Examensarbete

Handover Optimization in GSM

Johann Joachim Pavski

LiTH - MAT - EX - - 2015/10 - - SE
Handover Optimization in GSM

Department of Mathematics, Division of Optimization, Linköping University

Johann Joachim Pavski

LiTH - MAT - EX - - 2015/10 - - SE

Examensarbete: 30 hp

Level: A

Supervisor: Nils-Hassan Quttineh,
Department of Mathematics, Division of Optimization, Linköping University

Sebastian Lindqvist,
Ericsson, GSM Department

Examiner: Elina Rönnberg,
Department of Mathematics, Division of Optimization, Linköping University

Linköping: December 2015
Abstract

In telecommunications in general and in GSM in particular, the handover is a feature that guarantees a smooth transition of a call from one base station - that is for the purpose of this project an antenna - to another. In the recent ten years, the amount of data traffic through mobile telecommunications has doubled annually, putting an enormous strain on the network and forcing operators to upgrade with more and more base stations and new features. Although 3G and 4G are responsible for data traffic in most countries, GSM still provides more than 80% of the coverage for mobile devices around the world. Due to the increase in data traffic, 3G and 4G need to use more and more frequencies at the expense of GSM. An optimization of the GSM network is thus vital.

In this project, we research two methods to automatically choose the parameters of interest (PoI) that govern the handover feature in each cell which is, roughly speaking, the area of coverage of one antenna. In one of these methods, the choice of cell- and cell-to-cell-specific parameters has its origins in control theory while the other method is based on mathematical optimization.

In the mathematical sense, our goal is to optimize the quality of service over PoIs. Extensive simulations have been run using these PoIs in order to evaluate if and how the two different methods can effectively be used in reality. Several useful insights have been gained that will provide the basis for future work. The optimization approach in particular has proved to deliver good results within the limitations of the simulated environment used for testing.

Keywords: Optimization, heuristic, fuzzy control, costly global optimization, surrogate surface, GSM, handover.

URL for electronic version:

Acknowledgements

Thanks go to:
My supervisor Sebastian at Ericsson for advice and giving me the chance to work on this project. My supervisor Nils at the mathematical institution for patience, advice and literature. My examiner Elina. My colleague and co thesis worker Ahmad for being knowledgeable, fun and professional.
I want to especially recognize the whole GSM department at Ericsson for their kindness and availability: thanks to Birgitta, Ulf, Patrick, Ingmar, Jaqueline and Magnus.
Catharina for making me want to work and for making me happy, Miji for making me want to live, Hanna for caffeine pills and motivating me to keep studying, Julius for giving me discipline, Johan for teaching me Swedish and lifting, Melisa for being a true friend in difficult times, Benny for making my studies very pleasant, the Swedish education system for being free and more than fair, MAI for teaching me to find structure in what I believed to be chaos. Finally I would like to add a shout out to my mother for her guidance and patience.
Nomenclature

Most of the reoccurring abbreviations and symbols are described here.

Symbols

- $T = \{1, 2, 3, 4\}$, the set of types of design variables.
- $N_t$, the set of all design variables of type $t$
- $x_{jt}$, the $j$th design variable of type $t$
- $\mathcal{X}$, the space of design variables $x_{jt}$
- $y_i$, objective value of type $i$
- $C(x)$, a weighted sum of objective values
- $S(x)$, a surrogate surface, interpolating $C(x)$
- $\tilde{x}_t$, the design variable of type $t$. $\tilde{x}_t = x_{jt} \forall j \in N_t, t \in T$
- $\tilde{\mathcal{X}}$, the space of design variables $\tilde{x}_t$
- $\tilde{y}_i$, objective value of type $i$
- $C(\tilde{x})$, a weighted sum of objective values
- $S(\tilde{x})$, a surrogate surface, interpolating $C(\tilde{x})$
- $c$, the number of cells
- $n_c$, the number of cell to cell relations
Abbreviations
The field of telecommunications is a breeding ground for abbreviations. Their amount was limited as much as possible within this thesis, but the most important ones are listed here.

- E3 - Within this thesis, the name of the algorithm that decides when a handover should be performed.

- KPI - Key Performance Indicator. A counter in the system that tracks how parts of the network behave. It simply counts events. There are about 2500 of these in the base station controller. KPIs are what allows to quantify Quality of Service. In this project, we use a sum of a set of KPIs in order to define a cost that we want to minimize.

- AKPI - A KPI averaged over either a cell or a time period. The AKPIs are the objective values that are to be optimized.

- PoI - Parameter of Interest. Parameters that configure the E3. These are the design variables of the optimization problem.

- HDS - Higher Dimensional Search. Designates the control algorithm.

- CGO - Costly Global Optimization. The algorithm that is based on the principles of costly global optimization is called the CGO-approach.

- UE - User Equipment. A cell phone, a pad or any unit that transfers data via the GSM network.

- QoS - Quality of Service. An umbrella term for the network performance as viewed by the user.

- BSC - Base Station Controller, a central operating unit coordinating up to 2000 cells.
Contents

1 Introduction and Problem Definition 1
   1.1 Background .............................................. 1
   1.2 Formalities .............................................. 2
   1.3 Scope of the Thesis ..................................... 3
   1.4 Topics Covered .......................................... 4

2 Telecommunications Background 5
   2.1 The Telecommunications Problem ...................... 5
      2.1.1 Network Cell Plan Schematic .................... 5
      2.1.2 Schematic of the Handover Algorithm .......... 9
   2.2 Definition of the design variables .................... 10
   2.3 Definition of the Cost Function ...................... 11

3 Theory 13
   3.1 Introduction ............................................ 13
   3.2 Costly Global Optimization ........................... 13
   3.3 Self Optimizing Handover features ................... 17

4 Mathematical Problem Formulation 21
   4.1 Introduction ............................................ 21
   4.2 Costly Global Optimization ........................... 22
      4.2.1 Adaptation of the Algorithm .................... 23
   4.3 Higher Dimensional Search ............................ 25
      4.3.1 Structure of the Algorithm .................... 25
      4.3.2 Fuzzy Logic ....................................... 26

5 Results 29
   5.1 Costly Global Optimization Approach ................ 29
   5.2 Higher Dimensional Search Approach ................ 32

6 Discussion and Future Work 39
   6.1 Discussion of the Costly Global Optimization approach .... 39
   6.2 Discussion of the Higher Dimensional Search approach ... 40
   6.3 Future Work ............................................. 40

7 Conclusion 43

Pavski, 2015. xi
Chapter 1

Introduction and Problem Definition

This text is written as a master thesis at Linköping University by Johann Pavski with Nils-Hassan Quttineh and Sebastian Lindquist at Ericsson as supervisors and Elina Rönberg as examiner, with MikTex in \LaTeX{} in 2015.

This first chapter will define the problem, give some background on telecommunication, formulate the main questions we wish to answer and describe what topics are covered.

1.1 Background

Within the field of telecommunications, we define base stations as antennas and cells as the area that an antenna covers.

Due to the fact that in reality, the environment is very scattered and unpredictable, cells follow very scattered and unpredictable coverage patterns. Buildings, trees, different terrain heights, the weather and of course how the UE moves through this terrain lead to shadowing and fading, which in turn have an influence on the Quality of Service. If two users use the same frequency and are too close to each other, there will be interference. In order to avoid this, there is a frequency plan that sees to that frequencies are not reused in neighbouring cells.

When a UE moves in this environment, it inevitably crosses the border between cells and a handover has to be performed. This process is non-trivial, since the cell border has no clear, constant definition. Instead, the UE measures the Quality of Service of all neighboring base stations and transmits that list to the base station controller, the BSC. The BSC controls up to 2000 cells and within it, an algorithm decides which base station the UE should be connected to. This calculation is performed for each user equipment in each cell every 480ms. Handovers are costly both computationally and energy wise, they are also the reason for a high percentage of the call drops, i.e. terminated calls and thus unhappy users.

This algorithm – here called E3 for legal reasons – has slightly different parameters for each cell, since every cell is different. The E3-parameters currently have to be configured manually by field engineers, which is expensive and time
consuming and therefore often avoided by operators. Instead, a standard set of parameters is used that will work, but those are not optimal. Possible low QoS-areas are simply covered by more base stations. Thus, there is tremendous "over coverage" and many unnecessary handovers.

Ericsson would like to implement a feature that automatically sets these parameters based on the provided QoS. That means creating a feature that for each cell does the following:

Current QoS $\Rightarrow$ Automatic reconfiguration of the E3 $\Rightarrow$ better QoS.

This feature should be able to dynamically adapt the E3 function to changes in the environment such as a new building that causes changes in fading patterns, a new street that causes an increase in the number of UEs, the appearance of a new base station or any other similar event. Note that, within this project, reconfiguring the E3 simply means manipulating four parameters and each time such a manipulation has taken place, some time has to pass before we can, with a certain high enough confidence, state which impact this manipulation has had on the QoS.

Although the E3 is applied on each UE separately, the parameters (PoIs) that should be manipulated are either cell or cell-to-cell parameters. There is also a ripple effect, that means that a certain choice of PoIs for one cell will have an influence on the choice of PoIs in some neighboring cells. This mutual influence is simply referred to as "the ripple effect".

The ambition of this thesis is to make a prestudy on this feedback loop between QoS and the E3-configuration in a simulated environment and to make a proposal for how, in principal, the feature could be implemented.

1.2 Formalities

The project has been carried out at Telefonaktiebolaget L. M. Ericsson in Linköping by two thesis workers, Ahmad Bazzari and Johann Pavski (author of this Thesis). Johann Pavski tackled the task from a mathematical point of view, while Ahmad Bazzari provided the necessary telecommunications expertise. Two separate reports were written, this one at Linköpings University, while Ahmad Bazzaris report [1] was handed in at KTH Royal Institute of Technology in Stockholm. The reports refer to each other, but will be graded separately.

For confidentiality reasons, some telecommunications details and graphs may be altered in this report. In general, it is not stated what a parameter actually does, it is simply referred to as for example "POI1, cell to cell parameter". From a mathematical point of view, this does not alter the problem or the description of how it is solved. A third, internal report will be written that was handed in to Ericsson directly, where the correct and explicit numbers and definitions are stated. This third report was written by both Johann Pavski and Ahmad Bazzari together.

All figures in this report were produced by the author using Matlab [2], unless otherwise specified.
1.3 Scope of the Thesis

Purpose

The reason for initiating this project is the fact that the handover algorithm at Ericsson in its current form has room for improvement. Optimizing in this case refers to adapting it to each cell and its situation individually, which, as of now, is being done either manually or not at all. Ericsson wishes to implement a feature for automated adaptation. The problem statement is: How can the adaptation of the handover algorithm to individual cells and changing traffic- or other situations be automated?

Objective

The objective of this thesis is, in accordance with the problem statement, to find a way to let the handover algorithm automatically choose its PoIs. This is possible by building optimization algorithms that choose parameters for the E3 handover algorithm based on the measured QoS and evaluate the reaction of the radio environment.

Method

The methods chosen to find an answer to the problem statement were, among other things, to develop two algorithms that can autonomously choose parameters for the handover algorithm. One algorithm is rooted in the principles of optimization, while the other is following a control theory approach. Literature on previous similar attempts was considered. The algorithms were evaluated using an existing simulator, that has been expanded to allow testing of the handover functionality. A set of scenarios, for example different traffic and radio resource situations, was chosen to be simulated. An objective function weighing together different metrics of the QoS was developed and evaluated.

Limitations

Two primary limitations were in place: A time constraint on how long the project may take and the requirement of real world implementability. The latter means that computation times can not exceed certain limits since new information needs to be processed and used very fast in the actual application. While vast amounts of data are available, its usage has been constrained for this project. KPIs, that is event counters, are being used as data sources instead of full knowledge of the radio environment and all users in the system. Thus an integer linear programming formulation with full knowledge of this data is, to give an example, out of the question. Such a formulation is obtainable, but it would be too complex and it would take unreasonable time to solve. Additionally, data storage would pose an unreasonably large strain on the system considering the benefits.

We also limit the project to only consider one handover algorithm. Several others exist both within Ericsson, and owned by other corporations. In order to avoid interference, there is a frequency plan that sees to it that frequencies are not reused in neighbouring cells. This is for the most part done
automatically (Automatic Frequency Load Planning - AFLP). Within this thesis, we only consider AFLP networks. We limit the size of the network we simulate in order to be able to run what is considered to be a large amount of simulations with many different PoIs. The network simulated in this project consists of twelve cells. The control theory approach operates on all PoIs within the network, while the optimization approach operates on three cells within the twelve cell network. The E3 algorithm uses around 40 PoIs out of which we consider four. These four were deemed to be the most important due to their close relationship with the radio environment within the cell. There are around 2500 outputs from the network that can be considered, we will however consider five of them since these five are the ones most directly related to the handover. More outputs are interesting, but are not featured in the simulator and their inclusion was omitted due to time constraints.

1.4 Topics Covered

- Telecommunications Background
  The context of the problem is presented, important notions from telecommunications are introduced. Design variables (PoIs) and objective values (KPIs) are defined.

- Theory
  Literature and methods are presented.

- Mathematical Problem Formulation
  Our approaches are explained, partly through examples. The problems are formulated mathematically.

- Results
  Trial runs of the two algorithm approaches are presented.

- Discussion and Future Work
  The results are discussed and a list of approaches that were abandoned due to time limitations is presented.

- Conclusion
  A short summary of how the project turned out and when and how the results are useful.
Chapter 2

Telecommunications

Background

This chapter presents the problem in a telecommunications context. Some differences between the two approaches presented in this text are addressed. For a deeper understanding of the GSM network and handovers please refer to Halonen et al. in [3].

2.1 The Telecommunications Problem

This section is a schematic of the problem and data flow from a telecommunications point of view.

2.1.1 Schematic of the cell plan in a cellular network

Consider the schematic of cells in Figure 2.1. A cell is an antenna and its respective area of coverage. The hexagonal shape follows from overlapping circular coverage. The stars represent the position of the antennas in some of the cells (not all the antennas are drawn for the sake of simplicity).

Suppose now that the squiggly line in Figure 2.1 represents the path of a UE moving through the network. It starts a call in cell 1, and moves in the direction of cell 2. The E3 chooses the most suitable cell to be connected to every 480 ms. In order to make this decision, several measurements are considered and weighed against each other, using filters for the measurements and several offsets and hysteresis.

Say that the UE is fairly close to the antenna in cell 1, and that this antenna thus provides the best coverage. Moving closer and closer to cell 2, the signal strength from the antenna in cell 2 gradually increases until it is stronger than the signal from the antenna in cell 1, at which point the UE's connection will be transferred to the antenna in cell 2 instead.

Note that the cell border is defined by the algorithm, individually for each UE and that it is not predefined in the way that the lines are drawn in Figure 2.1. The procedure of ranking which cell is the best for a UE and then transferring the call is what we refer to as the handover.
The case described above is the absolutely simplest. Consider also the following two cases, that need to be addressed:

- Consider the squiggly line in Figure 2.2, representing a UE that is moving very close to a cell border and crossing it several times before definitely moving into one cell. In order to avoid “ping-pong handovers”, there is a hysteresis between cells that needs to be crossed before the handover is made to the new cell. We refer to the hysteresis as PoI 1 in this project.

- A UE is moving towards a new cell, but the signal quality in the old cell is very bad - for example due to high interference from many other UEs - and it is suitable to connect to another cell earlier than usual. The limit of how much earlier such a connection should be made is defined by a parameter in the E3. We refer to this parameter as PoI 2 within this project.
We also consider two additional parameters in the E3. Unlike PoI 1 and PoI 2, these are generally not considered in attempts to automate the choice of parameters for the handover algorithm.

- The filter length of the filtering of the signal strength that we measure from the different antennas. We refer to this parameter by the name of PoI 3 in this project.
- The filter length of the filtering of the signal quality in the serving cell, that is the cell that the UE is currently connected to. We refer to this parameter by the name of PoI 4 in this project.

Note that in reality, the cell distribution is anything but as regular as in Figure 2.1. While each cell has six geographical neighbors in Figure 2.1, a cell considers at least 6 and at most 32 other cells as its possible neighbors and candidates for a handover. A real life cell distribution could look as in Figure 2.3. In order to have a measure of how well a network performs, there are counters of events in the system. An event can be anything from a handover to a change in the energy an antenna outputs at a given point. There are about 2500 of these counters in the entire network, five of which we consider interesting within this project. We call these counters KPIs. The KPIs will be the metrics we use to decide on PoIs, and they will eventually be part of the cost function that we want to reduce in our optimization problem.
We consider the following KPIs:

- KPI 1 is the signal strength measured by the UE.
- KPI 2 is the speech quality for the UE.
- KPI 3 is the number of handovers in the cell per time unit.
- KPI 4 is the number of erroneous bits delivered to the UE.
- KPI 5 is the ratio of handover due to bad conditions in the cell and the total number of handover. This KPI is computed per time unit.

Each of these KPIs is formulated in such a way that reducing it is desirable. Figure 2.4 is a representation of the network that we simulated during the course of this project. Because of the effect of the irregularly distributed cells, we consider all cells in this network to be neighbours of all other cells.

Figure 2.4: The network plan we use in our simulations. Note that there is a wrap-around in the simulations, so a UE exiting the network to the left will reenter the network to the right. The three cells that are colored green are for example next to each other. The units on the x- and y-axes are meters. The figure has been produced by Ericsson.

\[\text{A time unit is in this case not a second, but a SACCH-period, that is 480ms. This is the time it takes the network to compute decisions on whether or not a handover should be performed.}\]
2.1.2 Schematic of the Handover Algorithm

Figure 2.5 represents a simplified version of the data flow between the E3 handover algorithm and the measurements of KPIs from the system. These measurements deliver an overview of the QoS in each cell. The result of the thesis is represented by the “Our Algorithm”-box which automatically chooses the cell parameters PoI and thereby changes the KPIs. This model of the system will be used in the approach that we call Higher Dimensional Search or HDS.

![Figure 2.5: Dataflow and ranking algorithm.](image)

While information on individual KPIs and their relation to each other, as well as their context in the system, is used in the HDS, the optimization approach models the system in a much easier way. Summing and weighing the KPIs into a singular cost value, we model the system to be optimized in the way represented by Figure 2.6. Note that in this approach, we consider four PoIs.

![Figure 2.6: How we model the system during the optimization approach. 4 values of PoIs are fed into a black box, and a single value is returned.](image)
2.2 Definition of the design variables

In optimization parlance, a problem has a set of design variables over which the objective function is to be optimized. The PoIs introduced in the previous section are the design variables. There are four PoIs, each of them integer valued within the following range:

1. PoI 1: 0-7
2. PoI 2: 0-7
3. PoI 3: 2-14
4. PoI 4: 4-14

There are thus \(8 \cdot 8 \cdot 11 \cdot 13 = 9152\) possible combinations of values. Note however that PoI 1 and PoI 2 can be set individually for each cell-to-cell-relation, while PoI 3 and PoI 4 can be set individually for each cell. The following formula allows for calculation of the number of combinations of design variables:

\[
\frac{8 \cdot c \cdot n_c}{2} \cdot \frac{8 \cdot c \cdot n_c}{2} \cdot 11 \cdot c \cdot 13 \cdot c
\]

The expression \(\frac{8 \cdot c \cdot n_c}{2}\) comes from 8 values for \(c\) cells that each have \(n_c\) neighbors. We divide by two since the cell to cell parameters are symmetrical, that means \(Pol_{nc} = Pol_{cn}\). For our simulated network, there are \(c = 12\) cells and each cell has \(n = 11\) neighbors, amounting to a total of approximately 5.7 billion combinations of PoIs in this network. Within the two approaches, we consider rather small subsets of these in one of the following two ways:

- We change all PoIs of the same kind simultaneously - for example all PoI 1 change in the same iteration of the algorithm for all cell-to-cell-relations or all PoI 3 change in the same iteration of the algorithm for all cells. In this case, we consider a set of 9152 PoI-combinations. This is done in the Costly Global Optimization or CGO-approach.

- We start at a certain set of PoIs, such as for example \(\{Pol 1, Pol 2, Pol 3, Pol 4\} = (3, 0, 4, 12)\). In each adaptation of the E3, each PoI may only change one step. This allows for example the following solutions: \((3, 0, 4, 13), (4, 1, 4, 11)\). Effectively, this method searches locally and it considers the entire set of more than 5.7 billion combinations of PoIs. This is done in the HDS-approach.
2.3 Definition of the Cost Function

This section describes the attempt to model the performance of a network on cell-level. KPIs are usually considered on UE-level, but since we want to operate on network level when optimizing, or for that matter controlling, the KPIs need to be introduced on a cell level.

The five KPIs were introduced in Section 2.1.1 and the cost function is modeled as a weighted sum of these. The following aspects are taken into account:

1. In order to measure cell wide or network wide activity, the KPIs are averaged over a cell and, when needed, over the entire network. In the case of the KPIs that are measured per time unit, such as the handover rate, this is a moving average. We denote the averaged KPIs by AKPIs.

2. The KPIs that are used to measure the handover performance are not equally important. They are therefore relatively weighted with a percentage $\alpha_i$, giving the most important KPI, that is the one that has to be reduced first and foremost, the highest weight. Each $\alpha_i$ being a percentage, we have $\sum_{i} \alpha_i = 1$, $i = 1 \ldots 5$.

3. We consider some AKPIs to be low enough when they are below a threshold value $\beta_i$, $i = 1 \ldots 5$. This is formulated by introducing the notation $\max(AKPI_i - \beta, 0)$.

4. The KPIs measure different events and thus have different magnitudes. In the attempt to normalize them, we divide each AKPI by a value $\gamma_i$, $i = 1 \ldots 5$.

Considering number 2 and 3 in this list together lead to the following behavior: When the most important AKPI, that is the one with the highest $\alpha_i$, is below its threshold $\beta$, it will no longer be reduced by the algorithm, however an AKPI with lower $\alpha$ will now be of relatively higher importance and is thus reduced. The reason for this formulation is that some KPIs have opposing objectives, and would require a multi-objective optimization approach. The relative weights and the thresholds allow for an overall cost function where some objectives are only considered when the more important ones are already satisfied.

The cost function is:

$$C(PoI) = \sum_{i=1}^{5} \alpha_i \cdot 1/\gamma_i \cdot \max(AKPI_i - \beta_i, 0) \quad (2.1)$$

The AKPIs are ultimately functions of the PoIs. This function can be formulated locally over a cell or over a set of cells but also globally over a network by manipulating how much the averages cover. In order to have a simple notation, we introduce $C_{cell}$ for a cost function that describes the performance in one cell and $C_{network}$ for the same thing in the entire network. The objective function $C$ is strictly positive.

As a side note, consider the number of PoI-combinations previously derived - more than 5.7 billion - and multiply it by the number of mobile units in such a network, which is, to take a typical number, in the neighborhood of several thousand per cell. Optimizing a network on UE-level has been tried and would theoretically “tidy up” a network, that means reduce interference levels and increase radio access availability - that means the number of possible connections per antenna - considerably. It is, however, computationally out of the question in the real application. Consider as an example [4].
Chapter 3

Theory

The principles of costly global optimization are introduced. Some earlier attempts to implement algorithms with similar goals as ours are presented. Fuzzy logic is introduced.

3.1 Introduction

In order to construct the two algorithms, one global and one local, the following methods and concepts are introduced:

- **Costly Global Optimization:**
  Sampling means measuring the KPIs of the network when subjected to a certain set of PoI. In order to tamper with the network as little as possible while retrieving as much information as possible, the behaviour of the KPIs needs to be modeled using as few samples as possible. Costly Global Optimization is a method developed to fulfill this need by strategically sampling a space of design variables, in this case the PoIs, and interpolating the rest of the space using for example radial basis functions.

- **Feedback Loops using Fuzzy Logic:**
  Fuzzy logic designates a method for mathematically handling linguistic notions such as “a little” and “very”. In classical boolean logic a variable can either be in a set or not, while fuzzy logic expands this to the possibility of a variable being partly in a set. When a field engineer formulates how to react to a KPI being “a little too high”, fuzzy logic can be used to define the set “high” and checking how much the objective KPI is in this set.
  In the locally operating algorithm, fuzzy logic is used to implement real world engineering knowledge.

3.2 Costly Global Optimization

Costly Global Optimization refers to an area of optimization that is used when a function evaluation is very costly, be it financially or computationally. Any
method that follows the concepts of costly global optimization requires relatively few function evaluations, as well as no information on the derivative, in order to produce a good estimation of an optimal value of a function. We will refer to the algorithm that is built upon the CGO principle during this thesis project as the CGO algorithm.

By the nature of the method, costly global optimization considers the function to be evaluated as a black box. Using as few function evaluations as possible, the entire function space is approximated using radial basis functions such as the cubic spline. Radial basis functions and global optimization of costly functions were linked together in 2001 by Gutmann, making this a relatively new and active area of research. An excellent overview and introduction into different methods that are used for interpolation and choice of sample points is given by Jones in [6]. An example for early implementations is Holmström et al. in [7] and Shoemaker et al. in [8].

While we operate on a simulator, the output of which is deterministic, the real network application will have to model the objective values as realizations of stochastic processes. Methods to handle this are very similar to the one presented here and are presented by among others Gutmann. Jones et al. presents a good introduction to the Design and Analysis of Computer Experiments (DACE) framework in [9]. Unlike radial basis function interpolation, the DACE method models the samples as realizations of stochastic processes. In [10], Yao et al. suggest a surrogate-based optimization method that uses neural networks to refine the choice of radial basis functions. This method takes the gradient of the objective function into account in order to enhance the accuracy of the surrogate surface. In the same paper, the authors suggest a concept to estimate a lower bound of the objective function based on the prediction error. Reaching a neighborhood of this estimation provides a possibility for a stopping criterion.

An example for uses of surrogate based global optimization is presented by Ayed et al. in [11]. Electromagnetic devices are usually evaluated using the finite element method, a computationally expensive calculation of differential equations. The application lends itself to interpolation through surrogate surfaces due to the smoothness of many electromagnetic phenomena. Areas of irregularity, such as near the edge of some piece of metal, are known a priori, so that sampling points can be chosen intelligently.

Referring to Quttineh in [12], we present the following example in order to explain the way the algorithm works. For details on how the method has been adapted to the project, please refer to Section 4.2.

Suppose for the purpose of an example some costly function $f(x)$ in two dimensions. Information on the derivative is not available. We see this function as a black box into which some design variables are sent and a function value is returned. Suppose also some set of box constraints for the design variables, that is a set of upper and lower boundaries for their values. An exhaustive - and very

---

1 Radial basis functions are not a necessity for interpolations, other possibilities are Gaussian process regression or using wavelets. Other radial basis functions than the cubic spline are possible, but were omitted in this project since the cubic spline produced good results.

2 Unlike the deterministic approach used in this thesis, DACE employs a process where statistical regression and cross correlation between the samples is used to maximize the likelihood of the sample. Mean and variance of the samples are thus taken into account.

3 This is for our purpose roughly the numerical derivative of the objective function as far as it can be estimated from the samples taken.
expensive - evaluation of $f(x)$ would generate Figure 3.1. The function $f(x)$ has been evaluated at several thousand points in order to create Figure 3.1.

In order to solve the problem $\min f(x)$, s.t. $l \leq x \leq u$, the principles of costly global optimization are used. A surrogate model, that is an interpolation surface that the algorithm operates on, is produced. Given some set of points where the costly function $f(x)$ has been sampled, the surrogate model is a function that approximates $f(x)$ in the points not yet sampled. Given a sufficient amount of “good” points sampled, optimizing the surrogate function gives a new set of points where we predict that an optimum may be found.

In [12], a pseudo code for finding a minimum of $f(x)$ is described. This pseudo code is reproduced in Algorithm 1. The algorithm is formulated specifically for minimization problems.

**Algorithm 1** Pseudo Code for Surrogate Model Algorithms

1: Find $n \geq d + 1$ initial sample points $x$ using some method
2: Compute costly $f(x)$ for initial set of $n$ points. Best point($x_{Min}, f_{Min}$)
3: while $n \leq MAXITER$ or $f_{Min} \geq MAXVALUE$ do
4: Use the sampled points $x$ to build a response surface model as an approximation of the costly function $f(x)$.
5: Find a new point $x_{n+1}$ to sample using some merit function.
6: Calculate the costly function value $f(x_{n+1})$.
7: Update best point ($x_{Min}, f_{Min}$) if $f(x_{n+1}) < f_{Min}$.
8: Update $n := n + 1$.
9: end while

The parameter $d$ is the dimension of the problem, $MAXITER$ is some upper bound on the number of function evaluations that we allow, and $MAXVALUE$ is an upper bound on the function value that we want to achieve. Whichever of $MAXITER$ and $MAXVALUE$ is reached first, serves as stopping criterion.
The choice of initial sample points is non trivial and possible options and concepts are elaborated by Quttineh et al. in [13].

Figure 3.2: Example of how the original function is captured better and better over the course of iterations.

In Figure 3.2 note how only 15 sampled points already give a good idea on where a global minimum may be found. In the lower right corner we see an interpolation with 90 sampled points, capturing all major characteristics of the original function. Compared to Figure 3.1, the number of evaluations is by far lower. While the function in this example is easy and cheap to evaluate, in the case of a costly function where each evaluation takes several hours, the interpolating approach reduces the cost severely.

**How to build a response surface model**

Let the set of sampled points be \( x = \{ x^{(1)}, x^{(2)}, \ldots, x^{(n)} \} \). This means that \( x \) is the set of all feasible solutions \( x \) that have been sampled and that gives us information about the system. Furthermore, let \( y \) be the corresponding objective values. A function is needed that uses these sampled points to cheaply
interpolate the entire function space. Our work is based on Powell in [14] and uses radial basis functions. We designate $n$ to be the number of sampled points and $d$ is the number of dimensions.

The following formula describes how to interpolate all $x$ given a set of sampled points $x$.

$$ S(x) = \sum_{i=1}^{n} \lambda_i \cdot \phi(\|x^{(i)} - x\|_2^2) + b^T x + a $$

where $\lambda \in \mathbb{R}^n$, $b \in \mathbb{R}^d$ and $a \in \mathbb{R}$. Furthermore, $\phi$ is the cubic spline $\phi(r) = r^3$.

The parameters $\lambda, b,$ and $a$ can be computed by solving the following system of linear equations:

$$ \begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} y \end{pmatrix}, $$

where $\Phi$ is an $n \times n$ matrix with $\Phi_{ij} = \phi(\|x^i - x^j\|_2)$ and

$$ P = \begin{pmatrix} x^{(1)} \\ 1 \\ x^{(2)} \\ 1 \\ \vdots \\ x^{(n)} \\ 1 \end{pmatrix}^T, \lambda = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{pmatrix}, c = \begin{pmatrix} b_1 \\ \vdots \\ b_d \end{pmatrix}, y = \begin{pmatrix} f(x^{(1)}) \\ \vdots \\ f(x^{(n)}) \end{pmatrix}. $$

As long as the rank of $P$ is $d + 1$, where $d$ is the dimension of the problem, this linear system of equations has a unique solution and gives us a radial basis function (RBF) interpolation. That means that the above defined surrogate model $S(x)$ interpolates the costly function $f(x)$, based on the sampled points $x$.

### 3.3 Self Optimizing Handover features in Telecommunication Networks

A short overview of literature that was considered when building the HDS algorithm. The term “Fuzzy Logic” is explained in more detail in Section 4.3.2.

### Introduction to Fuzzy Logic

Unlike boolean logic, which allows a statement or a variable to take on two values, either 1 or 0, fuzzy logic permits a variable to have any value between 0 and 1. A truth value can linguistically formulated be anywhere between completely true and completely false. While the statement $2 + 2 = 4$ is an absolute truth, water can be “a little cold” as well as “not very warm” at the same time. Fuzzy logic allows for a mathematical notation of this fact by defining two sets: “warm” and “cold” and stating how much $x = “the temperature of the water”$ is in each of these sets. For a more thorough explanation of how fuzzy logic can be used, we refer to Espinosa et al. in [15].
Automatic Link Balancing Using Fuzzy Logic Control of Handover Parameter

Controlling a simulated network environment using feedback loops has been tried by, among others, d’Orey in [16]. Here, a fuzzy control algorithm uses information from several KPIs to balance individual link gain (that is for the purpose of this thesis a metric for the QoS) and the radio resource consumption, in the case of [16] that is the power emitted from the antenna. A trade-off between uplink and downlink gains was observed - something that we could not work on in this thesis, since the simulator does not have an uplink feature. Note that this study was performed in 3G, not GSM.

Given a set of fuzzy logical statements in order to have some knowledge of the system, the paper suggests the following algorithm: A feedback loop between network measurements and network parameters is devised. KPIs are stored in a “knowledge database” and are statistically analyzed. Whenever a cell matches a reference criterion, that means there is enough information for a decision, the algorithm makes a decision based on both fuzzy logical rules and the memory. The study concludes that a fuzzy controlling algorithm allows to adjust static network and cell parameters in a dynamical environment.

Self-Optimization Of Handover Parameters In LTE Networks

In [17], Capdevielle et al. use an optimization approach in the attempt to minimize handover late detections, that is when a handover happens too late, and handover failure rates - these are the KPI. The PoIs being changed are handover margin, a3-offset and time to trigger.

While this study was performed in LTE, which shares even fewer similarities with GSM than the study referred to in the previous section, its choice of PoIs and KPIs is closer to the one in this thesis.
Capdevielle does not use a controlling approach but an optimization feedback loop featuring simulated annealing. This approach considers users individually and requires a much faster simulator than we had at our disposal in the course of this thesis, allowing for more function evaluations and the adjustment phase that is integral to simulated annealing. Capdevielle concludes that the approach led to a significant improvement of the KPIs, especially the handover failure rate for high-speed scenarios.

**Handover Parameter Optimization In WCDMA Using Fuzzy Controlling**

In [18], Werner et al. suggest a similar approach as the one presented in [16]. The PoI used is the cell individual offset (that is not the offset we treat in this thesis). The KPIs used are the network load and the outage probability. The major interest of [18] is the fact that the algorithm has been adapted to and tried out in a real city and not just in a simulator, although extensive simulations have been performed. Both the simulations and the test in the real environment proved the approach to be sound; KPIs were reduced significantly.

---

4 Simulated annealing is a heuristic that assigns temperatures $T$ to areas of design variables. The optimization algorithm starts at a point $x_i$ chosen by some initial design, upon which it randomly jumps to another point $x_{i+1}$ in some neighborhood of $x_i$. If the objective value in $x_{i+1}$ is lower than in any previously sampled point (“$\Delta \leq 0$”), then $x_{i+1}$ is accepted. If not, $x_{i+1}$ is accepted with a probability $P$ based on $T$ in that point (and rejected with $1-P$). A typical choice of probability distribution is $P(acceptance) = \exp\left(\frac{-\Delta}{T}\right)$. By analogy, the random jumps are more probable in “warmer regions”, while less movement occurs in areas with a low $T$ value. At appropriate points in the algorithm, the temperature is adjusted.
Chapter 4

Mathematical Problem Formulation

We introduce the problem from a mathematical point of view. Note that the spirit in which we started the project was to operate globally on a smaller box of constraints first. Within this box, we would find a suitable starting point from where to locally search in a higher dimensional set of constraints. The globally operating algorithm is called the CGO-algorithm, the second one is called the HDS-algorithm. Due to time constraints, the two approaches were never combined and are presented separately.

4.1 Introduction

Referring to Section 2.3, we model the output from the simulated network as a black box function $C(x)$ that we want to minimize, where $x$ is a set of PoIs. The output - previously AKPI - is $y_i(x), i = 1, \ldots, 5$.

Consider the following table for the sizes of the sets of variables, depending on the variable type, as well as the upper and lower limit of the variable. The parameters $c$ and $n_c$ are the number of cells and the number of neighbors for each cell. The parameter $k_t$ is the number of discrete values within the constraints for variables of type $t$.

<table>
<thead>
<tr>
<th>$t$</th>
<th>Size of $N_t$</th>
<th>$l_t$</th>
<th>$u_t$</th>
<th>$k_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$0.5 \cdot c \cdot n_c$</td>
<td>0</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>2</td>
<td>$0.5 \cdot c \cdot n_c$</td>
<td>0</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>$c$</td>
<td>2</td>
<td>14</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>$c$</td>
<td>4</td>
<td>14</td>
<td>11</td>
</tr>
</tbody>
</table>

Where $N_t = \begin{cases} 
\text{The numbered set of all cells} & t = 1, 2 \\
\text{The numbered set of all cell-to-cell-relations} & t = 3, 4 
\end{cases}$
Given this table, we define:

\[ x_{jt} = \text{Value of the PoI of type } t, \text{ cell index or cell-to-cell index } j. \]

\[ X = \text{The set of feasible values of } x_{jt}, \text{ defined by the hyperrectangle limited by } l_t \text{ and } u_t. \]

In a more mathematical way, we define the parameter 

\[ d = \sum_{t=1}^{4} N_t \]

and the set of feasible solutions 

\[ X = \{ x \in \mathbb{N}^d : l_t \leq x_{jt} \leq u_t, \ t = 1 \ldots 4, \ j \in N_t \} \]

as well as 

\[ \bar{x} = \text{one specific feasible solution, that means a point in the space } X. \]

We have the optimization problem:

\[ \min_{x \in X} C(x) = \sum_{i=1}^{5} \alpha_i \cdot 1/\gamma_i \cdot \max(y_i(x) - \beta_i, 0) \quad (4.1) \]

The function \( y_i \) maps a discretely valued vector \( x \) onto a real value, and the function \( C(x) \) maps \( y_i, i = 1, \ldots, 5 \), onto a single real value:

\[ C : \mathbb{X}^d \xrightarrow{y(x)} \mathbb{R}^m \xrightarrow{C(y)} \mathbb{R} \]

### 4.2 Costly Global Optimization

For an introduction to the concept of Costly Global Optimization, please refer to Section 3.2.

For methods based on surrogate models to work satisfactorily, it is desirable to have ten or less design variables. We thus consider the subset \( \bar{X} \) of \( X \) where

\[ \bar{X} = \{ x \in \mathbb{N}^d : l_t \leq x_{jt} \leq u_t, \ t = 1 \ldots 4, \ j \in N_t \}. \]

Hence, \( \bar{x}_t = x_{jt} \) \( \forall j, t \), that means we choose the same values for each type of PoI, disregarding which cell or cell-to-cell relation it designates.

We have the following subproblem of (4.1):

\[ \min_{\bar{x} \in \bar{X}} C(\bar{x}) = \sum_{i=1}^{5} \alpha_i \cdot 1/\gamma_i \cdot \max(y_i(\bar{x}) - \beta_i, 0) \quad (4.3) \]

The number of design variables is \( |T| = 4 \). Hence, \( |\bar{X}| = \prod_{t=1}^{4} k_t = 9152 \) combinations of feasible \( \bar{x} \), independent of the number of cells or the cell-to-cell relations.

---

1Because sampling objective values is costly, we want to sample as few points as possible. The higher the number of design variables and therefore the dimensionality of the problem, the more sparse the samples become. This sparsity becomes a problem during interpolation, because it affects the accuracy of the surrogate surface, there is simply to much space to interpolate. Thus, the surrogate surface becomes unreliable when the objective function has too many design variables.
4.2. Costly Global Optimization

4.2.1 Adaptation of the Costly Global Optimization Algorithm

In order to apply Costly Global Optimization to the problem studied in this thesis, a scenario has been extensively simulated. Extensive means that all feasible solutions in $\tilde{X}$ has been evaluated, amounting to over 9152 simulations. We call this scenario the test case. A large amount of trial runs allowed us to find good parameters that work in most cases. Three points have to be addressed:

1. Which method of initial sampling to choose:
   Several standardized ways of choosing initial points to evaluate exist. Such a method should generally cover the space of solutions and as many characteristics of the original function as possible. Obviously the number of function evaluations should be kept low at the same time. Latin hypercube sampling has proven to fulfill these requirements the best. It covers the given space rather evenly while avoiding to place sample points on a function characteristic more than once.
   Finding a satisfactory solution was usually possible within less than three iterations after initial sampling in the test case, however at the cost of the fact that some higher values have been evaluated while sampling. Note that we can not sample less than five points since the rank of $P$ in (3.2) needs to be $d + 1$ where $d$ is four (the four POIs).

2. How to choose $x_{n+1}$:
   There are generally two approaches to choosing the next point that should be evaluated. Either the minimum of the response surface is taken - this usually leads to getting stuck in a local minimum after some time - or the point with the largest minimum distance to all the already evaluated points is chosen - thus giving a better approximation of the given response surface and providing a jump away from possible local minima. We call the first approach simply minimum of the surrogate surface and the second one maximin of the distance to previously evaluated points. Within Algorithm 2 these approaches are referred to the minimum and the maximin.
   The previously mentioned extensive simulations proved that the maximin approach almost always had a negative impact on the amount of iterations it takes to reach a satisfying function value. The reason for that is the shape that the cost function has: it is strictly positive, however mostly close to zero. Using very few samples, the cubic spline interpolation interpolated large negative values. The larger those negative values are, the less accurate the interpolation. Choosing the minimum of the interpolation surface for the next sample thus provides the maximum improvement of the interpolation that is possible. The maximin is only used to escape local minima.
   In each iteration, the pseudo code in Algorithm 2 chooses a $x_{n+1}$. Line 4 in Algorithm 2 chooses $x_{n+1}$ to be some linear combination of the minimum of the response surface and a point that is in some relatively unexplored region of the space of design variables. An appropriate point in such a region is chosen by the maximin function.

---

2This is an observation from experience with the test case, it does not hold in general.
Algorithm 2 Pseudo Code for Choice of $x_{n+1}$

1: $\alpha = \text{random between 0 and 1}$
2: $x_{n+1} = \min(\text{response surface})$
3: if $|x_{n+1} - x_n| \leq \epsilon$ or $|f_{n-1} - f_n| \leq \delta$ then
4: $x_{n+1} = \text{round}(\alpha \cdot \maximin + (1 - \alpha) \cdot \min(\text{response surface}))$
5: end if

In Algorithm 2, the parameters $\epsilon$ and $\delta$ are strictly positive and need to be chosen based on experience with the objective function. While a radial basis function interpolation $S(x)$ of the costly function $f(x)$ allows for cheap and fast evaluations, finding a minimum on $S(x)$ is in general by itself a global optimization problem and can be very challenging. In our case, due to the discrete all-integer space of solutions $x$ containing only 9152 points, we can simply evaluate all these points and find the minimum.

3. How to choose a good stopping criterion:
Again, based on large amounts of trial runs, a good stopping criterion seems to be reaching a cost below 0.6. The maximum number of function evaluations has been chosen to be 8, since the real world application is a live network that can not be tampered with too much. Additionally, the cost decreased exponentially, which means that the amount of iterations needed to decrease the cost significantly after the first few iterations increases dramatically.

Please refer to Figure 4.1: The curve is very flat for a stopping criterion of $\text{cost} \leq 1$, however the amount of iterations needed to converge is out-of-the-question large when the stopping criterion is $\text{cost} \leq 0.5$. The reason for the variance approaching zero for lower stopping criteria is that there is a cutoff number of iterations. The more the trials diverge, the less the variance.

![Figure 4.1: A trial run on the test case with 100 runs per measurement.](image-url)
4.3 Higher Dimensional Search

The Higher Dimensional Search (HDS) gets its name from the fact that we do not work on a subset of $X$:
We consider each cell and each cell-to-cell relation individually, however, unlike in the CGO-approach, we choose a starting set of PoIs and move very locally around it to find a combination more suitable to the cell or the cell-to-cell relation. Thus if we were to allow a sufficiently high number of iterations, any point in $X$ could be reached. In each iteration, we only consider some neighborhood of the current solution, allowing each design variable to possibly change one step up or down.

The HDS approach requires a model of the system, that is a set of rules which defines the relations between $y(x)$ and $x$.

Note that the HDS is not a local search in the mathematical sense. In a local search, each iteration evaluates some neighborhood around a current solution, and the best point in this neighborhood is then accepted. In our algorithm, we only ever evaluate one point in each iteration and even accept points that result in worse KPIs.

We have the following optimization problem. It is formulated in the same way as (4.1):

$$\min_{x \in X} C(x) = \sum_{i=1}^{5} \alpha_i \cdot \frac{1}{\gamma_i} \cdot \max(y_i(x) - \beta_i, 0)$$

The number of elements in $X$ is $\prod_{t=1}^{4} N_t \cdot k_t = 2288 \cdot c^4 \cdot n_c^2$.
In this context, we use the cost function only as a possibility to see how well the algorithm performs but not as a source of information. We use individual, cell internal, KPIs in order to make decisions.

4.3.1 Structure of the Higher Dimensional Search Algorithm

The HDS algorithm borrows its basic structure from a control loop. For each cell, we have the loop represented in Figure 4.2. The box “AKPI to PoI mapping” is described in more detail in Figure 4.3.

![Figure 4.2: Dataflow in the HDS algorithm. AKPIs are averaged over cells or summed over C2C-relations.](image-url)
Referring to Figure 4.2 and 4.3 there are three main steps in choosing the new $x_{jt}$. The first step is the significance check: an AKPI is not significant if it is relatively too small. An example is very few UE's crossing the border between two cells. In such a case, a decision on a PoI-change is impossible.

The second step is the mapping, which is the mathematically interesting part. The starting point of the mapping were qualitative statements from engineering experience, such as: “Increasing PoI $x_t$ lowers KPI $y_i$ a little bit, but increases KPI $y_j$ fairly much”. It comes naturally to implement such statements in the form of fuzzy logic. The decision made in this part of the algorithm is:

“Given $KPI_i, i = 1, \ldots, 5$, should $x_{jt}, t \in T, j \in N_t$ increase, decrease or remain the same?”

The third step in Figure 4.3 ensures symmetric choices of cell-to-cell PoIs. It takes the form of a truth table.

### 4.3.2 Fuzzy Logic

In our formulation of the mapping from KPI to PoIs, we depend on the first chapter of [15]. Using the example in Figure 3.3 concerning the temperature of water, we define the sets $H_i$ and $L_i$, where $H_i = y_i$ is high

$L_i = y_i$ is low or $L_i = y_i$ is in an acceptable range respectively.

A KPI $y_i, i = 1, \ldots, 5$ can be high or low/acceptable and anything in between. The determination of how much $y_i$ is a member of the sets $H_i$ and $L_i$ is represented in the left graph in Figure 4.4.

The fuzzy sets for $x_{jt}$ are easier in their definition, since we define them to be mutually exclusive: $x_{jt}$ can either remain at its current value, increase or decrease. These sets are represented in the right graph in Figure 4.4.
We define the following variables and parameters:

- \( J_{jt} \): The set of all KPIs that are important to decide the change of \( x_{jt} \)
- \( f_{jt} \): A decision variable that weights the magnitudes of the KPIs
- \( \Delta x_{jt} \): \(-1, 0, 1\)
- \( \alpha_i \): a parameter that models how important \( y_i \) is for a decision on \( x_{jt} \)

Using the definitions in the previous table, we say that \( J_{jt} \) is mapped onto the variable \( f_{jt} \) which in turn is mapped on \( \Delta x_{jt} \) according to \( J_{jt} \rightarrow f_{jt} \rightarrow \Delta x_{jt} \). The decision variable \( f_{jt} \) is a weighted sum of truth values that represent whether KPIs are high or low, that means:

\[
    f_{jt} = \sum_{i \in J_{jt}} \alpha_i \times (y_i := L_i) \tag{4.4}
\]

Here, \( y_i := L_i \) represents a statement on how much \( y_i \) is a member of the set \( L_i \). Using the example in the left graph in Figure 4.4 the statement \( y_i := L_i \) returns the value 0.1, while the statement \( y_i := H_i \) would at the same time return the value 0.9.

Implementing a statement like “If \( y_i \) is high, THEN increase \( x_{jt} \)” is straightforward, IF-THEN-statements are basic to any programming language. However a statement such as “If \( y_i \) is fairly high, increase \( x_{jt} \) a little bit” becomes possible to implement using fuzzy sets.

Note the difference between boolean and fuzzy logic: \( y_i := H_i \) can exclusively return 0 and 1 when using boolean logic. When using fuzzy logic, the same statement can return any value between 0 and 1, depending on “how much” \( y_i \) is a member of the set \( H_i \).

The result \( f_{jt} \) in Equation (4.4) is a value on the x-axis of the right graph in Figure 4.4, telling us whether \( x_{jt} \) should be increased, remain the same or decreased.

Figure 4.4: The fuzzy sets for \( y_i \) and the decision variable \( f_{jt} \). The mapping from KPI to PoI is the mapping between the sets in the left graph onto the sets in the right graph.
Chapter 5

Results

We evaluate the algorithms through trial runs on several scenarios.

5.1 Costly Global Optimization Approach

Ultimately, the results of this project are the two algorithms. In this section, we review trial runs on a few scenarios that have been chosen according to their high congestion and aggressive behavior in terms of user movement.

An important difference between the higher dimensional search and the global optimization approach is that the CGO algorithm minimizes a cost function based on network wide AKPIs, while the HDS takes each KPI into account separately. While the HDS-results feature one graph for each KPI, we only show the global cost in the CGO related section of this chapter since specific KPIs have not been used for decision making.

We are considering six scenarios that, from a mathematical point of view, comprise six independent optimization problems. The scenarios differ in terms of congestion$^1$, number of users in the network, frequency reuse and movement patterns.

Note that the algorithm has an initial design phase consisting of five samples that are not part of the iterations. Figures 5.1 – 5.3 illustrate this by showing the distribution of the initial samples on the y-axis at $x = 0$. From 0 up to at most 8 on the x-axis (this was the stopping criterion in case no acceptable value was found) we can see the results of the algorithm iterating and sampling new values.

In scenario 4, 5 and 6, the goal was reached upon initial sampling. These are not presented in graphs, but rather in a table at the end of this section.

Scenario 1 is the scenario that was extensively simulated in order to tune the CGO algorithm. Referring to Figure 5.1, three iterations were needed to reach the goal of a global cost below 0.6. The scenario features a mix of different movements, where about one third of the users are moving on a fast street with 75km/h, one third are roughly stationary - this can be compared to people being in a building, but still moving a little bit - and one third follow a random walk pattern with about 5km/h. This scenario is a good approximation of a crowded downtown environment near a road with heavy traffic.

$^1$Here, congestion means how many users per available frequency there are.

Pavski, 2015.
Scenario 2 is a statistical outlier. It ultimately simulates a giant street that is 3.5km wide with a very high frequency load. Such a highly loaded scenario is extremely unlikely since no street is that big. A street of normal proportions would not be that highly loaded unless there was a traffic jam, in which case the movement would be slower (which means easier to handle from a telecommunications point of view). This scenario is thus mostly of academic interest and puts a roof on what can be done. In Figure 5.2 a minimum cost of 2.23 was reached within eight iterations after initial sampling, which means after 13 function evaluations. Experience from the extensive simulations with scenario 1 suggests that this cost cannot be decreased to the desired level of 0.6 within an acceptable number of iterations.

Scenario 3 is similar to the previous one. A different frequency reuse pattern leads to less interference between cells at the cost of less available radio frequencies. The relative congestion is thus even higher than in Scenario 2. The higher sample values indicate that the interference pattern has a big impact here. Please refer to Figure 5.3.

When running the algorithm on scenarios 4, 5 and 6, the stopping criterion was met during initial sampling and no iterations were performed. Scenario 4 is a version of scenario 1 with less congestion. Scenarios 5 and 6 feature less traffic than the other scenarios, but have different frequency reuse patterns. The achieved values are presented in Table 5.1.

![Figure 5.1: The value of the cost function in the first scenario during the iterations of the CGO algorithm.](image-url)
5.1. Costly Global Optimization Approach

Figure 5.2: The cost values achieved when iterating the algorithm over Scenario 2

Figure 5.3: The cost values achieved when iterating the algorithm over Scenario 3
Chapter 5. Results

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Initial Sampling Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5.10 1.10 0.70 0.40 0.30</td>
</tr>
<tr>
<td>5</td>
<td>1.96 1.39 1.00 0.76 0.25</td>
</tr>
<tr>
<td>6</td>
<td>6.70 0.36 0.35 0.34 0.33</td>
</tr>
</tbody>
</table>

Table 5.1: Initial sampling values for scenarios 4, 5 and 6.

For each of the scenarios presented in Table 5.1, a lower objective value could probably be achieved, but the ones that were found can be considered “good enough”.

5.2 Higher Dimensional Search Approach

As in the previous section, trial runs for six scenarios are presented. These are not necessarily the same scenarios as in the previous section. They differ in terms of frequency reuse, movement pattern and frequency load according to Table 5.2.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Frequency Reuse</th>
<th>Movement Pattern</th>
<th>Frequency Load</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1/3</td>
<td>Street like</td>
<td>20.8%</td>
</tr>
<tr>
<td>2</td>
<td>1/1</td>
<td>Mixed Movement</td>
<td>16.6%</td>
</tr>
<tr>
<td>3</td>
<td>1/1</td>
<td>Street like</td>
<td>20.8%</td>
</tr>
<tr>
<td>4</td>
<td>1/3</td>
<td>Street like</td>
<td>16.6%</td>
</tr>
<tr>
<td>5</td>
<td>1/1</td>
<td>Street like</td>
<td>16.6%</td>
</tr>
<tr>
<td>6</td>
<td>1/3</td>
<td>Mixed Movement</td>
<td>16.6%</td>
</tr>
</tbody>
</table>

Table 5.2: Initial sampling values for scenarios 4, 5 and 6.

The street like movement pattern corresponds to all users in the simulation moving at 75km/h, similar to Scenario 2 in Section 5.1. In the mixed movement pattern, about one third of the users are moving on a fast street with 75km/h, one third are roughly stationary and one third follow a random walk pattern with about 5km/h. The movement pattern is thus the same as in Scenario 1 in Section 5.1. A frequency load of 20.8% corresponds to a highly congested network in a downtown area while 16.6% is a bit less congested and corresponds to a relatively loaded network in a suburban area.

The results in this section are a bit harder to read. For each scenario, a baseline and five iterations of the HDS algorithm were performed and the individual KPIs are plotted. Thus, the x-axis in each subplot can be interpreted as the time axis, while the y-axes represent the amplitude of the KPIs. The KPIs are plotted individually because the algorithm handles them individually. The first subplot in each figure represents the global cost - that is the weighted sum of all KPIs - as well as averages over the cells of the PoI.

To better show even small changes in KPIs, different scales were used in for each KPI. The y-axes are different for each KPI and do not necessarily start at 0.
1. Scenario 1

In Figure 5.4, we note how KPI 3, 4 and 5 each are lowered at the expense of KPI 1 while KPI 2 remains roughly the same. The outlier in the first iteration is remedied by a special rule that allows PoI 3, represented by the blue line in the first subplot, to jump two steps instead of the usual $\Delta x = -1, 0, 1$. This happens in almost every scenario.

The initial set of PoIs was in this approach deliberately chosen to be non optimal in order to see if the HDS algorithm is capable of finding better solutions.

Figure 5.4: The KPIs over the five iterations.
2. Scenario 2

KPI 4 and 5 are decreased at the expense of KPI 1. The reason is that KPI 1 has the lowest relative weight in the objective function as well as in the fuzzy decisions.

Figure 5.5: The KPIs over the five iterations.
3. Scenario 3

As in the previous two scenarios, KPI 3, 4 and 5 decrease significantly at the expense of KPI 2, which has the highest weight in the objective function and the fuzzy decisions. We suppose that the problem lies in the lack of a proper stopping criterion. Note how in the first iteration all KPIs decrease at the expense of KPI 1, which is preferable compared to the outcome at the end of the fifth iteration.

Figure 5.6: The KPIs over the five iterations.
4. Scenario 4

After the outlier in the second iteration, (for which a solution needs to be found in the future) KPI 3, 4 and 5 decrease at the expense of KPI 1. This is desired, however not perfect: Since KPI 2 has the largest weight in the objective function, we would primarily like to decrease it, before decreasing others.

Figure 5.7: The KPIs over the five iterations.
5. Scenario 5
The results from this scenario are included to show a statistical outlier and, ultimately, the most undesirable outcome. The KPIs are in each iteration weighed against each other. In the base line, KPI 2-5 are within good margins, so the algorithm reduces KPI 1. This unfortunately leads to relatively large increases in all the other KPIs. In the second iteration, the algorithm recognizes this and reduces KPI 2-5 at the expense of KPI 1, which is desirable. Lacking a proper memory, however, the algorithm then proceeds to redo the same thing every second iteration. This scenario shows that we have not handled the contradicting objectives of KPI 1 and KPI 2-5 very well.

![Figure 5.8: The KPIs over the five iterations.](image)
6. Scenario 6 KPI 2 - the one with the largest weight - is well within its margin of insignificance\(^2\) and is thus not considered. The same goes for KPI 3. The algorithm reduces KPI 4 and KPI 5 at the expense of KPI 1 during the first two iterations. Noticing that KPI 4 and 5 are very low, the significance of KPI 1 is now high enough to decide to reduce it. This leads to an initial spike in KPI 4 and 5 (and, curiously, 2 and 3 as well), which is however taken care of in the last two iterations. All five KPIs are reduced in the fourth iteration in this scenario, making it the best example of how the HDS algorithm can be used successfully.

Figure 5.9: The KPIs over the five iterations.

\(^2\)KPI 2 is below the parameter \(\beta_2\), defined in Section 2.3.
Chapter 6

Discussion and Future Work

The results and limitations of the algorithms are discussed. Possible ways to develop them are listed.

6.1 Discussion of the Costly Global Optimization approach

The CGO-approach has delivered satisfactory results for highly congested scenarios such as Scenario 1, 2 and 3 in Section 5.1. However, in scenarios with smaller frequency load, less congestion or a lesser cost for another reason, a set of PoIs that result in a cost value below the stopping criterion will be found upon initial sampling. This can for example be seen in the results for Scenario 4, 5 and 6 in Section 5.1. Meanwhile, lowering the stopping criterion – some sampled cost value being below 0.6 – is probably not a possibility to adapt the CGO-approach to less congested scenarios. The cost function was in its current form designed to reflect and ultimately control the behavior of very highly congested scenarios, which means that smaller “bumps” and dependencies disappear due to too strong normalizations. Consider for example KPI 2. In order to normalize it, it is in each measurement divided by $\gamma = 2.5$, which for highly congested scenarios is a typical value that we want to fall short of (we want to reach 0.6 or less). We also consider a KPI 2-measurement below 2 as not significant, since such a value signifies that the majority of the users in the measured area or subset of cells is happy. Similar measures have been taken for the other KPIs.

However, when a set of cells is not very congested, such a definition of a KPI will simply “swallow” a lot of information, since the behavior of for example KPI 2, when it is below 2, is not even considered. In a future iteration of this thesis work, it may thus be interesting to define a “low cost” and a “high cost”, both of which can be computed during the initial sampling phase. Some threshold would then define whether or not the set of cells that has been sampled falls into the very congested field - in which case the current “high cost”-function will be used during the rest of the algorithm - or if a differently formulated cost
function will yield more accurate results.
For now, the global optimization algorithm almost universally disregards the real meaning of the KPIs. However, in reality, sampling of high objective values, which particularly occurs during initial sampling, could put the network in jeopardy with an inappropriate choice of Pols. An approach to evade this may be to define areas of Pols that are forbidden to sample.

There are two areas where the current approach might work, given an appropriate formulation of the cost function:

1. In not very congested areas. The risk of sampling bad KPIs is not particularly high and/or will not affect users too badly. There is simply not a high risk in trying things on a network that is not being used too much. Even if a bad PoI is sampled, not a lot of users will be unhappy.

2. When a new cell is introduced to an existing mature network, and the algorithm chooses the Pols for just that cell. Since the new cell has been introduced to deload the current network, the worst case scenario is that the network is not deloaded during a sample.

The CGO approach is based on the use of radial basis functions in order to interpolate the behavior of the cost. The actual cost is always positive, but the interpolation can at times attain negative values, especially during initial sampling. Experience from the simulations suggests that when choosing $x_{n+1} = x_{\text{min}}$, the point that is most negative is replaced with a sample and the accuracy of the interpolated surface is therefore increased at the point where the surface is least accurate. If the increase in accuracy is low, then $x_{\text{min}}$ will be very close to the minimum of the interpolation surface.

6.2 Discussion of the Higher Dimensional Search approach

The results from this algorithm show it to be a proof of concept more than a working algorithm, which has its reasons in the time limitations of the project. Several simplifications and assumptions have been made in the process; among other things, the algorithm started at a set of Pols deliberately chosen to be far from optimal, and given that the scenario was not too congested, an improvement can be seen in the KPIs. Heavy oscillations in the results between iterations suggest that the model is suboptimal. Too few iterations and the lack of a proper stopping criterion – again, due to time limitations – makes it impossible to make a qualitative assessment. Data mining from live networks and statistical understanding of exhaustive simulations may provide insight in how the model - that is the mapping between KPIs and Pols - can be adapted.

6.3 Future Work

None of the following items have been pursued due to time constraints.
1. While the CGO-approach has been built to work on either all cells in a network or on a subset, the higher dimensional search is built to work on all cells at once. The two algorithms are in their current implementation not compatible and the result of one cannot be used to work with the other. This is not a conceptual problem. In principle, the two algorithms would probably complement each other. It was the basic idea of this project to start with a global search in only four dimensions. The HDS algorithm would then be applied with the solution from the CGO approach as its starting point.

2. Some simple plotting of the KPIs over several ranges of PoIs using the database showed “bumpiness” for certain PoI ranges while smoother behaviour has been noticed for other ranges. This may provide information on whether sampling points can be trusted or not, since PoIs from a bumpy area of the cost function do not provide good input to the interpolation in the CGO-algorithm.

3. Within our project, sampling of every point takes the same amount of time. However, in order to avoid sampling points that promise to have a high cost, an “emergency cut off” could be implemented. The KPIs usually rise fast in the first simulated minute of the simulation and converge to some value after some time. If they rise too far too fast, the cost can be extrapolated in order to still produce a measurement for the interpolation. In the case of the HDS the same procedure could be used to get an evaluation and some information about the neighborhood - and the simulation (or the real trial in the application) can be aborted.

4. Sampling time needs to be extensively researched both in reality and in the simulator. Referring to Tilda Eriksson in [19], a change in KPIs is by far not as evident in reality as it is in the simulator due to high variance and fluctuations between different times of the day and between different days.

   According to Eriksson, a good sample time is one week. Modelling the KPI as a time series - that is a stochastic process - and filtering as much of the fluctuations as possible, sampling over a whole week would give enough confidence. However, modelling different days of the week or even times of the day separately may give the opportunity to reduce sampling time significantly. This may in return reduce the time it takes to find a satisfactory set of PoIs while reducing the risk of jamming the network, since a PoI is only sampled during a very short time.

   Generally, confidence and sampling time trade off against each other. Within the simulator, the sample time we have used is 300 seconds of simulated time, which is the time the KPIs need to converge to a stationary enough point. Obviously, there are no fluctuations being simulated, which is the reason such a short time works, compared to a week in real life. Simulating fluctuations in many different variables in the simulator may be vital to determine a good real life sampling time.

5. While the simulator is a black box in the CGO approach and we cannot treat it with the usual optimization methods which utilize derivative information, the interpolated surrogate surface can be differentiated. Such
a differentiation may yield insight into better ways to choose new sample points. Note that while the surrogate surface is four dimensional, a derivative may still provide some information on where a higher dimensional set of PoIs should be placed. This has to be tested in the simulator.

6. Pareto optimality is a state in which several possibly contradicting goals have reached an equilibrium where no singular goal can be optimized without making another one worse. Since the simulator does not feature dropping of calls, the contradiction of the desires to have as few handovers as possible and to have as few call drops as possible would require Pareto information.

We also have the two contradicting goals of reducing the number of handovers on one hand and reducing the four other KPIs that are included in the cost function on the other hand. This information can be obtained through data mining of live network data or more easily through extensive simulations.

7. The simulator has, in the small scale it was used here, a deterministic output. The KPIs will have to be seen in a stochastic way in a future project, where the probability of a KPI being in an optimal area should be increased. The control theory approach took this into account by running every iteration of a simulation for 10 different seeds, however the optimization approach ignored the stochastic nature of the KPI to reduce simulation time.

8. Vast amounts of live network data are available and can be mined statistically to develop mappings between KPIs and PoIs, leading to further applicable rules. This approach is closer to reality than the simulations that were run in this project, especially considering that the effects of means and variances can actively be included in the process.

9. It is mentioned in Section 6.1 why the current choice of the next sampling point in the CGO approach is effective. We are currently using an interpolation surface that ignores the fact that the cost is non negative. There are no radial basis functions that take non negativity of interpolations into account. It may be interesting to find a way of constraining the linear system of equations that is used to compute the interpolation surface to be positive. However, the system does not have any degrees of freedom. Other ways of interpolating may provide this freedom. At the same time, a different way of choosing new sample points has to be employed.

10. Early in the project, we discussed whether or not it may be possible to treat each cell as an individual optimization problem and to have rules such as a continuous derivative or a truth table for the transition between cells. Based on the results here, this is not possible because each cell would act “selfishly” and try to deload at the cost of overloading other cells. However, such an approach may be valid at least for some of the PoIs or in some special cases. This needs to be investigated.
Chapter 7

Conclusion

The purpose of this project was to investigate whether or not it is possible to automate the choice of handover related PoIs in the GSM network. It was the choice of the thesis workers to attempt two approaches, one rooted in control theory and one in mathematical optimization. Given the constraint of having worked in a simulated environment, the answer is yes, such an automatisation is possible.

An objective function has been formulated and its optimization leads to a viable choice of PoIs. Both the number of KPIs involved in the objective function and the number of PoIs to be chosen can easily be changed or reconfigured while maintaining the current structure of the algorithms.

The optimization approach was implemented using a costly global optimization algorithm. Given a realistic scenario, this approach chooses a suitable set of PoIs within at most eight iterations. The algorithm is capable of finding a close to optimal solution regardless of earlier parameter settings.

The control theory approach was implemented using a mapping between KPIs and PoIs based on experience from field engineers and simulations. While this approach has not been developed to a stage where it works reliably, the results form a proof of concept for an algorithm that can revise a given set of PoIs when there are changes in the radio environment. The remaining work is an engineering problem, not a conceptual one.
Bibliography


