

Figure 7.3: Convergence profile of the parallel-agent approach for the test network Net_2 .

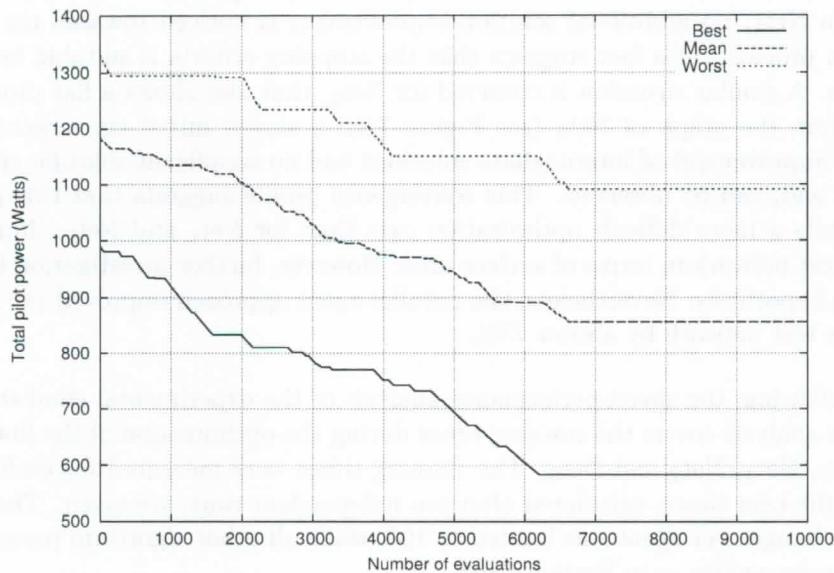


Figure 7.4: Convergence profile of the parallel-agent approach for the test network Net_3 .

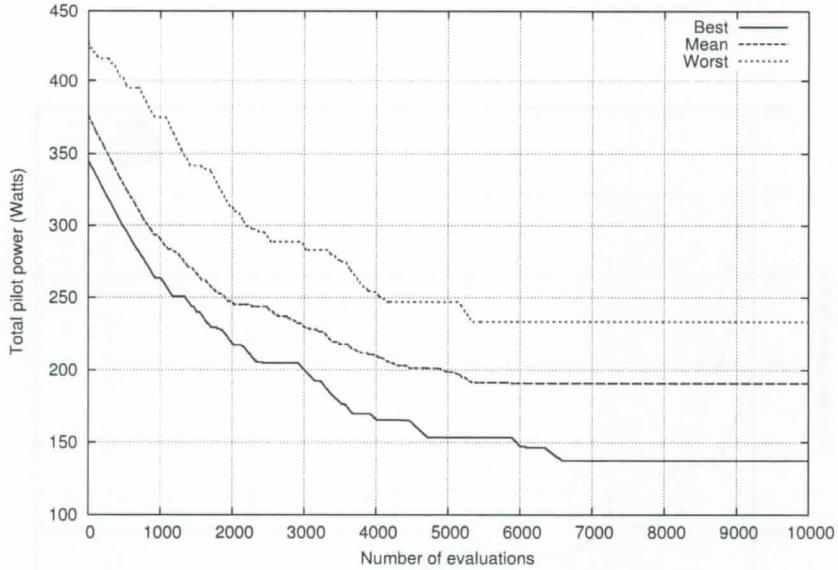


Figure 7.2: Convergence profile of the parallel-agent approach for the test network Net_1 .

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 Net_1 , Net_2 and 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 Net_1 , 129 for Net_2 , and 1,435 for Net_3 .

From the graphs of Net_1 (see Figure 7.2) and 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 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 Net_2 , that also shows a flat profile towards the end. From the graph of 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 Net_1 and 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., Net_1 , Net_2 and 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

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.

<i><These times are not final??></i>					
	CPU-MPI	GPU-MPI		GPU-GPU	
	Best time	Best time	Speed-up	Best time	Speed-up
<i>Net₁</i>	105,455	346	305x	67	1574x
<i>Net₂</i>	33,700	195	173x	46	733x
<i>Net₃</i>	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 ???%.

ZAKA > VA TUKAS "?!?" ?

SEA KAS NAWSKA?

ZAKA > WE AVERAGE INTS.

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 → **MULTI-HOAOAL** (?) 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. **APPROACHES.**

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.

↑
AND RESULT ANALYSIS

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].

VESTIGI R&D

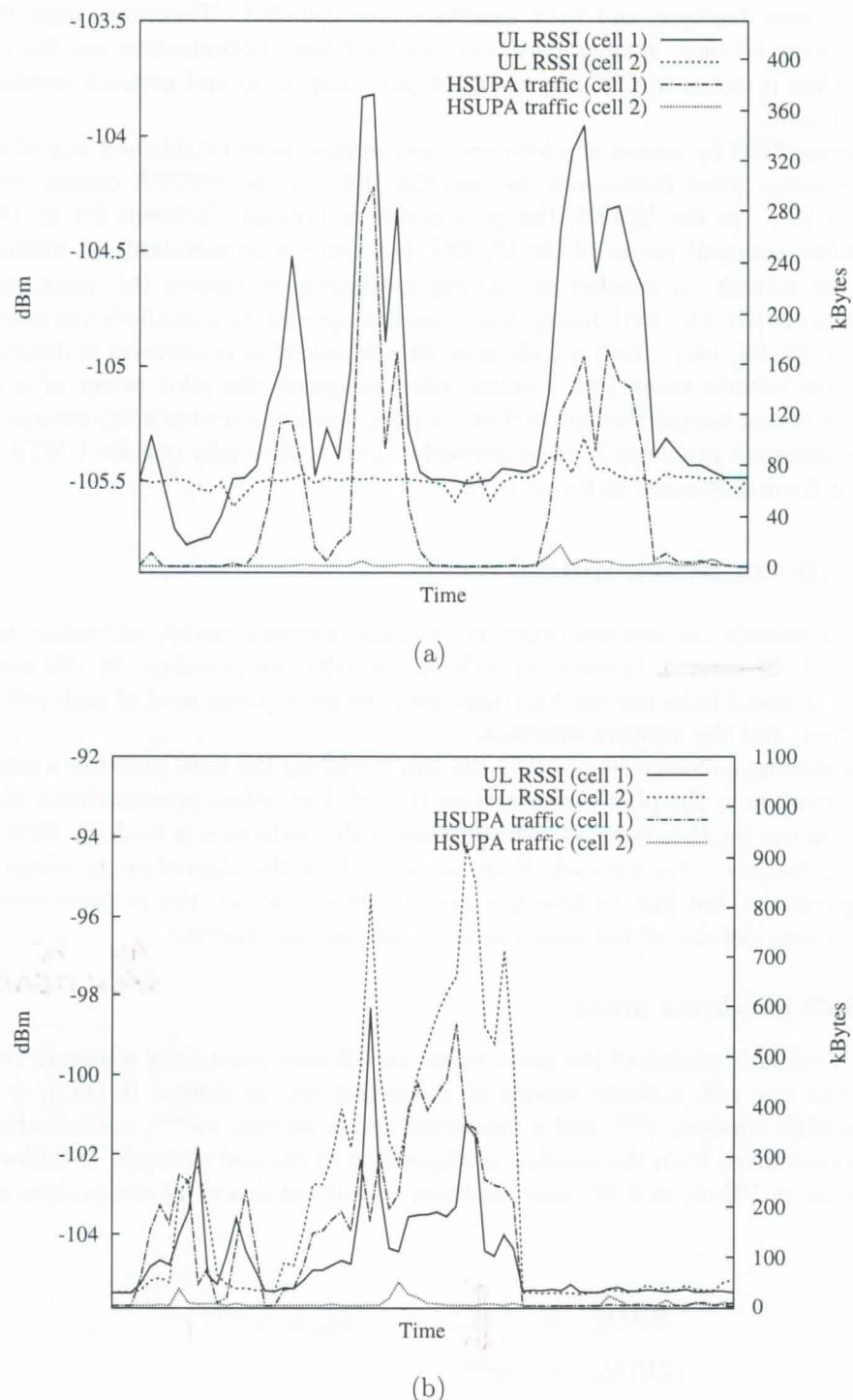


Figure 8.1: HSUPA traffic and uplink interference with: (a) balanced downlink and uplink SHO conditions, and (b) unbalanced downlink and uplink SHO conditions.

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 ± 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.

*ALL SE CAKKO REALNO
STRENNENBE < 0.01 dB ??!*

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, γ^{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:

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

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 γ^{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:

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

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

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

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

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

It is clear that the coverage is the most important quality aspect from the network point of view (penalty factor pf_{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_{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_{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.:

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

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.

1 $c' \leftarrow \text{pick_random_cell}(C)$
 repeat
 2 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.

ČE horovis
 o "steps"
 horaso biti
 ornačeni
 v koni

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, Net₇, 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 km². Table 8.1 shows some characteristics of the test network used, and Figure 8.2 shows the area under radio coverage, A_{covered} , within A_{total} .

Table 8.1: Technical characteristics of Net₇, the test network used for the SHO-balancing problem.

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

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:

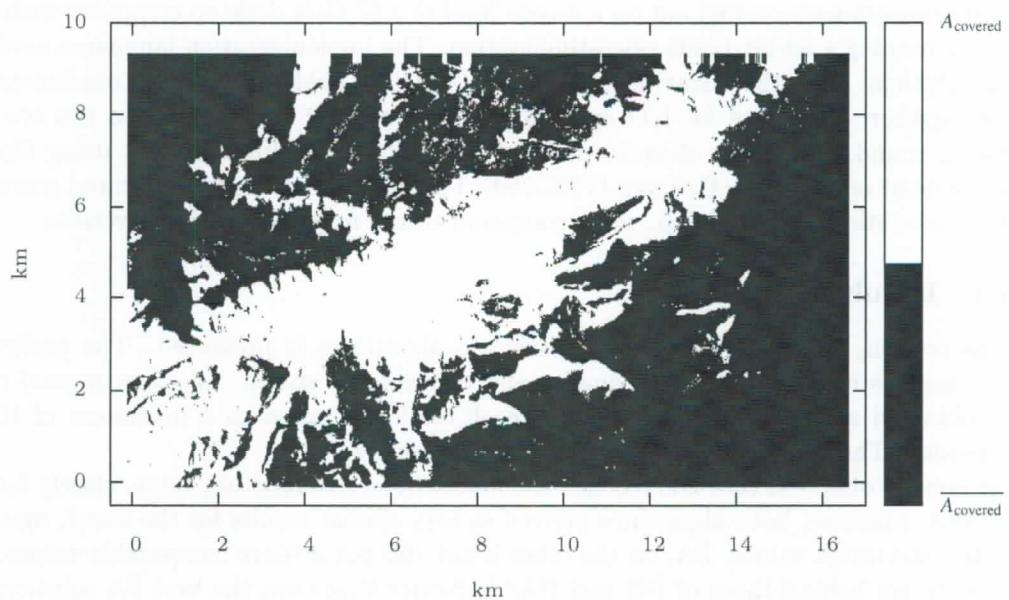


Figure 8.2: Area under radio coverage, A_{covered} , and without radio coverage, $\overline{A}_{\text{covered}}$, within the complete geographical area, A_{total} , of test network Net₇.

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

DU RA' NOGECI RUSTI

- $t_{initial} = 125$, 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.

TO NI OK. ČI ČE DELAS PRINTER JAVO
BĘTĘ ČI SO IN PLĘNĘ WŁAWI V KAZZI 8214 SEZIRE

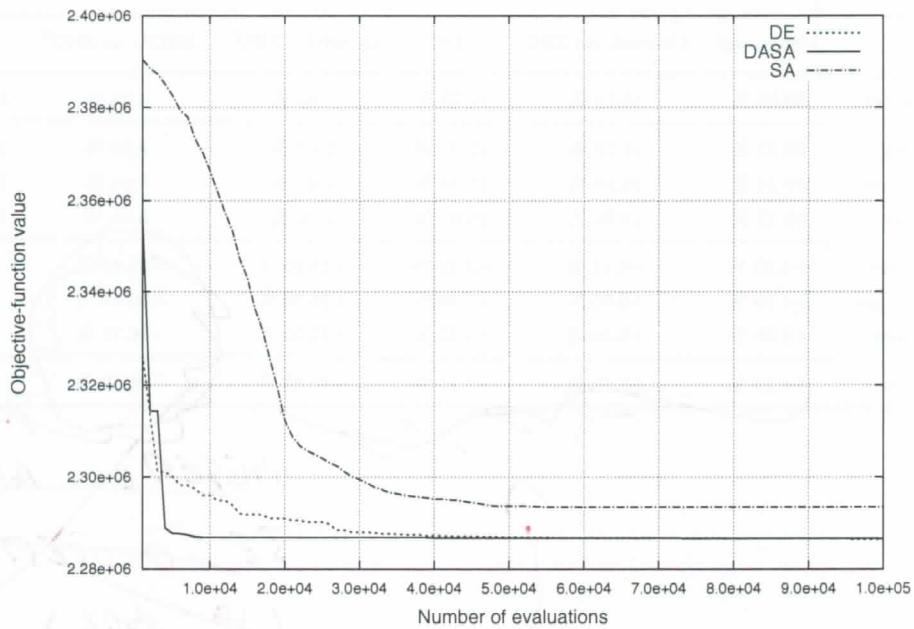


Figure 8.3: Convergence analysis for each of the three algorithms, i.e., DE, DASA and SA, showing the best results obtained for the SHO-balancing problem.

ERKEN) WE AVG.!

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[↓], no SHO[↑]'), which shows an expected decline, since it is the optimization aspect with the lowest penalty-factor value.

AVG.

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 [†] , SHO [†]	SHO [†] , no SHO [†]	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 %	—

Handwritten notes in red ink:

- Based on all our
- SE or TEP
- LIE BOC RONIN BEN
- + ALI TA
- SEN VLDL LASWEZ?
- 1. part (part)

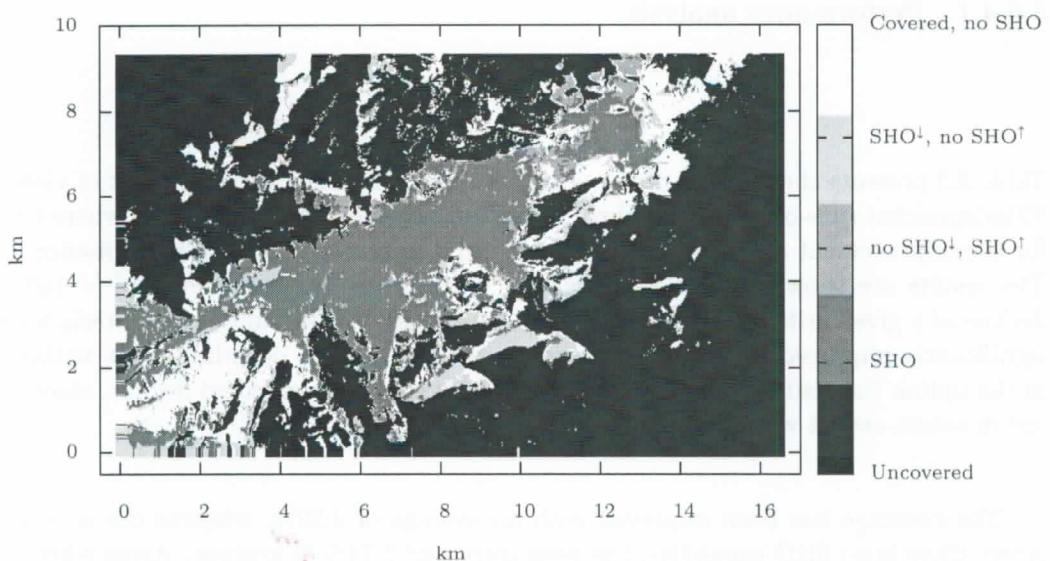


Figure 8.4: Spatial distribution of the SHO areas before the optimization.

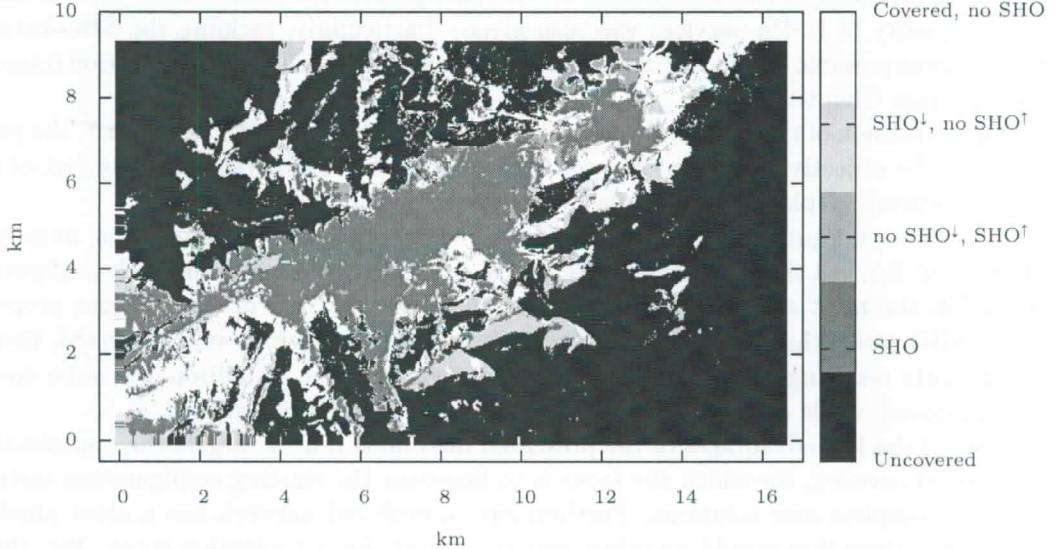


Figure 8.5: Spatial distribution of the SHO areas after the optimization.

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 ‘ $\text{SHO}^\downarrow, \text{no SHO}^\uparrow$ ’ 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 ± 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.

HANJKA POUVANÉK NOVOSTI

KOT PRI 6 IN 7: