60 m may be used for parameter tuning. Also, they notice that although the model accuracy improved after the parameter tuning, it gives inadequate results when used for predicting radio propagation over distant areas.

### 9.3 Parameter tuning of the radio-propagation model

The effectiveness of the decision-making process during radio-network planning is tightly coupled with the precision achieved by the propagation model used. In order to obtain a radio-propagation model that most accurately reflects the propagation characteristics of the area covered by each radio cell in the network, the parameters of the mathematical model should be adapted to the environment into which it is to be used. The current state-of-the-art method for such parameter tuning depends on existing field-measurement data [3, 178], which are collected in advance for the area covered by the target network cells. Starting from an a-priori best-known set of parameters, empirically calculated by the radio engineers at Telekom Slovenije, d.d., this approach adapts the model parameters so that the deviation of the radio-propagation prediction to a given set of field measurements is minimized.

To calculate the radio-propagation predictions, the empirical radio-propagation model, previously introduced in Chapter 6, Section 6.3.2, is used. Recall that the model contains a vector of adaptable parameters,  $\beta$ . For this reason, this mathematical model is especially appropriate for tuning, since it can be adapted to a given scenario and its local conditions by adjusting the values of the vector  $\beta = (a_0, a_1, a_2, a_3)$ , the elements of which represent

- $a_0$  the reference loss or offset;
- $a_1$  the loss slope due to distance of the receiver from the transmitter;
- $a_2$  the loss slope due to height of the transmitter antenna;
- $a_3$  the loss slope due to the combined effect of the distance and height of the antenna.

The parameter tuning is performed per cell to improve local fitting of the radio predictions, being its resulting solution a vector  $\beta_c$  for the target cell  $c \in C$ .

Also, recall that the model includes an extra term in order to adequately predict signal-loss effects due to foliage, buildings and other fabricated structures. These loss factors are based on the land usage, i.e., clutter data. Here, twelve different clutter categories are recognized. Table 9.1 lists these categories, including their label numbers and descriptions.

#### 9.3.1 Field measurements

In mobile networks, a moving mobile device (or user equipment, UE) constantly performs cell selection/reselection and handover (see Chapter 3, Section 3.1) in order to keep the best possible connection to the network. Within this context, the best connection is selected by measuring the signal strength or quality of the neighboring cells. In LTE networks, the UE measures two parameters from the reference signal of the network, namely the Reference Signal Received Power (RSRP) and the Reference Signal Received Quality (RSRQ).

Table 9.1: Clutter-category label numbers and descriptions for the signal-loss due to clutter of the radio-propagation model.

Clutter category	Description
0	Urban area without buildings, mostly roads
1	Suburban area
2	Urban area
3	Dense urban area
4	Agricultural area
5	Forestall area
6	Swamp area
7	Dry open land area with special vegetation
8	Dry open land area without special vegetation
9	Water area
10	Industrial area
11	Park area

For a certain frequency bandwidth, RSRP measures the average received power over the resource elements that carry cell-specific reference signals. RSRP is applicable in both idle mode (e.g., waiting for a call) and connected mode (e.g., during a call). During the procedure of cell selection/reselection in idle mode, RSRP is used. On the other hand, RSRQ is only applicable when the UE is in connected mode.

The radio-propagation prediction involves calculating the network coverage over a certain region, and thus to the UEs within it. Hence, in the first place, the focus is on accurately predicting the best connection the UE would select in idle mode and the RSRP measurements it uses.

Here, the field measurements representing the RSRP at a given location were collected using a small truck equipped with a spectrum analyzer. The spectrum analyzer is connected to an external omni antenna mounted on the roof of the truck, at roughly 2 m above the ground, taking measurements at a rate of 2 Hz. To accurately establish their location points, a GPS unit was used. These GPS-informed locations have been tested to be compliant with the 60 m limit mentioned in [3]. The measurements cover most of the streets within the target area, with over 300,000 individual points, collected for more than 140 network cells.

To minimize the error impact in measured RSRP, all field measurements are processed so that a single value, the median, is calculated for each of the measured locations. This processing step improves measured-data quality in terms of possible deviations due to external factors, e.g., driving speed. The resulting RSRP is then used to estimate the path-loss prediction at the corresponding location, the resolution of which matches that of the DEM and clutter data.

### 9.3.2 Linear least squares

The approach for tuning of the radio-prediction model is to correlate the field measurements with the predicted received-signal values, i.e., RSRP. The new parameter set originates from the minimization of an error criterion. Similar to [3, 83, 178], the minimization criterion is the squared-sum difference between the predicted and the observed RSRP levels, i.e.,

$$E(\beta_c) = \sum_{i=1}^{m_c} (p_c - pl_c(i, \beta_c) - fm_i)^2, \quad (9.1)$$

Table 9.2: Several properties of Net<sub>8</sub>, Net<sub>9</sub> and Net<sub>10</sub>, the test networks used for the experimental simulation of the framework automated tuning.

	Number of cells	Area (km <sup>2</sup> )	Field-measurement proportion (%)
Net <sub>8</sub>	12	103.74	5.41
Net <sub>9</sub>	130	1298.02	12.02
Net <sub>10</sub>	6	386.38	2.30

where  $E(\beta_c)$  is the observed error for cell  $c$  given the parameters  $\beta_c$ ,  $p_c$  is the transmit power of cell  $c$ , and  $pl_c(i, \beta_c)$  is the path loss of cell  $c$  at the same geographical point of  $fm_i$ , i.e., the  $i$ -th field measurement out of a set of measurements for cell  $c$ , with cardinality  $m_c$ .

A necessary condition for the linear least-squares method to find the global minimum is the linear relationship between the predicted path loss,  $pl_c(i, \beta)$ , and the vector  $\beta_c$  [178]. This condition can be verified by calculating the first derivative of  $E(\beta_c)$  in terms of the components of vector  $\beta_c = \langle a_{0c}, a_{1c}, a_{2c}, a_{3c} \rangle$ , i.e.,  $\frac{\partial E(\beta_c)}{\partial a_{0c}} = 0$ ,  $\frac{\partial E(\beta_c)}{\partial a_{1c}} = 0$ ,  $\frac{\partial E(\beta_c)}{\partial a_{2c}} = 0$ , and  $\frac{\partial E(\beta_c)}{\partial a_{3c}} = 0$ .

### 9.3.3 Simulations

The simulations carried out for this part of our work consist of building the matrices of observed-error values,  $E(\beta_c)$ , for each cell  $c$  in the target network. The linear systems of equations are then individually solved by applying the linear least squares method presented in the previous section, which involves the evaluation of one radio-coverage prediction per cell. Each solved system holds a unique solution for a cell  $c$ ,  $c \in C$ , denoted by the vector  $\beta_c$ .

#### 9.3.3.1 Test networks

The test networks, Net<sub>8</sub>, Net<sub>9</sub>, and Net<sub>10</sub>, are subsets of a real LTE network deployed in Slovenia by Telekom Slovenije, d.d. The path-loss predictions are calculated using PRATO, with a DEM and clutter map of 25 m<sup>2</sup> resolution, and a receiver height of 2 m above ground level. A transmission radius of 16 km defines the coverage prediction area around each network transmitter, thus limiting the path-loss prediction to a distance where it is feasible for a UE to connect to a cell, with a RSRP greater or equal to -124 dBm [115]. At the same time, the selected transmission radius provides enough overlap among neighboring cells to calculate the network coverage over the whole region. Table 9.2 provides more information about the test networks used, showing the number of network cells, the area surface, and the covering proportion of the collected field measurements in terms of the total area of each test network.

Net<sub>8</sub> represents a network deployed over a dominant agricultural area with almost flat terrain, some forests and waters streams. Net<sub>9</sub> is deployed over a densely populated urban area, containing high buildings, parks and avenues. The last one, Net<sub>10</sub>, represents a network deployed over hilly terrain, including some smaller villages and vast forests. It is important to note that the number of deployed cells is directly related to the population density within the region of each test network. This relationship can be derived from the information listed in Table 9.2, where the number of cells is shown along the area of every test network. For a clearer characterization of the test networks in terms of terrain types and their extent, the proportion of each clutter category with respect to the total area of every test network is shown in Table 9.3.

Table 9.3: Clutter-category proportions, expressed in percent, in terms of the surface area of each of the test networks. The category legend is given in Table 9.1.

	Cat.0	Cat.1	Cat.2	Cat.3	Cat.4	Cat.5	Cat.6	Cat.7	Cat.8	Cat.9	Cat.10	Cat.11	Total
Net <sub>8</sub>	0.53	4.53	1.68	0.45	71.89	17.94	0.07	0.00	0.03	2.21	0.67	0.00	100.00
Net <sub>9</sub>	0.91	5.53	9.48	3.84	29.73	48.57	0.14	0.03	0.03	0.76	0.86	0.12	100.00
Net <sub>10</sub>	0.15	3.99	1.14	0.11	26.50	67.13	0.26	0.00	0.00	0.36	0.36	0.00	100.00

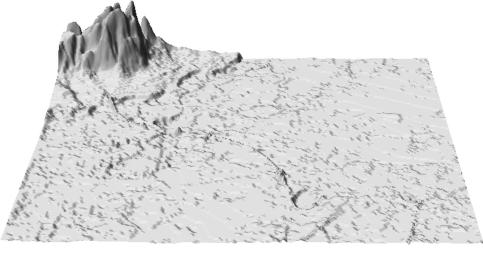


Figure 9.1: Terrain profile of test network Net<sub>8</sub>, dominated by a flat agricultural area.

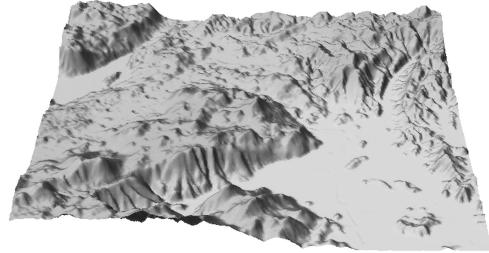


Figure 9.2: Terrain profile of test network Net<sub>10</sub>, dominated by forested hills.

The terrain profiles are most relevant for Net<sub>8</sub> and Net<sub>10</sub>, since none of them contains densely populated urban areas. Note that the terrain shown in Figure 9.1 is mostly flat, since the agricultural area prevails in Net<sub>8</sub>. In contrast, the terrain for Net<sub>10</sub> is dominated by hills, which are mostly covered by dense forests, including some small villages in the valleys (see Figure 9.2).

In the following, the default parameters of the radio-propagation model correspond to the values provided by the engineers of the Radio Network department at Telekom Slovenije, d.d.

### 9.3.3.2 Experimental environment

The simulations were carried out on several computing nodes of the previously presented DEGIMA cluster [72] at the NACC of the Nagasaki University in Japan (see Chapter 6, Section 6.5). Groups of 3, 4, and 34 nodes were used for executing the simulations of the different problem instances, i.e., Net<sub>8</sub>, Net<sub>9</sub> and Net<sub>10</sub>, respectively.

### 9.3.4 Results

The results of applying the linear least squares method to fit the parameters of the radio-propagation model to a set of field measurements are presented in this section. Bar charts were prepared to show the cumulative distribution of the absolute error between the radio-propagation prediction and the field measurements (see Figures 9.3, 9.4, and 9.5). Each bar represents an open interval, expressed in dB, denoting the proportion of points that deviate from the prediction in the given number of dB. For example, in Figure 9.3 (a), it can be observed that the proportion of predicted points differing from the field measurements in 35 dB or more is around 16%, whereas the proportion differing in less than 5 dB is 10%. These values correspond to the test network Net<sub>8</sub> before applying the model-parameter fitting. For comparison, in Figure 9.3 (b), the absolute-error distribution for the same test network is given, but with the model parameters fitted to the available field measurements. Notice how the proportions describing the biggest deviation have dropped to under 5% (35 dB and more)

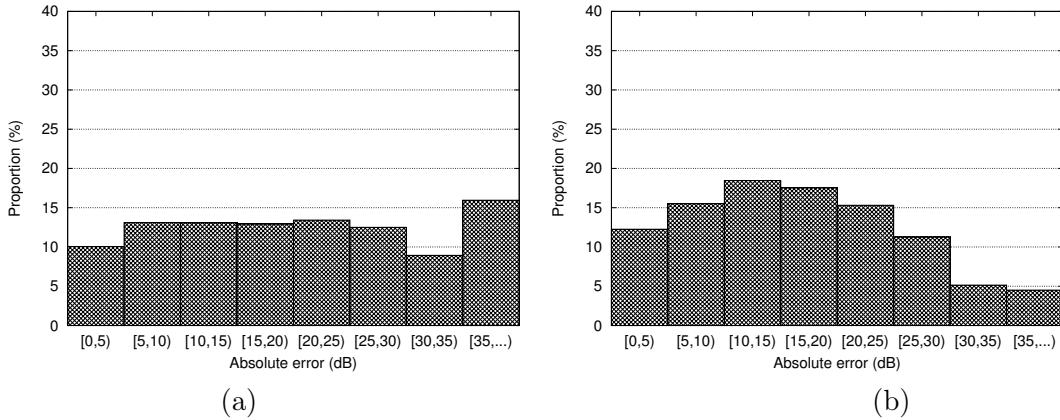


Figure 9.3: Error distribution of the radio prediction for network Net<sub>8</sub>: (a) with default parameter values, and (b) with fitted parameter values.

and to less than 6% (30 dB to 35 dB), respectively. Moreover, it is clear how all proportions improved, raising the bars towards the left-hand side of the chart and lowering them on the right-hand side.

The error distributions of the radio-propagation prediction for test network Net<sub>9</sub> using the default parameters and the fitted ones are given in Figures 9.4 (a) and 9.4 (b), respectively. In this case, the improvement is even more significant than for the previous test network, clearly showing that the tuned propagation model represents the local radio propagation conditions more accurately.

For the last test network, Net<sub>10</sub>, the error distributions are depicted in Figure 9.5 (a) using the default parameters, and Figure 9.5 (b) for the tuned ones. Similar to the first test network, Net<sub>8</sub>, it can be clearly seen how the proportions of highest error deviations have been lowered with respect to those with lower deviation values.

The overall results confirm that fitting the parameters of the radio-propagation model to the field measurements of each network cell significantly improves the quality of the calculated radio-propagation predictions. Indeed, since the default parameters were provided by the radio engineers after following a traditional configuration approach for the whole network, we may conclude that our automated fitting method is not only simpler, but also superior in terms of solution quality.

However, it is worth pointing out the particular reasons behind the considerable better results for Net<sub>9</sub>, when compared to those of Net<sub>8</sub> and Net<sub>10</sub>. Clearly, the relative quantity of available field measurements directly affects the quality of the calculated results, making the least squares approximation rougher and thus less precise for the networks where the field-measurement proportion is lower. See Table 9.2 in Section 9.3.3.1 for a reference of the field-measurement proportions with respect to the area surface of each test network. Similar findings were confirmed by other authors, who worked on the adjustment of radio-propagation models to different environments [83]. For the sake of completeness, it is worth pointing out that some researchers have already started working on different ways on how to improve this aspect [115, 116].

## 9.4 Clutter optimization

In order to further improve the accuracy of the radio-prediction calculation over a given regional environment, the signal losses due to clutter have to be optimized. As it was mentioned before, there are several reasons for the predicted signal-loss values to be inaccurate.

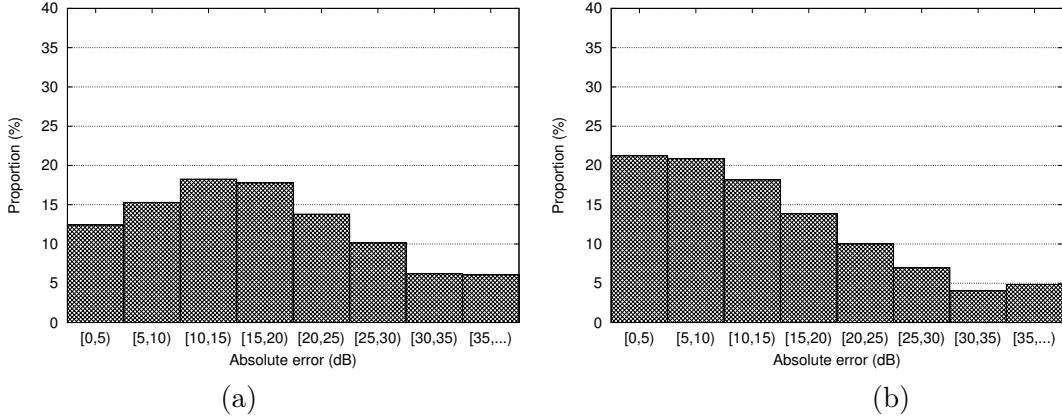


Figure 9.4: Error distribution of the radio prediction for network Net<sub>9</sub>: (a) with default parameter values, and (b) with fitted parameter values.

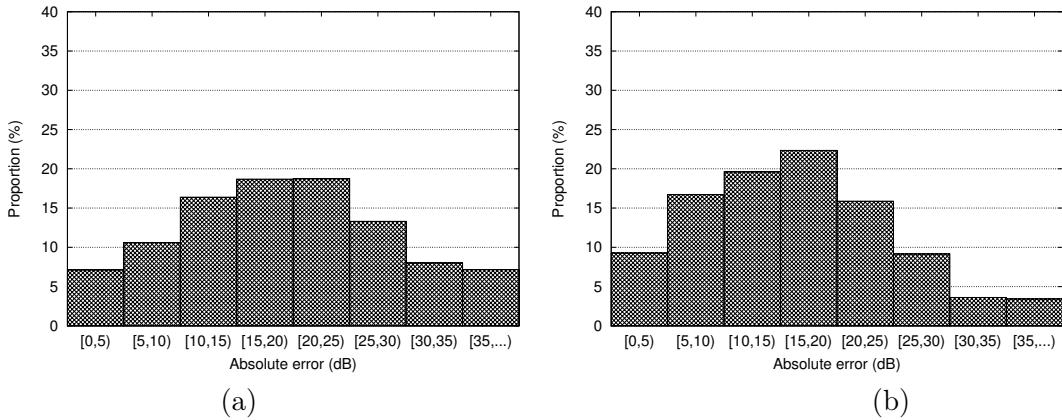


Figure 9.5: Error distribution of the radio prediction for network Net<sub>10</sub>: (a) with default parameter values, and (b) with fitted parameter values.

Seasonal changes are also among them, like tree foliage or snow, demolition or construction of buildings and parks, different kinds of forests or agricultural areas, etc. These changes are only noticeable through regular updates of accurate land-usage data. However, in the short term, updates are only available in the form of feedback by means of field-measurement campaigns.

In the following, a metaheuristic algorithm is used to optimize the clutter losses of several regions within a target radio network. This is done over groups of network cells within different regions of the target network, e.g. agricultural, urban or hilly. In terms of coverage planning, a regional classification of signal losses due to clutter improves the accuracy of the radio-coverage prediction.

In contrast to the parameter tuning of the mathematical model presented in Section 9.3, an analytical approach is not used for tackling this problem. Instead, the DASA metaheuristic algorithm (see Chapter 2, Section 2.3.2) is the tool of choice for optimizing the signal losses due to clutter.

There are several reasons for choosing the DASA as the optimization algorithm in the context of this problem. First, the benefits of metaheuristic algorithms for solving optimization problems, particularly in the context of radio networks, has been demonstrated by several authors [19, 56, 83, 102] in general, and in this thesis in particular (see Chapters 4,

7 and 8). Second, in [92], the authors validated the suitability of the algorithm for solving numerical problems.

### 9.4.1 Optimization objective

The optimization objective consists of adjusting the loss values of the different clutter categories, i.e.,  $pl_{\text{CLUT}}(d_{(x,y)})$  in Equation (6.3) of Chapter 6, Section 6.3.2, according to a set of field measurements of multiple cells covering a given geographical region. The same three data sets used in Section 9.3.3 are used in for the clutter-optimization problem: the first for Net<sub>8</sub>, the second for Net<sub>9</sub>, and the third for Net<sub>10</sub>. Each region is independently optimized, so that the radio-propagation predictions of each set of network cells minimizes the mean-squared error against the field measurements, i.e.,

$$\min f_{\text{clut}} = \sum_{i=1}^{m_c} \frac{(p_c - pl_c(i, \beta_c^*) - fm_i)^2}{m_C} \quad \forall c \in C, \quad (9.2)$$

where  $f_{\text{clut}}$  is the optimization objective to be minimized in each of the three regions,  $p_c$  is the transmit power of cell  $c$ ,  $fm_i$  is the  $i$ -th field measurement of cell  $c$ , the set of which has cardinality  $m_c$ , and  $m_C$  is the number of field measurements of all the cells within a given region. Similar to the least-squares approach in Section 9.3.2,  $pl_c(i, \beta_c^*)$  represents the path loss of cell  $c$  at the same geographical point of the  $i$ -th field measurement, and  $\beta_c^*$  denotes the fitted parameter values of the prediction model for cell  $c$ .

For a reference of the different clutter categories used by the radio-propagation model, see Table 9.1.

### 9.4.2 Differential ant-stigmergy algorithm

As it has been mentioned before, the optimization algorithm we have chosen for the clutter-optimization problem is the DASA (see Chapter 2, Section 2.3.2). The mapping between the clutter-optimization problem and DASA is as follows

$$X_a = \{x_0, x_1, \dots, x_i, \dots, x_{11}\} \quad (9.3)$$

where  $X_a$  is the solution vector of ant  $a$  during the minimization process, and  $x_i$  represents the  $i$ -th clutter category within a given region or network. At the end of every iteration, and after all the ants have created solutions, they are evaluated to establish if any of them is better than the best solution found so far.

### 9.4.3 Simulations

In this case, one simulation round consists of multiple iterations of several steps. An iteration begins by generating a solution vector for each of the ants in the DASA colony. The following step involves the parallel evaluation of the solution vector carried by an ant, i.e., one radio-propagation prediction per worker process. The final step is calculating the objective-function value, as defined in Equation (9.2), before sending it back to the master process for the DASA to generate the next set of solutions. Figure 9.6 depicts the way PRATO performs the parallel objective-function evaluation over the worker processes, while the optimization algorithm runs on the master process.

As opposed to the least-squares case, presented in Section 9.3.3, a much larger number of evaluations is needed for this kind of optimization. Therefore, it is essential to exploit the parallel nature of the framework in order to simultaneously evaluate the radio-coverage

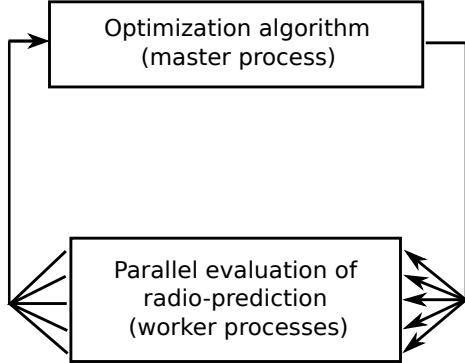


Figure 9.6: PRATO architecture and data flow during the clutter-optimization phase. The optimization algorithm runs on the master process, while the radio-propagation predictions of the involved network cells run in parallel over several worker processes.

prediction of multiple cells. Otherwise, a metaheuristic approach would not be feasible, due to the computational time required to reach a reasonable solution.

The default clutter-loss values (in dB) for each clutter category, which are listed in the second column of Table 9.4, were provided by the radio experts of the Radio Network department at Telekom Slovenije, d.d. These values were empirically calculated using a classical (i.e., manual) approach, the results of which are regularly loaded into a commercial radio-planning application in order to validate the radio coverage at a network-wide level. The radio experts also suggested limiting the optimized clutter-loss values to a maximum of 40 dB.

The stopping criteria for the optimization runs involves limiting the maximum number of objective-function evaluations. In this sense, the limits for  $\text{Net}_8$  and  $\text{Net}_{10}$  were set to 200, whereas for  $\text{Net}_9$  the value was 500, since this network contains the largest number of cells. Overall, the framework completed 48,000 objective-function evaluations, i.e., 576,000 radio-coverage predictions for  $\text{Net}_8$  and 288,000 for  $\text{Net}_{10}$ , whereas for  $\text{Net}_9$  the number of objective-function evaluations was 120,000, for a total of 15,600,000 radio-coverage predictions.

Regarding the parameters that control the behavior of the DASA, they were set to the following values

- $m = 240$ , the number of ants;
  - $b = 10$ , the discrete base;
  - $q = 0.2$ , the pheromone dispersion factor;
  - $s_+ = 0.01$ , the global scale-increasing factor;
  - $s_- = 0.01$ , the global scale-decreasing factor; and
  - $e = 1.0^{-2}$ , the maximum parameter precision.

#### 9.4.4 Results

The results achieved by the optimization process are shown in Table 9.4. The solutions are given for each of the test networks, along with the empirically-calculated (default) loss values due to clutter. Hyphens represent clutter categories for which there are no field measurements available. Consequently, it is not possible to evaluate the objective-function for them.

Table 9.4: Clutter-category losses after the optimization. The default losses for each clutter category are given along the solutions for each of the test networks. All values are expressed in dB.

Clutter category	Default	Net <sub>8</sub>	Net <sub>9</sub>	Net <sub>10</sub>
0	5.0	13.71	11.30	17.90
1	15.0	12.39	16.67	-
2	13.0	16.04	17.04	15.69
3	28.0	19.59	18.01	23.00
4	12.0	11.48	9.71	10.80
5	20.0	16.26	11.62	16.26
6	15.0	-	-	-
7	8.0	-	13.49	-
8	5.0	-	13.50	-
9	1.0	17.50	5.60	-
10	20.0	8.26	16.75	16.63
11	8.0	-	18.93	-

The optimized loss for the first clutter category, 0, representing urban area without buildings, is larger than the default value in all three networks. This may be attributed to the fact that these areas are not completely open, mostly surrounded by forests (Net<sub>8</sub> and Net<sub>10</sub>) and buildings of different sizes (Net<sub>9</sub>). As for the category 1, representing suburban area, the value for Net<sub>8</sub> is lower than the default one, mainly because this network is deployed over a predominant agricultural area, i.e., suburban areas are less dense here. On the other hand, the value for Net<sub>9</sub> is larger, indicating a building density above the average, whereas for Net<sub>10</sub>, the value could not be calculated due to lack of measurements. For the category 2, representing urban area, the optimized values are above the default ones, clearly showing an underestimation of the manual approach. However, the relation among the clutter losses corresponds with the population density in each of the regions, being Net<sub>10</sub> the less dense urban area among the three networks. The optimized values of the category 3, representing dense urban area, are lower than the default ones. This indicates that the dense urban areas in these regions have a lower density than the average case. Representing agricultural area, the category 4 gets a value very close to the default one for Net<sub>8</sub> and Net<sub>10</sub>, whereas for Net<sub>9</sub>, the value is lower, indicating that this type of land is mostly open near the city, e.g., without plantations. As for the category 5, representing forests, the results correspond with the type of forest that dominates each of the test-network regions. Namely, Net<sub>8</sub> and Net<sub>10</sub> are dominated by dense forests presenting leave foliage, whereas in Net<sub>9</sub> the forests are mostly coniferous and more sparse. Keeping the most of the default loss values for the categories 6, 7 and 8, the results of the next category, 9, representing water, indicate creeks and rivers in these area are almost entirely surrounded by forests (Net<sub>8</sub>) or buildings (Net<sub>9</sub>), since none of the regions lays by the sea. As for the industrial area, denoted by the clutter category 10, lower loss values than the empirically-calculated defaults appear. This indicates the presence of sparse industrial buildings in Net<sub>8</sub>, and a higher density of mostly commercial buildings for Net<sub>9</sub> and Net<sub>10</sub>. The last clutter category, 11, could not be calculated for Net<sub>8</sub> and Net<sub>10</sub> due to lack of field measurements.

Notice that the relationships among the different clutter categories is correctly kept for the three test networks. For example, it can be observed that the clutter loss for dense urban area (category 3) is higher than the values of the urban area (category 2), as well as of the agricultural area (category 4). Hence, the results reflect physically feasible losses, despite the higher deviation from the default losses shown by the category 2, and the lower

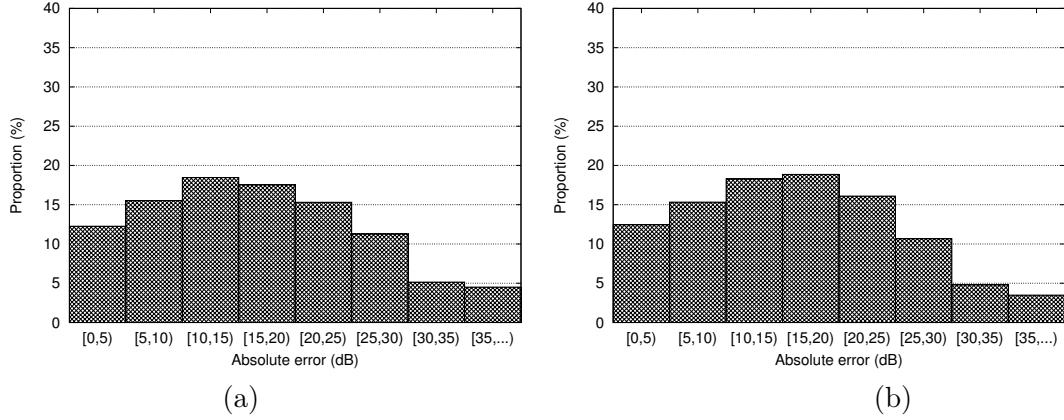


Figure 9.7: Error distribution of the radio prediction for network Net<sub>8</sub>: (a) with fitted parameter values and default clutter losses, and (b) with fitted parameter values and optimized clutter losses.

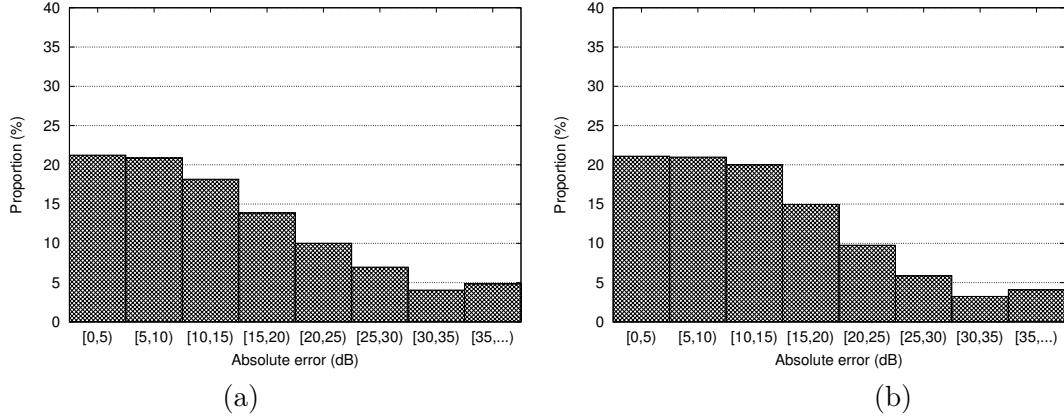


Figure 9.8: Error distribution of the radio prediction for network Net<sub>9</sub>: (a) with fitted parameter values and default clutter losses, and (b) with fitted parameter values and optimized clutter losses.

deviation for the category 3, again with respect to the default losses. Such relations hold for different categories of all three test networks, strongly suggesting the correctness of the applied optimization approach and the evaluation methodology used. Consequently, it can be confirmed that the combination of PRATO and a metaheuristic algorithm is applicable for performing optimization of clutter losses, since it is capable of reflecting the physical phenomena appearing in real-world conditions and improving the quality of radio-propagation predictions for three, geographically different radio-network instances.

Similar to Section 9.3.4, bar charts show the cumulative distribution of the absolute error between the signal-propagation prediction and the field measurements (see Figures 9.7, 9.8, and 9.9). Figure 9.7 (a) depicts the error distribution of the prediction for test network Net<sub>8</sub> using the fitted model parameters, whereas Figure 9.7 (b) shows the absolute-error distribution for the same test network, but using the optimized clutter losses (see Table 9.4, column Net<sub>8</sub>). Notice how the error distributions show an improvement when the optimized clutter losses are used, lowering the biggest (right-most) deviations even more.

The error distributions of the radio-propagation predictions for test network Net<sub>9</sub> using fitted parameters and default clutter losses, and fitted parameters with optimized clutter losses are shown in Figures 9.8 (a) and 9.8 (b), respectively. Similar to Net<sub>8</sub>, the improvement

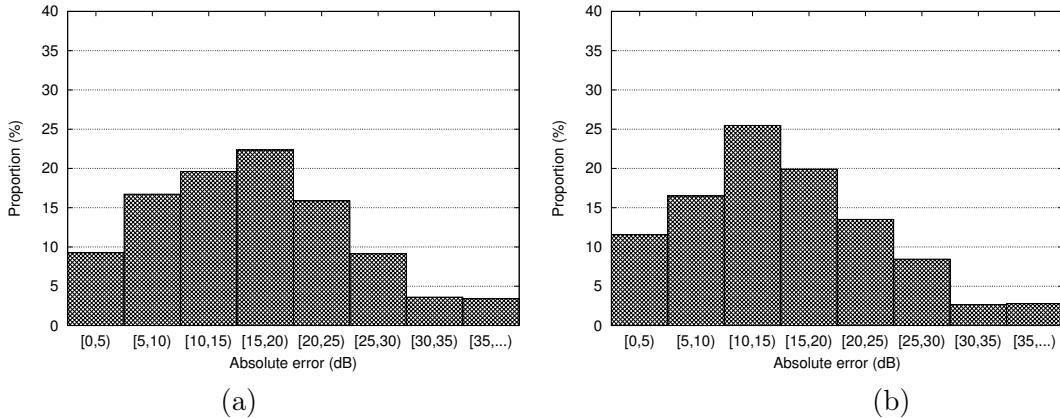


Figure 9.9: Error distribution of the radio prediction for network Net<sub>10</sub>: (a) with fitted parameter values and default clutter losses, and (b) with fitted parameter values and optimized clutter losses.

appears in the biggest deviations, since their values are lower than when using the default clutter losses.

For the last test network, Net<sub>10</sub>, the error distributions are depicted in Figure 9.9 (a) using the default clutter losses, and Figure 9.9 (b) for the optimized ones. Again, fitted model parameters are used for both simulation sets. In this case, a more significant improvement than for the other two networks appears, clearly showing the favorable effect of the optimization process. Moreover, this result suggests that some of the default clutter losses considerably fail in representing the actual physical conditions in the geographical region of this network.

The presented results confirm that the optimization of clutter losses with respect to field measurements improves the quality of the calculated radio-propagation predictions. Considering the default clutter losses were empirically calculated by the radio engineers for the whole network, the convenience of the automated optimization procedure is clear. Indeed, these advantages are a consequence of a simpler method that automatically delivers radio-predictions of superior quality, thus accurately representing the physical properties of a given environment.

Because of the stochastic nature of the optimization algorithm, the results of 30 independent runs were collected in order to have enough data for them to be statistically relevant. In other words, the robustness of the solutions, that have been presented in the previous section, is analyzed here.

To this end, Table 9.5 shows the solutions reached by the DASA for each of three test networks. The calculated signal losses are depicted with the minimum, maximum and average values for every clutter category, along with their standard deviations. Again, hyphens represent clutter categories for which there are no field measurements available, and thus they could not be optimized. In order to easily visualize the data shown in Table 9.5, box plots based on the same values are provided in Figure 9.10. It can be observed that the standard deviation is low for almost all optimized categories, indicating a consistent convergence of the optimization algorithm. However, some exceptions are noticeable, e.g., the category 9 in Net<sub>8</sub> and Net<sub>9</sub>, representing water. In this case, there is a lower density of field measurements near water trails since none of the test networks lays by the sea. Regarding the categories 1 (suburban area), 7 and 8 (dry open land area) of Net<sub>9</sub>, we again find very few field measurements over this areas, thus higher standard-deviation values appear. Based

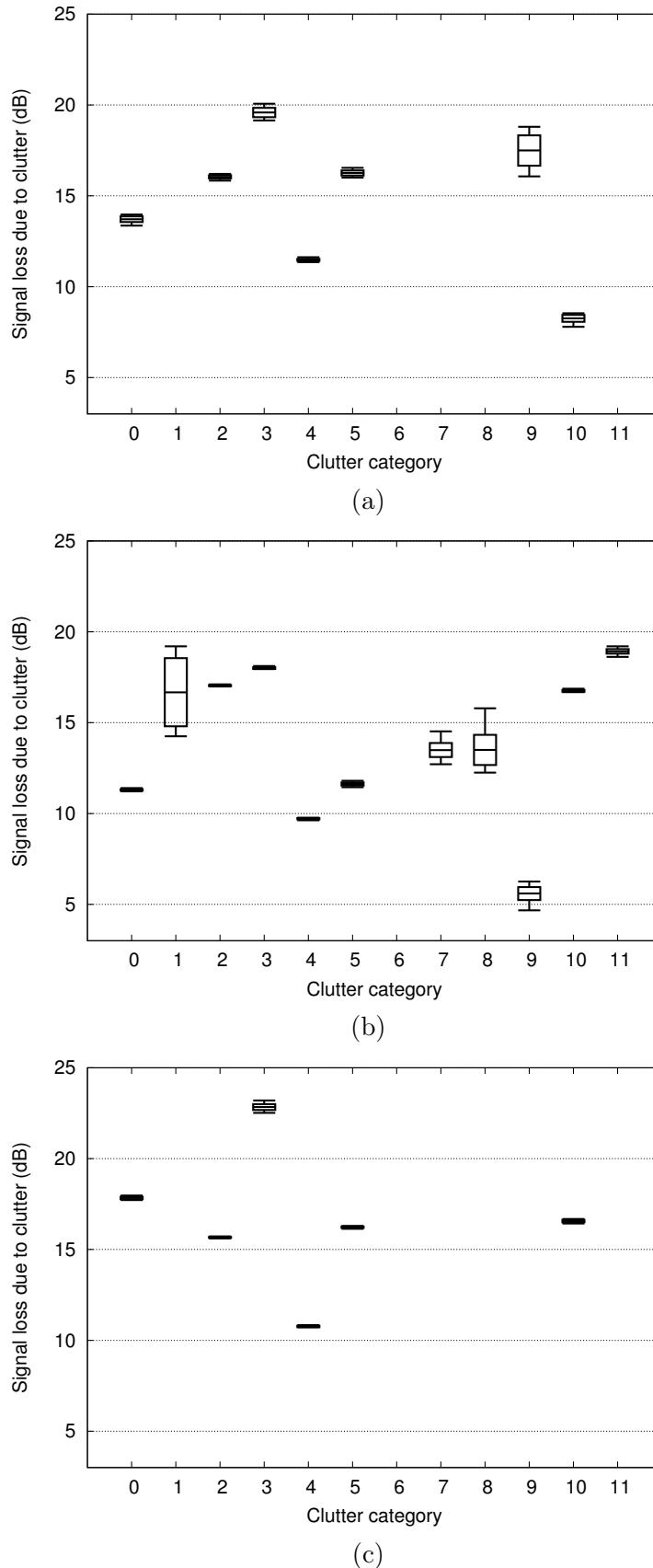


Figure 9.10: Box plots representing the statistical-analysis values of Table 9.5, for the solutions of the clutter-optimization process of each test network: (a) Net<sub>8</sub>, (b) Net<sub>9</sub>, and (c) Net<sub>10</sub>.

Table 9.5: Statistical analysis of the optimization solutions for each test network. All values are expressed in dB. The corresponding box plots are depicted in Figure 9.10.

Cat.	Net <sub>8</sub>				Net <sub>9</sub>				Net <sub>10</sub>			
	Min	Max	Avg	St.dev.	Min	Max	Avg	St.dev.	Min	Max	Avg	St.dev.
0	13.36	13.97	13.71	0.15	11.22	11.40	11.30	0.04	17.72	17.85	17.90	0.07
1	-	-	-	-	14.25	19.20	16.67	1.87	-	-	-	-
2	15.84	16.21	16.04	0.08	16.99	17.11	17.04	0.03	15.64	15.72	15.69	0.03
3	19.15	20.07	19.59	0.25	17.95	18.12	18.01	0.04	22.68	23.20	23.00	0.16
4	11.35	11.62	11.48	0.05	9.63	9.77	9.71	0.03	10.73	10.84	10.80	0.03
5	16.00	16.54	16.26	0.14	11.45	11.80	11.62	0.08	16.19	16.30	16.26	0.04
6	-	-	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	12.71	14.52	13.49	0.39	-	-	-	-
8	-	-	-	-	12.25	15.79	13.50	0.83	-	-	-	-
9	16.07	18.80	17.50	0.83	4.67	6.26	5.60	0.35	-	-	-	-
10	7.79	8.54	8.26	0.19	16.68	16.87	16.75	0.05	16.50	16.68	16.63	0.07
11	-	-	-	-	18.62	19.20	17.04	0.13	-	-	-	-

on these findings, the standard deviation of the optimized clutter losses can be considered a quantity indicator of the field measurements required by the optimization process, i.e., a higher standard deviation denotes more field measurements are needed to optimize the target clutter category.

## 9.5 Summary

This chapter shown the suitability of PRATO as a network-planning tool by tackling two coverage-planning problems, which were tested over the newly deployed LTE network in Slovenia. The first one involved the parameter tuning of the empirical radio-propagation model using a snapshot of field measurements. The second one considered the optimization of the clutter losses over different regions of the country, therefore adapting them to the local conditions of the geographical region of each network.

The combination of the afore-mentioned techniques with PRATO provides an environment-adaptable framework for radio-network planning. This delivers a tool with a considerable improvent of the solution accuracy of the analyzed instances, especially if compared to traditional, i.e., manual or semi-automated, coverage-planning methods.

Additionally, the simulation results indicate that PRATO is applicable for planning and optimization of real-world radio networks, since it is capable of tackling bigger problem instances in a feasible amount of time. Particularly, several of the presented instances show a large computational-time complexity, which is beyond reach for a serial implementation of an automated approach. Moreover, the parallelization capabilities provided by the framework create new problem-solving possibilities. Indeed, even computational-intensive tasks such as stochastic optimization, which involves millions of radio-prediction evaluations, are feasible if PRATO is used.

## 10 Performance assessment within real network-planning scenarios

In this chapter, real-world, network-planning activities on different radio networks are presented. The objective is to compare PRATO with a radio-prediction enterprise tool in terms of quality and performance. To this end, the engineers of the Radio Network department at Telekom Slovenije, d.d., provided us with their radio-prediction industrial software and guidelines to perform some typical radio-planning activities.

### 10.1 Measurements and simulation comparison

We have also developed two modules for the evaluation of the accuracy of the designed radio coverage prediction tool with respect to the actually measured results. The first module, db.CompareResults, compares the calculated values with the values obtained during a field measurement campaign. The module reads each row of the text file with field measurement data and saves the x and y coordinates, the measured signal level and the name of the cell. The module first maps the coordinates to the nearest coordinates in the database (as they do not necessarily coincide), finds the corresponding row in the database and then extracts the received signal level for the required cell. The data for the measured and calculated signal level, their difference in decibels along with the location, the used path loss prediction model and the cell name are finally written to the output text file (Table 3).

The second module, r.EvaluateSimulations, compares the field measurements and the strongest simulation signal levels, neglecting the information about the serving cell. The module output is a textual file including the same fields as the output of the db.CompareResults module. The module enables comparison of field measurements with simulation results calculated with the designed radio coverage prediction tool or the commercial tool TEMS. Because of different input raster files (a GRASS raster file includes values for maximal received power while a TEMS raster file contains path loss values) additional selection is done with the “GRASS MaxPower raster”. In the case of the TEMS input raster file, an average transmitted power value must be entered for the receive power calculation.

### 10.2 Coverage-prediction performance analysis

The performance and accuracy of the developed modules for radio signal coverage prediction was investigated by comparing simulation results and field measurements. The reference values were obtained by comparing field measurements with simulation results acquired from the professional radio signal coverage prediction program TEMS. In both simulation tools we used the modification of the Okumura-Hata propagation model [23].

The performance of the new software package was investigated for different types of networks (GSM, UMTS) and terrains (hilly and almost flat rural, urban, and suburban). The evaluation of the developed software for different frequency bands is presented first, followed by an analysis for a different terrain type.

The accuracy of the GRASS prediction software can be verified from the charts on Fig. 6, 7 and 8. On the left side, the charts comparing the measurements and calculations with the GRASS radio coverage prediction software are depicted, while graphs showing the comparison between the measurements and calculations with the TEMS software package are on the right.

Fig. 6 and 7 are presenting the simulation results from both tools and the field measurements in suburban environment for 900MHz and 2040MHz. Received power charts clearly shows that the simulation results match the measured values rather well. Slightly better agreement can be perceived in the 900MHz frequency band (Fig. 7). The deviation among the measurements and simulations for both software applications is depicted in the second raw of diagrams in Fig. 6 and 7. It is evident that the difference between the diagrams on the left- and on the right-hand side for both frequencies is minor. Thus, it can be concluded that the results from the developed radio coverage tool are comparable with the results from the TEMS application and are independent from the used frequency band.

Additional analyses were done on different terrain types. The analyses for the flat rural environment are depicted in Fig. 8. The curves on the charts showing the difference between the measurements and simulations for both software applications have similar course. This confirms applicability of the developed software also for arbitrary terrain types.

The developed radio coverage software gives similar results as the professional TEMS software irrespective of the operational frequency or chosen terrain type. The computed values are comparable also for different distances between the base station and the receiver. Some negligible differences between the results originate from the fact that the implemented path loss model in the TEMS software used in the simulation is not entirely available and thus cannot be realized in the GRASS software in a completely identical way.

Additionally, execution performance of the developed modules was evaluated in terms of the required processing times on our system (processor Intel Core2 Quad CPU 2,66GHz, disk WD2500KS, OS Linux RHEL5). The simulated configuration included eight transmission antennas on four locations (base stations), therefore requiring four model and eight sector computations. Two different geographic regions were used: a small one, “Ljutomer”, encompassing all transmission locations (15 x 13km, resolution 25m) and a large one “Slovenia” (whole Slovenia, 285 x 185km, resolution 100m). The effective transmission radius was limited to 10km. The results for single core execution are given in Table 4. The hataDEM model was not simulated for the whole Slovenia region since its clutter map was not available to us. The r.MaxPower module was not run with DBF database on the whole Slovenia region since the internal GRASS DBF processing is very memory inefficient, keeping the whole database in the main memory and hence running out of memory for large regions.

### 10.2.0.1 Performance analysis

In the following, in order to confirm the benefits of PRATO as a parallel simulation framework for radio networks, we contrast the running time of a serial implementation deployed in one host against PRATO over multiple processors.

We have adapted the serial implementation for execution in a feedback loop together with the DASA. To this end, the implemented code dealing with input-data load and set up has been moved outside the loop, so that the objective-function evaluations alone, including the radio-propagation predictions, are executed in each iteration of the optimization algorithm. This way, only the running time is compared as a performance metric, excluding load time from the analysis, since both implementations use the same input-data sets.

The running-time measurements for the adapted serial implementation were collected on a host featuring an i7-2600K quad-core processor. The measurements for PRATO were taken

Table 10.1: Measured and estimated running times of the clutter-optimization process using the adapted serial implementation. Times are depicted in seconds for the three test networks.

	Net <sub>1</sub>	Net <sub>2</sub>	Net <sub>3</sub>
Fastest radio-coverage prediction	0.526333	0.526333	0.526333
Estimated objective- function evaluation time	6.315996	68.423290	3.157998
Estimated optimization- process time	$3.031678 \cdot 10^5$	$8.210794 \cdot 10^6$	$1.515839 \cdot 10^5$

over multiple processors, with only the master process running on an i7-2600K CPU. Please refer to Section 9.3.3.2 for a reference of the hardware configuration and process layout used during the experimental simulations.

We have performed 300 independent time-measurements of the adapted serial implementation. From this set, we pick the minimum (fastest) running time of a single radio-propagation prediction (0.526333 sec) in order to linearly estimate the running time of the complete optimization process for each of the three test networks. Time-measurement gathering for the running time of PRATO over multiple processors has been performed during the simulation runs presented in the previous section, from which the minimum time is used for comparison. All gathered running-time measurements and estimations for the adapted serial implementation are depicted in Table 10.1. We may observe that the estimated running times of the optimization process using the adapted serial implementation are unfeasible for practical use. Specifically, the smallest network, Net<sub>3</sub>, shows an estimated running time of over 42 hours, while the biggest network instance, Net<sub>2</sub>, raises over 2,200 hours!

The best running times of the parallel implementation with multiple processes deployed on the computer cluster are shown in Table 10.2. The speed-up factors for each test network, ranging from 4.50 to 119.58, are also depicted in the third row. These values are an estimation, since they are based on the estimated running times of the adapted serial version. Despite this, we may clearly notice how the estimated efficiency values, calculated as the speed-up factor over the number of processes, increase with the problem size, thus improving the exploitation of the computing resources of the cluster. This behavior corresponds with the speed-up measurements presented in Figure ???. Nevertheless, the obtained speed-up factors are substantial, lowering the running time of the optimization process from 84.2 to 8.7 hours for Net<sub>1</sub>, from 2,280.8 hours to 19.1 hours for the biggest instance, Net<sub>2</sub>, and from 42.1 to 9.3 hours for Net<sub>3</sub>, making the presented optimization approach feasible for practical use in real-world radio networks.

### 10.3 Summary

Precise and efficient planning of the wireless telecommunication systems requires efficient and exact radio signal coverage calculations. The high price and limited functionalities of the existing professional network planning tools compels to look for alternative solutions.

Table 10.2: Measured running times (in seconds) of the clutter-optimization process using PRATO over multiple processors. The number of deployed worker processes and speed-up factors are also depicted for the three test networks.

	Net <sub>1</sub>	Net <sub>2</sub>	Net <sub>3</sub>
Fastest optimization-time	$3.154711 \cdot 10^4$	$6.866361 \cdot 10^4$	$3.368531 \cdot 10^4$
Number of worker processes	12	130	6
Estimated speed-up	9.61x	119.58x	4.50x
Estimated efficiency	0.80	0.92	0.75

The needs can be fulfilled using an open-source system which gives possibilities to improve the existing models based on measurements, or to develop entirely new path loss prediction models. This paper presented a radio signal coverage prediction software tool developed for the open-source GRASS system. After a short introduction of the GRASS GIS system, a detailed description of the coverage prediction software was given. The tool enables a high level of flexibility and adaptability. It is composed of several GRASS modules for path loss calculation, a sectorization module, a module for radio signal coverage calculation, and additional modules for preparing input data and analyzing simulation results. Modules can be used individually or through the r.radcov module, written in Python, which interconnects individual modules into a complete radio signal propagation software package. At the end, the developed software was evaluated by comparing with the field measurements and simulation results obtained from a professional software application. The radio signal coverage prediction software implementation was quite straightforward, as API is well developed and documented. The set of built-in C functions is adequate. The possibility to study parts of the already implemented code is also very helpful. Extensive performance analyses showed satisfactory results. Compared to a professional network planning tool, the computation speed is slightly lower while the result accuracy is completely comparable irrespective of the terrain type or operational frequency. For better agreement between simulations and measurements, additional model tuning will be performed. In our future work, we also plan to expand the functionalities of the developed software package and build additional path loss modules for the urban and hilly rural environments that will also include the elements of ray tracing techniques and additional environment data [29]. The achievement made so far represents a strong base for future work and is interesting both from the point of view of researchers as well as network developers. The whole source code of the radio signal coverage prediction tool together with detailed instructions will be publicly available at <http://commsys.ijes.si/en/software/grassradiocoverage>. The tool can be freely used, modified and upgraded with new path loss modules.

## 11 Conclusion and further work

The methods and tools introduced in this thesis have a direct correlation with the simulation and optimization of radio networks in general and with those of the 3G and 4G in particular. We will introduce the steps necessary to mitigate the problem of experimental reproducibility found in most published works, by simplifying setup, execution, and sharing of experimental results. With the development of the framework, we will evaluate and assess the possibilities it offers as a support system for simulation and optimization problems of radio networks. By including parallel programming techniques for computer clusters and GPUs, we intend to go beyond the classical methodology provided by previous works, taking advantage of the inherent parallelism of some optimization techniques. We will also be looking into the application of many advances in HPC that should provide the framework with the computing power needed to improve the simulation process, thus enhancing its scalability to support real-world 3G radio networks. Finally, since we believe that this work will only become truly productive through the cooperation and long-term development of the scientific and engineering community, we will be releasing the source code, algorithms, documentation, and data to the public domain. This way, anyone will be able to use and to extend the framework for their own needs. Encouraging cooperation and sharing of experimentation-related tools and data should be a common goal from which everyone will benefit.

We have presented an open-source simulation framework for coverage-planning and optimization of radio networks (PRATO)<sup>1</sup>.

Encouraged by the favorable results, further work will include abstracting the introduced MWD principle into a multi-purpose parallel framework such as Charm++ [86], which provides a functionality for overlapping execution and communication, as well as fault tolerance.

Encouraged by the favorable results, further work will include abstracting some of the introduced principles and methodology into a multi-purpose library for parallel simulation of radio-coverage predictions, which shall be published as free and open source software. The aim of such tool is to ultimately validate the suitability and usefulness of the presented framework.

Furthermore, in the context of radio-coverage planning activities carried out at the Radio Network department of Telekom Slovenije, d.d., supplementary testing of PRATO as a coverage-planning tool is currently being conducted. So far, the performed predictions show evidence of faster and high-quality results when compared with well-established industrial software tools.

In addition, as PRATO is also a free and open-source software project<sup>2</sup>, it can be readily modified and extended to support, for example, other propagation models and post-processing algorithms. This characteristic provides it with a clear advantage when compared to commercial and closed-source tools.

Comparison of our experimental results with other algorithms dealing with the same and similar problems would be useful. However, this task is not straightforward, since the results

---

<sup>1</sup>The source code is available for download from the corresponding author's home page, <http://csd.ijs.si/benedicic/>

<sup>2</sup>The source code is available for download from <http://cs.ijs.si/benedicic/>

of several works (e.g. [58, 164]) depend on black-box evaluations, making experimental association very difficult, if possible at all.

All in all, we consider that the present work provides a robust foundation for future work on grid-based metaheuristics with expensive objective-function evaluation.

In future work, we will consider further analysis of our parallel-agent approach, including experimentation with different parameters, in order to gain better understanding of the dynamics leading the metaheuristic during the search process. Multi-GPU environments present an interesting possibility, where evaluator(s) and worker agents are run on separate GPU devices.

To further improve the presented results, dynamic effects, such as fast power control, should be included in the simulations, particularly for recognizing dynamic functionality like SHO in a WCDMA mobile system. Another extension of the current work is to incorporate antenna tilt as an additional objective of the optimization process. This should certainly include experimentation with models and algorithms that support multiobjective optimization.

It is important to note that some methods proposed in this thesis have been particularly designed for problems emerging in radio planning of 3G networks. Despite this, they may be adapted to other standards, e.g. GSM or LTE, without loss of generality. Moreover, some of them may even be applied in other research areas, e.g., parallelization techniques in the area of geographical information sciences.

As it was mentioned in the introduction, it is important to understand that the models used in scientific simulations and engineering never offer a perfect model of the system they represent, but only a subset of its composition and dynamics. For this reason, experimentation and expert observation will always be essential as reference points for understanding the studied phenomena. Consequently, problems categorized as of large size and of considerable complexity represent a challenge, because of the different involved disciplines and the degree of difficulty of their modeling. Radio networks, in particular those of the third and fourth generations, fall under this characterization.

It is very important to develop a feeling for the properties, advantages and drawbacks of the respective methods. Moreover, the recommendations of radio experts regarding the interpretation of the solutions and the feedback from everyday network operation are an essential input for establishing high-quality optimization methods. In this sense, and based on our own experience, the expert's advice is irreplaceable and a most valuable contribution to the research work.

## 11.1 Scientific contributions

The work in this thesis has led to the following original contributions to science:

1. A unified anomalous and suspicious behavior detection framework: We proposed a unified framework for detection of anomalous and suspicious behavior that can be observed from complex, spatio-temporal sequential data generated by an agent moving in a physical environment. The framework incorporates several components to address the main challenges and is demonstrated empirically in three studies.

2. Contribution to anomalous and suspicious behavior detection: We gave the first clear problem definition and established a theoretical framework for anomalous and suspicious behavior detection from agent traces to show how to optimally perform detection. We discussed why detection error is often inevitable and proved the lower error bound. We further provided several heuristic approaches that either estimated distributions required to perform detection or directly rank the behavior signatures using machine-learning approaches.

The contributions of this thesis to the fields of telecommunications and computer sciences include the following:

- State-of-the-art overview of optimization methods for 3G radio networks.
- Design and development of a framework that provides an open environment for radio network simulations, implemented for execution on computer clusters and GPUs. The framework will allow the scientific community to share a common domain to run the simulations needed by modern optimization methods, since most currently available simulation tools are proprietary and therefore unsuitable for experimental reproducibility.
- Improvement of quality and speed of renowned mathematical models, used for radio propagation predictions, by applying parameter optimization and parallelization techniques. The expected speed improvement should be of at least one order of magnitude.
- Proposal of a new algorithm, based on autonomous agents, to solve the service coverage problem. The solved problem instances should be of bigger size than ever solved in the literature and reach equal or better quality thereof. This should make our approach applicable for large real-world problem instances and data sets.
- Identification of a new optimization problem in 3G radio networks that deals with soft-handover alignment of downlink and uplink areas. By solving this problem, we should avoid abnormal network functioning in areas where there is soft-handover capability in the uplink, but none in the downlink. So far this problem has been solved manually by radio experts.
- Empirical comparison of the proposed metaheuristic algorithm against the existing state-of-the-art optimization algorithms on the soft-handover alignment problem.



## 12 Bibliography

- [1] 3GPP. Functionality in early GSM releases. <http://www.3gpp.org>, accessed May 2009.
- [2] 3GPP. General UMTS architecture, v4.0.0. <http://www.3gpp.org>, accessed May 2009.
- [3] Erik Aarnæs and Stian Holm. Tuning of empirical radio propagation models effect of location accuracy. *Wireless Personal Communications*, 30(2-4):267–281, 2004.
- [4] Yassir A Ahmad, Walid A Hassan, and Tharek Abdul Rahman. Studying different propagation models for LTE-A system. In *Computer and Communication Engineering (ICCCE), 2012 International Conference on*, pages 848–853. IEEE, 2012.
- [5] S Akhter, K Aida, and Y Chemin. Grass gis on high performance computing with mpi, openmp and ninf-g programming framework. In *Proceeding of ISPRS 2010*, 2010.
- [6] S. Akhter, Y. Chemin, and K. Aida. Porting a GRASS raster module to distributed computing Examples for MPI and Ninf-G. *OSGeo Journal*, 2(1), 2007.
- [7] Edoardo Amaldi, Antonio Capone, and Federico Malucelli. Planning umts base station location: Optimization models with power control and algorithms. *Wireless Communications, IEEE Transactions on*, 2(5):939–952, 2003.
- [8] Edoardo Amaldi, Antonio Capone, and Federico Malucelli. Radio planning and coverage optimization of 3G cellular networks. *Wireless Networks*, 14(4):435–447, August 2007.
- [9] Edoardo Amaldi, Antonio Capone, and Federico Malucelli. Radio planning and coverage optimization of 3G cellular networks. *Wireless Networks*, 14(4):435–447, 2008.
- [10] Marc P Armstrong, Mary Kathryn Cowles, and Shaowen Wang. Using a computational grid for geographic information analysis: a reconnaissance. *The Professional Geographer*, 57(3):365–375, 2005.
- [11] Mikhail J Atallah. *Algorithms and theory of computation handbook*. CRC press, 2002.
- [12] Fredrik Athley and Martin N Johansson. Impact of electrical and mechanical antenna tilt on lte downlink system performance. In *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, pages 1–5. IEEE, 2010.
- [13] Mehmet Aydin, Jun Yang, and Jie Zhang. *Applications of Evolutionary Computing*, volume 4448 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [14] J-FM Barthelemy and Raphael T Haftka. Approximation concepts for optimum structural design - a review. *Structural optimization*, 5(3):129–144, 1993.
- [15] Mokhtar S Bazaraa, Hanif D Sherali, and Chitharanjan Marakada Shetty. *Nonlinear programming: theory and algorithms*. Wiley-interscience, 2006.

- [16] Pamela Begovic, Narcis Behlilovic, and Elma Avdic. Applicability evaluation of Okumura, Ericsson 9999 and winner propagation models for coverage planning in 3.5 GHZ WiMAX systems. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, pages 256–260. IEEE, 2012.
- [17] Lucas Benedičič, Felipe Cruz, Tsuyoshi Hamada, and Peter Korošec. A GRASS GIS parallel module for radio-propagation predictions. *International Journal of Geographical Information Science*, (under review), 2013.
- [18] Lucas Benedičič, Mitja Štular, and Peter Korošec. Pilot power optimization in UMTS: a multi-agent approach. In *Proceedings of the 13th International Multiconference Information Society - IS 2010*, volume A. Jožef Stefan Institute, October 2010.
- [19] Lucas Benedičič, Mitja Štular, and Peter Korošec. Balancing downlink and uplink soft-handover areas in UMTS networks. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE, 2012.
- [20] Lucas Benedičič, Mitja Štular, and Peter Korošec. A GPU-based parallel-agent optimization approach for the service-coverage problem in UMTS networks. *Computing and Informatics*, (In press), 2013.
- [21] Lucas Benedičič, Felipe A. Cruz, Tsuyoshi Hamada, and Peter Korošec. A GRASS GIS parallel module for radio-propagation predictions. *International Journal of Geographical Information Science*, (Submitted for publication), 2013.
- [22] Lucas Benedičič, Marko Pesko, Tomaž Javornik, and Peter Korošec. An adaptable parallel simulation framework for LTE coverage planning. *Simulation*, (under review), 2013.
- [23] Leonora Bianchi, Marco Dorigo, Luca Maria Gambardella, and Walter J Gutjahr. A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing: an international journal*, 8(2):239–287, 2009.
- [24] R. Blazek and L. Nardelli. The GRASS server. In *Proceedings of the Free/Libre and Open Source Software for Geoinformatics: GIS-GRASS Users Conference*, 2004.
- [25] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.
- [26] D.R. Butenhof. *Programming with POSIX threads*. Addison-Wesley Professional, 1997.
- [27] I. Campos, I. Coterillo, J. Marco, A. Monteoliva, and C. Oldani. Modelling of a Watershed: A Distributed Parallel Application in a Grid Framework. *Computing and Informatics*, 27(2):285–296, 2012.
- [28] L. Chen, K. Sandrasegaran, R. Basukala, F.M. Madani, and C.C. Lin. Impact of soft handover and pilot pollution on video telephony in a commercial network. In *Communications (APCC), 2010 16th Asia-Pacific Conference on*, pages 481–486. IEEE, 2010.
- [29] L. Chen and D. Yuan. Automated planning of CPICH power for enhancing HSDPA performance at cell edges with preserved control of R99 soft-handover. *Proceedings of IEEE ICC'08*, 2008.

- [30] L. Chen and D. Yuan. Automated planning of CPICH power for enhancing HSDPA performance at cell edges with preserved control of R99 soft handover. *Proceedings of IEEE ICC'08*, 2008.
- [31] L. Chen and D. Yuan. CPICH Power Planning for Optimizing HSDPA and R99 SHO Performance: Mathematical Modelling and Solution Approach. In *Proc. IFIP 1st Wireless Days Conference (WD)*, pages 1–5, 2008.
- [32] L. Chen and D. Yuan. Fast algorithm for large-scale UMTS coverage planning with soft-handover consideration. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 1488–1492. ACM, 2009.
- [33] L. Chen and D. Yuan. Coverage planning for optimizing HSDPA performance and controlling R99 soft handover. *Telecommunication Systems*, pages 1–12, 2011.
- [34] G. Cheung, W. Tan, and T. Yoshimura. Real-time video transport optimization using streaming agent over 3G wireless networks. *IEEE transactions on multimedia*, 7(4):777, 2005.
- [35] D.J. Cichon and T. Kurner. Propagation prediction models. *COST 231 Final Rep*, 1995.
- [36] Andrea Clematis, Mike Mineter, and Richard Marciano. Guest editorial: high performance computing with geographical data. *Parallel Computing*, 29(10):1275–1279, 2003.
- [37] C. Clemenccon, J. Fritscher, M. Meehan, and R. Rühl. An implementation of race detection and deterministic replay with MPI. *EURO-PAR'95 Parallel Processing*, pages 155–166, 1995.
- [38] Yoann Corre and Yves Lostanlen. Three-dimensional urban EM wave propagation model for radio network planning and optimization over large areas. *Vehicular Technology, IEEE Transactions on*, 58(7):3112–3123, 2009.
- [39] Michael Creutz. Microcanonical monte carlo simulation. *Physical Review Letters*, 50(19):1411–1414, 1983.
- [40] Felipe A Cruz, Simon K Layton, and Lorena A Barba. How to obtain efficient gpu kernels: An illustration using fmm & fgt algorithms. *Computer Physics Communications*, 182(10):2084–2098, 2011.
- [41] B.M. Cunningham, P.J. Alexander, and A. Candeub. Network growth: Theory and evidence from the mobile telephone industry. *Information Economics and Policy*, 22(1):91–102, 2010.
- [42] Marisa da Silva Maximiano, Miguel A Vega-Rodríguez, Juan A Gómez-Pulido, and Juan M Sánchez-Pérez. Analysis of parameter settings for differential evolution algorithm to solve a real-world frequency assignment problem in GSM networks. In *Advanced Engineering Computing and Applications in Sciences, 2008. ADVCOMP'08. The Second International Conference on*, pages 77–82. IEEE, 2008.
- [43] George Dantzig. Maximization of a linear function of variables subject to linear inequalities. *New York*, 1951.

- [44] S. Das and P.N. Suganthan. Differential evolution: A survey of the state-of-the-art. *Evolutionary Computation, IEEE Transactions on*, (99):1–28, 2010.
- [45] Leila De Floriani, Paola Magillo, and Enrico Puppo. Applications of computational geometry to geographic information systems. *Handbook of computational geometry*, pages 333–388, 1999.
- [46] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.
- [47] Manuel Duque-Antón, Dietmar Kunz, and Bernhard Ruber. Channel assignment for cellular radio using simulated annealing. *Vehicular Technology, IEEE Transactions on*, 42(1):14–21, 1993.
- [48] Harald Eckhardt, Siegfried Klein, and Markus Gruber. Vertical antenna tilt optimization for LTE base stations. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.
- [49] A. Eisenblätter, A. Füngenshuch, H. F. Geerdes, D. Junglas, T. Koch, and A. Martin. Optimization methods for UMTS radio network planning. In *Intl. Conference on Operations Research*, pages 31–38. Springer, September 2003.
- [50] A. Esposito, L. Tarricone, S. Luceri, and M. Zappatore. Genetic Optimization for Optimum 3G Network Planning: an Agent-Based Parallel Implementation. *Novel Algorithms and Techniques in Telecommunications and Networking*, pages 189–194, 2010.
- [51] Rob Farber. *CUDA application design and development*. Morgan Kaufmann Publishers Inc., 2011.
- [52] Sonja Filiposka and Dimitar Trajanov. Terrain-aware three-dimensional radio-propagation model extension for NS-2. *Simulation*, 87(1-2):7–23, 2011.
- [53] Michael C Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192–215, 2002.
- [54] M. Garcia-Lozano, S. Ruiz, and J. Olmos. CPICH power optimisation by means of simulated annealing in an UTRA-FDD environment. *Electronics Letters*, 39:1676, 2003.
- [55] M. Garcia-Lozano, S. Ruiz, and J. Olmos. CPICH power optimisation by means of simulated annealing in an UTRA-FDD environment. *Electronics Letters*, 39(23):1676–7, 2003.
- [56] Mario García-Lozano, María A Lema, Silvia Ruiz, and Flaminio Minerva. Metaheuristic Procedure to Optimize Transmission Delays in DVB-T Single Frequency Networks. *Broadcasting, IEEE Transactions on*, 57(4):876–887, 2011.
- [57] H.G. Gauch Jr. *Scientific method in practice*. Cambridge University Press, 2002.
- [58] A. Gerdenitsch. *System capacity optimization of UMTS FDD networks*. PhD thesis, Technische Universität Wien, June 2004.
- [59] A. Gerdenitsch, S. Jakl, M. Toeltsch, and T. Neubauer. Intelligent algorithms for system capacity optimization of umts fdd networks. *IEE Conference Publications*, 2003(CP494):222–226, 2003.

- [60] A. Gerdenitsch, Jakl S., and M. Toeltsch. The use of genetic algorithms for capacity optimization in UMTS FDD networks. *Proc. 3rd International Conference on Networking (ICN'04)*, March 2004.
- [61] Dirk Gerstenberger. Introduction to lte-advanced. *LTE-The UMTS Long Term Evolution: From Theory to Practice, Second Edition*, pages 613–622, 2011.
- [62] S.C. Ghosh, R.M. Whitaker, S.M. Allen, and S. Hurley. Optimising CDMA Cell Planning with Soft Handover. *Wireless Personal Communications*, pages 1–27, 2011.
- [63] Fred Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22(4):455–460, 1975.
- [64] Fred Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 13(5):533–549, 1986.
- [65] Zhaoya Gong, Wenwu Tang, David A Bennett, and Jean-Claude Thill. Parallel agent-based simulation of individual-level spatial interactions within a multicore computing environment. *International Journal of Geographical Information Science*, (ahead-of-print):1–19, 2012.
- [66] F. Gordejuela-Sánchez, D. López-Pérez, and J. Zhang. A two-step method for the optimization of antenna azimuth/tilt and frequency planning in OFDMA multi-hop networks. In *Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly*, pages 1404–1409. ACM, 2009.
- [67] Fernando Gordejuela-Sánchez, David López-Pérez, and Jie Zhang. A two-step method for the optimization of antenna azimuth/tilt and frequency planning in ofdma multi-hop networks. In *IWCNC '09: Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing*, pages 1404–1409, New York, NY, USA, 2009. ACM.
- [68] Fernando Gordejuela-Sánchez and Jie Zhang. Lte access network planning and optimization: a service-oriented and technology-specific perspective. In *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pages 1–5. IEEE, 2009.
- [69] Chris Gregg and Kim Hazelwood. Where is the data? Why you cannot debate CPU vs. GPU performance without the answer. In *Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on*, pages 134–144. IEEE, 2011.
- [70] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: portable parallel programming with the message passing interface*, volume 1. MIT press, 1999.
- [71] Qingfeng Guan, Phaedon C Kyriakidis, and Michael F Goodchild. A parallel computing approach to fast geostatistical areal interpolation. *International Journal of Geographical Information Science*, 25(8):1241–1267, 2011.
- [72] T. Hamada and K. Nitadori. 190 TFlops astrophysical N-body simulation on a cluster of GPUs. In *Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–9. IEEE Computer Society, 2010.

- [73] Tao Han and Nirwan Ansari. Optimizing cell size for energy saving in cellular networks with hybrid energy supplies. In *Global Telecommunications Conference (GLOBECOM 2012), 2012 IEEE*, 2012.
- [74] Q. Hao, B.H. Soong, E. Gunawan, J.T. Ong, C.B. Soh, and Z. Li. A low-cost cellular mobile communication system: a hierarchical optimization network resource planning approach. *IEEE Journal on Selected Areas in Communications*, 15(7):1315–1326, 1997.
- [75] Masaharu Hata. Empirical formula for propagation loss in land mobile radio services. *Vehicular Technology, IEEE Transactions on*, 29(3):317–325, 1980.
- [76] Masaharu Hata. Empirical formula for propagation loss in land mobile radio services. *Vehicular Technology, IEEE Transactions on*, 29(3):317–325, 1980.
- [77] Kenneth A Hawick, Paul David Coddington, and HA James. Distributed frameworks and parallel algorithms for processing large-scale geographic data. *Parallel Computing*, 29(10):1297–1333, 2003.
- [78] H. Holma and A. Toskala. *WCDMA for UMTS: Radio access for third generation mobile communications, Third Edition*. John Wiley & Sons, Inc. New York, NY, USA, 2005.
- [79] H. Holma and A. Toskala. *WCDMA for UMTS: Radio access for third generation mobile communications, Third Edition*. John Wiley & Sons, Inc. New York, NY, USA, 2005.
- [80] H. Holma and A. Toskala. *HSDPA/HSUPA For UMTS*. John Wiley & Sons, 2006.
- [81] A. Hrovat, I. Ozimek, A. Vilhar, T. Celcer, I. Saje, and T. Javornik. Radio coverage calculations of terrestrial wireless networks using an open-source GRASS system. *WSEAS Transactions on Communications*, 9(10):646–657, 2010.
- [82] F. Huang, D. Liu, X. Tan, J. Wang, Y. Chen, and B. He. Explorations of the implementation of a parallel IDW interpolation algorithm in a Linux cluster-based parallel GIS. *Computers & Geosciences*, 37(4):426–434, 2011.
- [83] Lianfen Huang, Xiaoxin Chen, Zhibin Gao, and Hongxiang Cai. Online propagation model correction based on PSO algorithm in LTE SON system. In *Anti-Counterfeiting, Security and Identification (ASID), 2012 International Conference on*, pages 1–4. IEEE, 2012.
- [84] Qunying Huang, Chaowei Yang, Karl Benedict, Abdelmounaam Rezgui, Jibo Xie, Jizhe Xia, and Songqing Chen. Using adaptively coupled models and high-performance computing for enabling the computability of dust storm forecasting. *International Journal of Geographical Information Science*, (ahead-of-print):1–20, 2012.
- [85] IST-MOMENTUM. Models and Simulations for Network planning and Control of UMTS. <http://momentum.zib.de>, [Online; accessed 15-April-2010].
- [86] Laxmikant V. Kale and Abhinav Bhatele, editors. *Parallel Science and Engineering Applications: The Charm++ Approach*. Taylor & Francis Group, CRC Press, November 2013.
- [87] H. Kargupta and D. E. Goldberg. Search, blackbox optimization, and sample complexity. In *4th Workshop on Foundations of Genetic Algorithms (FOGA)*, pages 291–324. Morgan Kaufmann, 1996.

- [88] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 302–311. ACM, 1984.
- [89] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671, 1983.
- [90] Andreas Kloeckner. GPU from Python with PyOpenCL and PyCUDA. <http://www.bu.edu/pasi/materials/>, [Online; accessed 5-April-2010].
- [91] Gary A Kochenberger et al. *Handbook of metaheuristics*. Springer, 2003.
- [92] Peter Korošec, Jurij Šilc, and Bogdan Filipič. The differential ant-stigmergy algorithm. *Information Sciences*, 192:82–97, 2012.
- [93] Peter Korošec, Jurij Šilc, and Bogdan Filipič. The differential ant-stigmergy algorithm. *Information Sciences*, 192(1):82–97, 2012. doi: 10.1016/j.ins.2010.05.002.
- [94] G Labeledz, K Felix, V Lev, and D Schaeffer. Handover control issues in very high capacity cellular systems using small cells. In *International Conference on Digital Land Mobile Radio Communications*, 1987.
- [95] J. Laiho, A. Wacker, and T. Novosad. *Radio Network Planning and Optimisation for UMTS*. John Wiley & Sons, Inc. New York, NY, USA, 2002.
- [96] AM Law. *Simulation modeling and analysis*. Boston, MA [etc.]: McGraw-Hill, 2007.
- [97] Suk-Bok Lee, Ioannis Pefkianakis, Adam Meyerson, Shugong Xu, and Songwu Lu. Proportional fair frequency-domain packet scheduling for 3GPP LTE uplink. In *INFOCOM 2009, IEEE*, pages 2611–2615. IEEE, 2009.
- [98] J. Lempiäinen and M. Manninen. *UMTS Radio Network Planning, Optimization and QoS Management for Practical Engineering Tasks*. Kluwer Academic Publishers Norwell, MA, USA, 2004.
- [99] Xia Li, Xiaohu Zhang, Anthony Yeh, and Xiaoping Liu. Parallel cellular automata for large-scale urban simulation using load-balancing techniques. *International Journal of Geographical Information Science*, 24(6):803–820, 2010.
- [100] Kai Lieska, Erkki Laitinen, and Jaakko Lahteenmaki. Radio coverage optimization with genetic algorithms. In *Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on*, volume 1, pages 318–322. IEEE, 1998.
- [101] Sean Luke. *Essentials of Metaheuristics*. Lulu, 2009. Available for free at <http://cs.gmu.edu/~sean/book/metaheuristics/>.
- [102] Samip Malla, Birendra Ghimire, Mark C Reed, and Harald Haas. Energy efficient resource allocation in OFDMA networks using ant-colony optimization. In *Communications and Information Technologies (ISCIT), 2012 International Symposium on*, pages 889–894. IEEE, 2012.
- [103] A. Maria. Introduction to modeling and simulation. In *Proceedings of the 29th conference on Winter simulation*, pages 7–13. IEEE Computer Society, 1997.
- [104] G. Marsaglia. Seeds for random number generators. *Communications of the ACM*, 46(5):90–93, 2003.

- [105] R. Mathar and T. Niessen. Optimum positioning of base stations for cellular radio networks. *Wireless Networks*, 6(6):421–428, 2000.
- [106] Christian Mehlührer, Josep Colom Colom Ikuno, Michal Šimko, Stefan Schwarz, Martin Wrulich, and Markus Rupp. The Vienna LTE simulators-Enabling reproducibility in wireless communications research. *EURASIP Journal on Advances in Signal Processing*, 2011(1):1–14, 2011.
- [107] Silvio Priem Mendes, JA Gómez Pulido, JM Sanchez Perez, et al. A differential evolution based algorithm to optimize the radio network design problem. In *e-Science and Grid Computing, 2006. e-Science'06. Second IEEE International Conference on*, pages 119–119. IEEE, 2006.
- [108] Gordon E Moore et al. Cramming more components onto integrated circuits. *Proceedings of the IEEE*, 86(1):82–85, 1998.
- [109] A. Munshi et al. The OpenCL specification version 1.0. *Khronos OpenCL Working Group*, 2009.
- [110] M. Nawrocki, H. Aghvami, and M. Dohler. *Understanding UMTS radio network modelling, planning and automated optimisation: theory and practice*. John Wiley & Sons, 2006.
- [111] M. Nawrocki, H. Aghvami, and M. Dohler. *Understanding UMTS radio network modelling, planning and automated optimisation: theory and practice*. John Wiley & Sons, 2006.
- [112] M. Nawrocki, H. Aghvami, and M. Dohler. *Understanding UMTS radio network modelling, planning and automated optimisation: theory and practice*. John Wiley & Sons, 2006.
- [113] A. Neskovic and N. Neskovic. Microcell electric field strength prediction model based upon artificial neural networks. *AEU-International Journal of Electronics and Communications*, 64(8):733–738, 2010.
- [114] Markus Neteler and Helena Mitasova. *Open Source software and GIS*. Springer, 2008.
- [115] Michaela Neuland, T Kurner, and Mehdi Amirijoo. Influence of Different Factors on X-Map Estimation in LTE. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.
- [116] Michaela Neuland, T Kurner, and Mehdi Amirijoo. Influence of positioning error on X-map estimation in LTE. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1–5. IEEE, 2011.
- [117] NVIDIA. CUDA. <https://developer.nvidia.com/category/zone/cuda-zone>, [Online; accessed 24-August-2013].
- [118] Ofcom. Table of base station totals. Available from: <http://stakeholders.ofcom.org.uk/sitefinder/table-of-totals/>, 2012.
- [119] A. Osterman. Implementation of the r.cuda.los module in the open source GRASS GIS by using parallel computation on the NVIDIA CUDA graphic cards. *Elektrotehniški Vestnik*, 79(1-2):19–24, 2012.

- [120] John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. GPU computing. *Proceedings of the IEEE*, 96(5):879–899, 2008.
- [121] M Tamer Özsu and Patrick Valduriez. *Principles of distributed database systems*. Springer, 2011.
- [122] John David Parsons and Prof J David Parsons. *The mobile radio propagation channel*, volume 2. Wiley New York, 2000.
- [123] Andreas Pillekeit and Bruno Müller-Clostermann. A hybrid simulation framework for the evaluation of common radio resource management scenarios. *Simulation*, 88(4):481–501, 2012.
- [124] Giuseppe Piro, Luigi Alfredo Grieco, Gennaro Boggia, Francesco Capozzi, and Pietro Camarda. Simulating LTE cellular systems: an open-source framework. *Vehicular Technology, IEEE Transactions on*, 60(2):498–513, 2011.
- [125] Kenneth V Price, Rainer M Storn, and Jouni A Lampinen. Differential evolution a practical approach to global optimization. 2005.
- [126] K.V. Price, R.M. Storn, and J.A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer-Verlag New York Inc., 2005.
- [127] Rolf Rabenseifner, Georg Hager, and Gabriele Jost. Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes. In *Parallel, Distributed and Network-based Processing, 2009 17th Euromicro International Conference on*, pages 427–436. IEEE, 2009.
- [128] L. Raisanen and R. Whitaker. Comparison and evaluation of multiple-objective genetic algorithms for the antenna placement problem. *Mobile Networks and Applications*, 10:79–88, February 2005.
- [129] Sara Modarres Razavi and Di Yuan. Performance improvement of lte tracking area design: a re-optimization approach. In *Proceedings of the 6th ACM international symposium on Mobility management and wireless access*, pages 77–84. ACM, 2008.
- [130] Jani Ronkkonen, Saku Kukkonen, and Kenneth V Price. Real-parameter optimization with differential evolution. In *Evolutionary Computation, 2005. The 2005 IEEE Congress on*, volume 1, pages 506–513. IEEE, 2005.
- [131] Shane Ryoo, Christopher I Rodrigues, Sara S Baghsorkhi, Sam S Stone, David B Kirk, and Wen-mei W Hwu. Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*, pages 73–82. ACM, 2008.
- [132] A.B. Saleh, S. Redana, J. Hämäläinen, and B. Raaf. On the coverage extension and capacity enhancement of inband relay deployments in LTE-Advanced networks. *Journal of Electrical and Computer Engineering*, 2010:4, 2010.
- [133] Oriol Sallent, Jordi Pérez-Romero, Juan Sánchez-González, Ramon Agustí, Miguel Ángel Díaz-Guerra, David Henche, and Daniel Paul. A roadmap from UMTS optimization to LTE self-optimization. *Communications Magazine, IEEE*, 49(6):172–182, 2011.

- [134] Juan J Sánchez, Gerardo Gómez, David Morales-Jiménez, and J Tomás Entrambasaguas. Performance evaluation of OFDMA wireless systems using WM-SIM platform. In *Proceedings of the 4th ACM international workshop on Mobility management and wireless access*, pages 131–134. ACM, 2006.
- [135] P. Sarkar. A brief history of cellular automata. *ACM Computing Surveys (CSUR)*, 32(1):107, 2000.
- [136] Dana Schaa and David Kaeli. Exploring the multiple-GPU design space. In *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–12. IEEE, 2009.
- [137] N. Shabbir, M.T. Sadiq, H. Kashif, and R. Ullah. Comparison of Radio Propagation Models for Long Term Evolution (LTE) Network. *arXiv preprint arXiv:1110.1519*, 2011.
- [138] S. Shepler, M. Eisler, D. Robinson, B. Callaghan, R. Thurlow, D. Noveck, and C. Beame. Network file system (NFS) version 4 protocol. *Network*, 2003.
- [139] S. Siegel and G. Avrunin. Verification of halting properties for MPI programs using nonblocking operations. *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pages 326–334, 2007.
- [140] I. Siomina. Pilot power management in WCDMA networks: coverage control with respect to traffic distribution. In *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 276–282. ACM New York, NY, USA, 2004.
- [141] I. Siomina, P. Varbrand, and Di Yuan. Automated optimization of service coverage and base station antenna configuration in UMTS networks. *IEEE Wireless Communications*, 13(6):16–25, 2006.
- [142] I. Siomina and D. Yuan. Pilot power optimization in WCDMA networks. 4, March 2004.
- [143] I. Siomina and D. Yuan. Enhancing HSDPA performance via automated and large-scale optimization of radio base station antenna configuration. In *Proc. 67th IEEE Vehicular Technology Conference (VTC2008-Spring)*, pages 2061–2065, 2008.
- [144] I. Siomina and D. Yuan. Enhancing HSDPA performance via automated and large-scale optimization of radio base station antenna configuration. In *Proc. 67th IEEE Vehicular Technology Conference (VTC2008-Spring)*, pages 2061–2065, 2008.
- [145] I. Siomina and D. Yuan. Minimum pilot power for service coverage in WCDMA networks. *Wireless Networks*, 14(3):393–402, 2008.
- [146] Iana Siomina and Di Yuan. Minimum pilot power for service coverage in WCDMA networks. *Wireless Networks*, 14(3):393–402, June 2007.
- [147] Iana Siomina and Di Yuan. Minimum pilot power for service coverage in WCDMA networks. *Wireless Networks*, 14(3):393–402, June 2007.
- [148] Iana Siomina and Di Yuan. Minimum pilot power for service coverage in WCDMA networks. *Wireless Networks*, 14(3):393–402, June 2007.

- [149] Lingyang Song and Jia Shen. *Evolved cellular network planning and optimization for UMTS and LTE*. CRC Press, 2010.
- [150] R. Stallman et al. GNU general public license. *Free Software Foundation, Inc., Tech. Rep*, 1991.
- [151] J.E. Stone, D. Gohara, and G. Shi. OpenCL: A parallel programming standard for heterogeneous computing systems. *Computing in Science and Engineering*, 12:66–73, 2010.
- [152] Michael Stonebraker. SQL databases v. NoSQL databases. *Communications of the ACM*, 53(4):10–11, 2010.
- [153] R. Storn and K. Price. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.
- [154] Gordon L Stüber. *Principles of mobile communication*. Springer, 2011.
- [155] B. Suman and P. Kumar. A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the operational research society*, 57(10):1143–1160, 2005.
- [156] Siham Tabik, Luis Felipe Romero, and Emilio López Zapata. High-performance three-horizon composition algorithm for large-scale terrains. *International Journal of Geographical Information Science*, 25(4):541–555, 2011.
- [157] Siham Tabik, A Villegas, and Emilio López Zapata. Optimal tilt and orientation maps: a multi-algorithm approach for heterogeneous multicore-GPU systems. *The Journal of Supercomputing*, pages 1–13, 2013.
- [158] Siham Tabik, Emilio López Zapata, and Luis Felipe Romero. Simultaneous computation of total viewshed on large high resolution grids. *International Journal of Geographical Information Science*, 2012.
- [159] E.G. Talbi. *Metaheuristics: from design to implementation*. Wiley, 2009.
- [160] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. Wiley, 2009.
- [161] A-A Tantar, Nouredine Melab, E-G Talbi, B Parent, and D Horvath. A parallel hybrid genetic algorithm for protein structure prediction on the computational grid. *Future Generation Computer Systems*, 23(3):398–409, 2007.
- [162] Ajay Thampi, Dritan Kaleshi, Peter Randall, Walter Featherstone, and Simon Armour. A sparse sampling algorithm for self-optimisation of coverage in lte networks. In *Wireless Communication Systems (ISWCS), 2012 International Symposium on*, pages 909–913. IEEE, 2012.
- [163] Third Generation Partnership Project. *Technical specification group services and systems aspects: Network architecture v3.6.0 (Release 1999)*. <http://www.3gpp.org/>.
- [164] U. Türke and M. Koonert. Advanced site configuration techniques for automatic UMTS radio network design. *Proc IEEE VTC 2005 Spring*, 2005.

- [165] Ulrich Türke. *Efficient methods for WCDMA radio network planning and optimization*. Springer, 2007.
- [166] A. Valcarce, G. De La Roche, Á. Jüttner, D. López-Pérez, and J. Zhang. Applying FDTD to the coverage prediction of WiMAX femtocells. *EURASIP Journal on Wireless Communications and Networking*, 2009:1–13, 2009.
- [167] K. Valkealahti, A. Hoglund, J. Parkkinen, and A. Flanagan. WCDMA Common Pilot Power Control with Cost Function Minimization. *statistics*, 1000(1):1, 2002.
- [168] P. Värbrand and D. Yuan. A mathematical programming approach for pilot power optimization in WCDMA networks. 2003.
- [169] M. Vasile and M. Locatelli. A hybrid multiagent approach for global trajectory optimization. *Journal of Global Optimization*, 44(4):461–479, 2009.
- [170] Andrej Vilhar, Andrej Hrovat, Igor Ozimek, and Tomaž Javornik. Efficient open-source ray-tracing methods for rural environment. In *Proceedings of the 16th WSEAS International Conference on Computers*, pages 15–17, 2012.
- [171] Mladen A Vouk. Cloud computing—issues, research and implementations. *Journal of Computing and Information Technology*, 16(4):235–246, 2008.
- [172] Achim Wacker, Jaana Laiho-Steffens, Kari Sipila, and Mika Jasberg. Static simulator for studying wcdma radio network planning issues. In *Vehicular Technology Conference, 1999 IEEE 49th*, volume 3, pages 2436–2440. IEEE, 1999.
- [173] Shaowen Wang. A cyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers*, 100(3):535–557, 2010.
- [174] Michael J Widener, Neal C Crago, and Jared Aldstadt. Developing a parallel computational implementation of amoeba. *International Journal of Geographical Information Science*, 26(9):1707–1723, 2012.
- [175] Daniel Wong and Teng Joon Lim. Soft handoffs in CDMA mobile systems. *Personal Communications, IEEE*, 4(6):6–17, 1997.
- [176] Howard Xia, Henry L Bertoni, Leandro R Maciel, Andrew Lindsay-Stewart, and Robert Rowe. Radio propagation characteristics for line-of-sight microcellular and personal communications. *Antennas and Propagation, IEEE Transactions on*, 41(10):1439–1447, 1993.
- [177] Jun Yang, Mehmet Emin Aydin, Jie Zhang, and Carsten Maple. UMTS base station location planning: a mathematical model and heuristic optimisation algorithms. *Communications, IET*, 1(5):1007–1014, 2007.
- [178] Mingjing Yang and Wenxiao Shi. A linear least square method of propagation model tuning for 3G radio network planning. In *Natural Computation, 2008. ICNC’08. Fourth International Conference on*, volume 5, pages 150–154. IEEE, 2008.
- [179] Gavin Yeung, Mineo Takai, Rajive Bagrodia, Alireza Mehrnia, and Babak Daneshrad. Detailed ofdm modeling in network simulation of mobile ad hoc networks. In *Proceedings of the eighteenth workshop on Parallel and distributed simulation*, pages 26–34. ACM, 2004.

- [180] Ling Yin, Shih-Lung Shaw, Dali Wang, Eric A Carr, Michael W Berry, Louis J Gross, and E Jane Comiskey. A framework of integrating GIS and parallel computing for spatial control problems—a case study of wildfire control. *International Journal of Geographical Information Science*, 26(4):621–641, 2012.
- [181] S. Ying, F. Gunnarsson, K. Hiltunen, E. Res, and C. Beijing. CPICH power settings in irregular WCDMA macro cellular networks. *14th IEEE Proceedings on Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003*, 2, 2003.
- [182] Jie Zhang, Jun Yang, Mehmet E Aydin, and Joyce Y Wu. Mathematical modelling and comparisons of four heuristic optimization algorithms for WCDMA radio network planning. In *Transparent Optical Networks, 2006 International Conference on*, volume 3, pages 253–257. IEEE, 2006.



## 13 Acknowledgments

The authors would like to especially thank the radio engineers at Telekom Slovenije, d.d., for cooperating and sharing their professional expertise throughout the creation of this work. This project was co-financed by the European Union, through the European Social Fund.

The contents presented in Chapter 6 are the results of joint research work, conducted with the Nagasaki Advanced Computer Center (NACC) of the Nagasaki University. In this context, T. Hamada, head of NACC, acknowledges support from the Japan Society for the Promotion of Science (JSPS) through its Funding Program for World-leading Innovative R&D on Science and Technology (First Program).

The authors would like to especially thank the staff at the Nagasaki Advanced Computing Center (NACC) of the Nagasaki University for their support while making the computer cluster DEGIMA available for the experimental simulations. We are also grateful with the radio engineers at Telekom Slovenije, d.d., for supplying the network data sets and sharing their professional expertise throughout the creation of this work.

This project was co-financed by the European Union, through the European Social Fund.