Simultaneous optimization of channel and power allocation for wireless cities

M. R. Tijmes BSc

BT Mobility Research Centre
Complexity Research Group
Adastral Park
Martlesham Heath, Suffolk IP5 3RE
United Kingdom

University of Twente
Department of Electrical Engineering
P.O. Box 217
7500 AE Enschede
The Netherlands

Report Date: May 14, 2008
Period of Work: 04/02/2008 – 16/05/2008
Supervisors: Dr. K. M. Briggs
Dr. ir. A. Meijerink

Abstract

BT is installing wireless access points in several British cities. For these wireless cities the protocol used is IEEE 802.11b, that has 13 partially overlapping frequency channels in the UK and most of Europe. Four different modulations schemes are specified in the 2.4 GHz band with a maximum data rate of 11 Mbps.

This report discusses the problem of simultaneous channel and power allocation and presents an exact algorithm to solve this problem to optimality.

To optimize the user throughput and coverage area it is advantageous to use all available frequence channels to minimize the spectral overlap. Another way to reduce the interference is to minimize the spatial overlap by varying the transmit power of each access point. In this way the coverage area of each access point is regulated and the interference can be minimized.

Optimizing the channel and power allocation is a combinatorial optimization problem with a time complexity that is larger than exponential. The optimization problem is solved with an newly proposed branch-and-bound method, named local best-first search. This is a combination of a depth-first and best-first search. The advantage of using this alternative algorithm is that it has improved pruning and finds the optimal solution faster for this type of problem.

First, the simpler problem of optimizing the channel allocation solely is solved. For all access points the powers are assumed equal in this case. The results show that there is a trade-off between the computational time and channel spacing. A smaller channel spacing results in a better objective value, but requires more time to find the optimal solution. An interesting result is that a channel spacing of 3 performs on average better than a channel spacing of 2.

Next, the problem is extended to joint channel and power optimization, where the transmit power of the access points is also varied. Several objectives are discussed which model the optimization problem. Some introductory results are presented and show that a small number of power levels cause only little improvement in the objective. However, using more rigid power control (e.g. turning access points off) shows larger improvements. The results presented can be used for further research on joint channel and power optimization.

Preface

This report has been written as part of my Master's program in Electrical Engineering that I am pursuing at the University of Twente, The Netherlands. During the two-year during Master phase a compulsory part is an external training which is preferably performed in a foreign country. The aims of this training are to learn how to function in an organization as a young engineer, to practice and enlarge knowledge and skills, and to experience an international work situation.

My specialization is Telecommunication Networks perfectly aligns with the area of expertise of BT. Therefore, I am very pleased that I was able to find a position at BT to perform my external training. During my stay I have learned and obtained new skills which will help me in pursuiing my future career.

I want to thank BT and specifically Keith Briggs who arranged a placement for me at BT. Furthermore, I want to thank Keith Briggs for all useful discussions and his help and inspiration during the prolongation of my placement.

Martlesham Heath, May 14, 2008

Martin Tijmes
m.r.tijmes@alumnus.utwente.nl

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless internet is becoming more and more common in today's world. Still more hand-held devices are available that can access the internet through WLAN and more services are becoming available for these purposes. The main advantage of wireless internet is of course mobility.

In the future we might reach a scenario where wireless internet is available everywhere. At the moment a first step is made by creating 'wireless cities'. In several cities, spread all over the world, a large number of access points are installed to give wireless coverage over a certain area. Some are already commercially deployed, but most of these set-ups are still in an experimental stage and are used for research purposes.

One of the main research questions is how to assign the available frequency channels to the access points and how to select the appropriate transmitter power. Both these aspects are of influence on the coverage and the received interference in the area. Also, the positioning of the access points is subject to some constraints, because of landline access (backhaul) and restricted installation locations, and has to be accounted for.

BT and wireless cities

BT is installing wireless access points (APs) in several British cities. Usually the protocol used for these APs is IEEE 802.11b. These APs can use 11 (US) or 13 (EU/UK) different channels (which partially overlap) and have an upper power limit. IEEE 802.11b is standardized in 1999, operates in the 2.4GHz ISM-band and has a maximum data rate of 11Mbps.

To optimize the user experience (e.g. throughput and coverage) the channels and powers of all APs have to be set in such a way that the received interference is minimized, while retaining a large coverage. This is a NP (non-deterministic polynomial) combinatorial optimization problem, as the number of possible channel and power assignments grows faster than exponentially when the number of access points increases.

Assignment

The goal of this assignment is to provide a reliable optimization program and find to what extent (i.e. size of wireless city and the number of access points) exact optimization is applicable in real-life situations.

This assignment can be split up in two parts: the optimization of the channel allocation of access points and the optimization of the power allocation. At first the powers of all access points are assumed equal, and an optimal solution using channel allocation will be found. Later on the transmit powers of all access points can be varied and power optimization is used to increase the performance even further.

Report outline

This report starts with an introduction on combinatorial optimization (CO) in Chapter 2. Some well known examples are discussed and specifically why the wireless cities problem is combinatorial.

In Chapter 3 we continue the discussion on CO by presenting a branch-and-bound method. This method is an algorithm to solve CO problems to optimality and is discussed in general. The branch-and-bound method is used to solve the wireless cities problem as will be shown in subsequent chapters.

After discussing combinatorial optimization and the branch-and-bound method in general, Chapter 4 relates this to the channel optimization problem. Models are presented to minimize the average and maximum interference.

The results obtained from the models described in Chapter 4 are presented and discussed in Chapter 5.

To reduce the received interference even further, power variation is added to the optimization problem. A new model for joint channel and power allocation is presented in Chapter 6.

Chapter 7 will present introductory results on the joint channel and power allocation problem.

Finally, in Chapter 8 the results for both channel and power allocation are summarized, together with the conclusions. At the end of this chapter are some recommendations are given for further research.

Chapter 2

Combinatorial optimization

2.1 Introduction

Combinatorial optimization (CO) problems are problems where the best solution out of a large but finite number of alternative solutions has to be chosen. Combinatorial refers to the finite number of alternative solutions and optimization refers to the most optimal solution that has to be chosen depending on the objective.

The best solution optimizes the objective of the problem. These objectives are either a minimization or a maximization of an objective function $f(x)$. In case of a minimization problem (most common) the objective can be written as:

$$\min_{x \in S} f(x) \tag{2.1}$$

where x represents an alternative solution in the solution space S and $f(x)$ is the objective function of x. The best x will solve the objective to optimality.

All (alternative) solutions are subject to constraints. All constraints can be wrapped into a boolean function $c(x)$. The function will return *true* if all constraints are satisfied and *false* otherwise. If all constraints are met, a solution is said to be feasible. Infeasible solution are outside the solution space and cannot be used to optimize the objective.

Example

One of the best known CO problems is the traveling salesman problem (TSP). In this problem a salesman has to travel to a finite number of cities, which he all has to visit once, and finally return to the city where he started from. The objective of this problem is to minimize the total traveling distance. An example is given in Figure 2.1, where the weight on the edges represent the traveling distance between cities.

The following can be said about the TSP:

- x: vector containing the cities in the order the salesman has to visit them
- $c(x)$: the constraints are that the salesman has to visit every city once

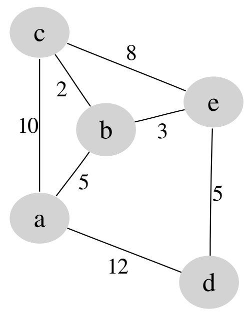


Figure 2.1: Traveling salesman problem

- S: the solution space consists of all routes that satisfy the constraints
- $f(x)$: the objective function gives the total traveling distance for a route x
- $\min f(x)$: the total objective is to find the smallest $f(x)$

In this case x is a vector containing the cities in the order that the salesman has to travel to them. The constraints are that the salesman has to visit every city and each city can only be visited once, and these define the solution space S. The objective function $f(x)$ will give the total traveling distance, depending on the vector x.

2.2 Exact vs. heuristic

An exact algorithm solves a CO problem to optimality, which is the best possible solution, whereas a heuristic method determines a feasible solution that is probably good, but not proved to be optimal.

Often the computational time needed for an exact algorithm is very large, whereas it is small for a heuristic (approximation method). However, for relatively small problems, an exact algorithm can still solve a problem to optimality in reasonable time.

A disadvantage of a heuristic is that you don't know exactly how good the solution produced is, compared with the best solution possible.

2.3 Time complexity

The complexity of a CO problem can be defined as the number of elementary operations in an algorithm. The term complexity is in this case related to the computational time and specifies the time to solve a certain problem with the algorithm. The computational time is normally given as an absolute value, whereas the time complexity is often denoted as a function of the size of the problem:

$$O(n^2) \tag{2.2}$$

The statement above means that the time complexity is in the order of n^2, where n is the size of the problem.

Class P vs. NP

There are mainly two classes of CO problems. The first class of problems consist of those where the time complexity can be bounded from above by a polynomial. Hence the class P (polynomial time).

The other class of problems consist of problems which are very unlikely to be solved in polynomial time. Hence the class NP (non-deterministic polynomial time).

For this type of problems the time complexity explodes with an increasing size of the problem (e.g. $O(2^n)$). For the traveling salesman problem the number of feasible solutions equals $(n-1)!$, with n the number of cities, and thus grows faster than exponentially.

2.4 Wireless cities

The use of combinatorial optimization techniques are applicable for wireless city environments. I.e. an area inside a city that has wireless coverage for access to the internet.

The problem of optimizing the channel and power allocation of access points (APs) in wireless cities is a NP CO problem, as mentioned in Chapter 1. This means that the time complexity grows exponentially with the size of the problem (i.e. number of access points) that needs to be solved.

The use of a heuristic method will give a reasonable solution in little time. However, the problem with heuristics is that we don't know how good or bad this reasonable solution is. With the use of tricks a problem of fair size can still be solved to optimality with an exact algorithm in reasonable time.

The branch-and-bound method is such an exact algorithm which is presented in Chapter 3 in a general way.

Chapter 3

Branch-and-bound

The branch and bound method is a widely used algorithm to solve problems to optimality and with that find the best solution possible. Branch-and-bound is a so-called enumerative method, because it displays all possible solutions in a tree-like structure. The method differs from complete enumeration by discarding large parts of the tree from evaluation. This is called pruning and makes branch-and-bound a powerful method.

3.1 Introduction

The branch-and-bound tree consists of several layers, called depths. In case of the traveling salesman problem (TSP), the number of depths is equal to the number of cities that have to be visited and in the case of wireless cities equal to the number of wireless access points (APs). The tree always starts in the root node at depth 0. From this node an iterative process of branching, bounding and pruning starts. In Figure 3.1 an example is given for the TSP, where 3 cities have to be visited, and the wireless cities problem, where 3 APs have to be assigned to 1 of 2 different channels.

The total tree consists of a large number of nodes and edges. The nodes represent a partial solution, except for leaf nodes (at the bottom) which represent complete solutions. In case of the TSP the nodes represent the (partial) in-order vector of cities to be visited or in the case of wireless cities the (partial) assignment of channels to APs. The edges connect the nodes and form paths from the root node to the leaf nodes. Each path in the tree represents a different complete solution.

The tree is traversed in a process of branching, bounding and pruning. Each of these processes are discussed next.

3.1.1 Branching

Branching is the creation of several edges originating from a single node, resulting in new nodes at the next depth. The number of edges originating from each node is defined by the number of possible assignments to the following node. In the case of the

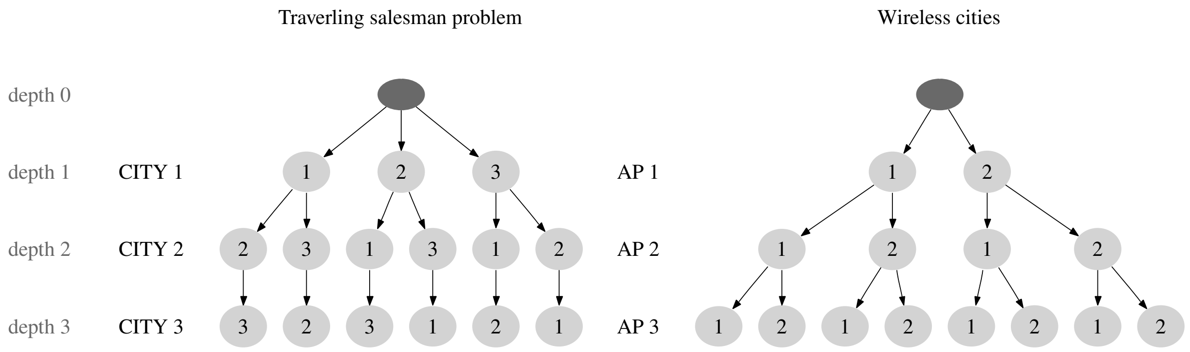


Figure 3.1: Branch-and-bound tree for the traveling salesman problem (TSP) and the wireless cities problem

TSP the number of edges is equal to the number of cities that have not yet been visited and the case of wireless cities the number of possible channels.

Each edge represents an assignment of a possible choice to the following node. In the case of the TSP each edge represents a city and in the case of wireless cities each edge represents a certain channel. The total number of edges emerging is equal to the number of cities that have not yet been assigned or visited, or the number of possible channel allocations.

3.1.2 Bounding

In the branch-and-bound method there are two bounds that are important for pruning: the lower bound α and upper bound ω. The calculation of these bounds is called bounding.

α is computed at every node and represents the evaluation of a partial solution. Every node that emerges below this node will have an α that is greater or equal than the previously calculated lower bound.

ω is a global bound and represents the best complete solution that is known so far in the whole tree and can only be found at leaf nodes.

3.1.3 Pruning

The computation of the lower and upper bounds (i.e. bounding) give us powerful tools for pruning. Pruning discards parts of the tree which can only result in a complete solution that will be worse that the one already found.

The case for which this is true, is when the lower bound α is larger than the upper bound ω (i.e. $\alpha > \omega$). As α can never get smaller, when traversing down the tree, there can never emerge a solution that is better than ω. Thus, if for any node α is larger than ω, the node and all its children (nodes on subsequent depths) can be discarded.

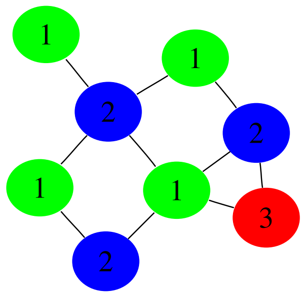


Figure 3.2: Optimized graph coloring: at least 3 different colors are needed to color this graph

Nodes are often also discarded when α is equal to ω. This means that the best solution that can be found traversing down the tree will at best be equal to the current best solution. For matters of computational time, we have no interest in equal solutions.

Example 3.1

Another combinatorial optimization (CO) problem that can be solved with the branch-and-bound method is the graph-coloring problem. In this problem a color has to be assigned to every node and the objective is to minimize to the number of different colors that are used. The constraint in this problem is that two nodes connected by an edge, cannot be assigned the same color. An optimized solution for 8 nodes is given in Figure 3.2.

In the example shown in Figure 3.2 there are 8 nodes (n) that need to be colored. The maximum number of different colors (max) that can be used is equal to the number of nodes, i.e. also 8. After optimizing, the figure reveals that the minimum number of colors that need to be used is 3.

The optimization process is performed with the branch-and-bound algorithm called MINIMIZE(), given in Algorithm 1. This algorithm describes a recursive depth-first search (see Section 3.2.1) in pseudo-code, which traverses a tree and evaluates the nodes in a systematic way.

3.2 Methods

There are several versions of the branch-and-bound method. Each of them has its advantages for different types of problems. The difference between these methods is mainly based on what branch to follow up on or what node to continue with, i.e. the order in which the nodes in the tree are evaluated.

Traversing a tree in different ways and evaluating nodes in a different order will influence the speed of the optimization process, as the pruning process may differ for

Algorithm 1 MINIMIZE()

Parameters:

Integer $depth \geqslant 0$, initialized to 0
Integer n, equal to the number of nodes
Vector x of length n, containing the assigned node colors
Vector z of length n, remembering the x that gave the best $f(x)$ so far
Function $c(x)$, checking if all constraints are satisfied by x
Function $\alpha(x)$, representing a lower bound for $f(x)$
Real ω, representing the upper bound, initialized to infinity
Integer max, the number of different colors available

$depth++$
for $i = 1$ to max **do**
    $x[depth - 1] = i$
    if $c(x)$ & $(\alpha(x) < \omega)$ **then**
        if $depth == n$ **then**
            $\omega = \alpha(x)$
            $z \leftarrow x$
        else
            MINIMIZE()
        end if
    end if
end for
$depth--$
return

each method. Deciding which method to use depends on the type of problem. Several variants and guidelines are discussed next.

3.2.1 Depth-first search

The depth-first search (dfs) traverses down the tree to a leaf node as quickly as possible, starting from the root node. Once a leaf node is reached, the path is backtracked and the first possibility to traverse down again is taken to find the next leaf node. The graph structure is shown in Figure 3.3a.

As the dfs quickly reaches a leaf node, it quickly finds a feasible solution. This solution gives a first indication of the objective value and gives an upper bound ω useful for pruning. A dfs algorithm is easy to implement in a recursive program.

3.2.2 Breadth-first search

In a breadth-first search (bfs) all nodes are evaluated depth-wise. This means that first all nodes at depth 1 are evaluated and then all nodes at depth 2, unregarded of the parent nodes. The graph structure is shown in Figure 3.3b.

The bfs evaluates the tree in a structured manner, pruning the parts of the tree which do not satisfy the constraints or have lower bounds α larger than the upper bound ω. However, it takes a long time to reach a leaf node and find a first complete solution.

3.2.3 Best-first search

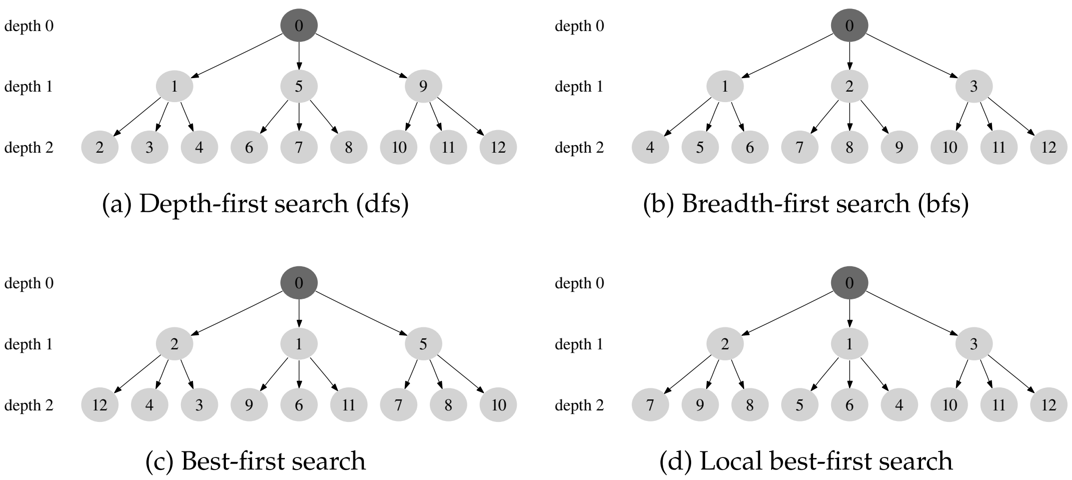
In a best-first search the tree is traversed, starting each time from the best node known. In other words, a heuristic is used to decide the next node to continue with. This heuristic may help finding the optimal solution quicker, but this will not be true for every case.

When a node is branched on, for all children the lower bound α is determined. The next node to branch on, is the node with the lowest lower bound α in the whole tree. The search therefore always continues with the best possible node. The graph structure is shown in Figure 3.3c.

The best-first search finds the best possible solution as fast as possible, but requires a lot of state information to find the next best node to continue with, which can be a burden on the computational time.

3.2.4 Local best-first search

The local best-first search is a combination of depth-first and best-first search. In this method the tree is traversed downwards until a leaf node is reached, as in a dfs. However, at each depth the best node is chosen to continue with, depending on each child's lower bound α, as with a best-first search. However, note that the next node is chosen



(a) Depth-first search (dfs)

(b) Breadth-first search (bfs)

(c) Best-first search

(d) Local best-first search

Figure 3.3: Tree structures for several variants of the branch-and-bound method

locally and not global. When a leaf node is reached, the path is backtracked and for the first parent node with unevaluated children encountered, the best child is chosen to continue with. The graph structure is shown in Figure 3.3d.

As the dfs, the local best-first search also finds a feasible solution quickly. However, by choosing the best node to continue with, the best solution will be found quicker.

Chapter 4

Channel optimization - introduction

With the methods and tools discussed in Chapter 2 and 3 the channel optimization problem can be analyzed and modeled. This chapter gives a problem description together with a branch-and-bound model.

4.1 Problem description

Channel optimization is optimizing the channel allocation for access points to maximize the throughput experienced by a user.

The scenario considered is a urban area with a number of access points up to 35. The protocol used is IEEE 802.11b, which can use 13 channels in the UK and most parts of Europe, in the 2.4 GHz range. The center frequencies are spaced 5 MHz apart, but adjacent channels still have considerable overlap in the frequency spectrum, see Figure 4.1. The bandwidth of each channel is considered to be 20 MHz, but spreads outs over a wider range in reality. The exact overlapping factors are given in Table 4.1. All access points (APs) are considered to use equal transmit power.

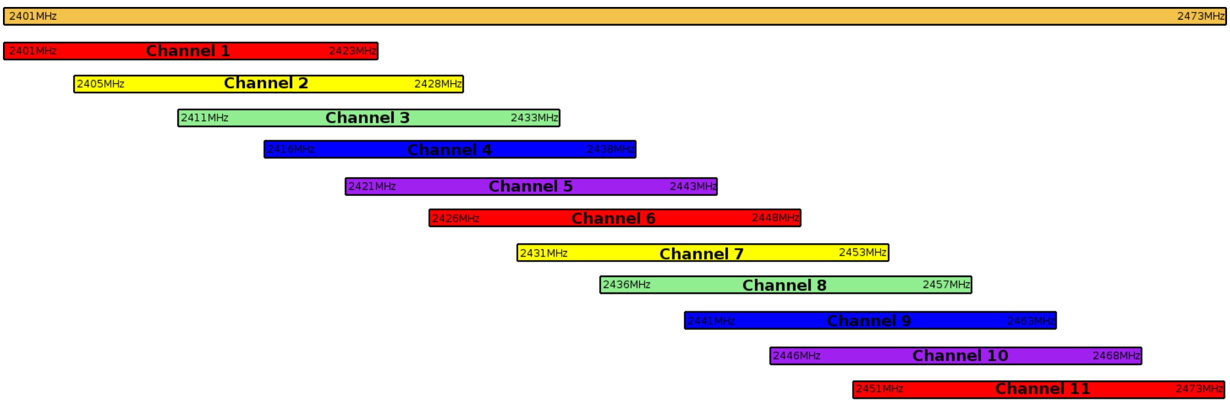


Figure 4.1: Spectral channel overlap in IEEE 802.11b (US channels)

spacing	*overlap factor*	*spacing*	*overlap factor*
0	1.00	7	$0.54 \cdot 10^{-4}$
1	0.73	8	$0.18 \cdot 10^{-4}$
2	0.27	9	$0.79 \cdot 10^{-5}$
3	$0.37 \cdot 10^{-1}$	10	$0.32 \cdot 10^{-5}$
4	$0.54 \cdot 10^{-2}$	11	$0.18 \cdot 10^{-5}$
5	$0.84 \cdot 10^{-3}$	12	...
6	$0.18 \cdot 10^{-3}$	13	...

Table 4.1: Overlapping factors in IEEE 802.11b, based on the channel spacing [4]

The outdoor range in open field (including 1 wall) is about 140 meters, but for an urban area the range would typically decrease to 50 or 100 meters, depending on the surroundings. Therefore, the mean separation between the access points should be in the order of this range, to obtain a coverage area that is as large as possible and results in the least amount of interference. For simulation purposes a mean separation of 50 meters is assumed.

For practical purposes the positions of the APs will come from real-life situations, but for simulation purposes the AP locations are considered to be placed in a quasi-random way such that the coverage area will be as large as possible. Quasi-random number generators are discussed in Appendix A.

4.1.1 Model description

The optimization criterion that is used in the proposed model is based on the received interference at any access point. This results in a simplified objective compared to the actual objective of maximizing the user throughput. This simplification will speed up the algorithm.

Instead of computing the interference at every possible location, the interference is approximated by a single measurement at the AP for its coverage area. The approximation error will be small for locations situated nearby the AP, because the relative distance from these locations to the AP is very small, compared with the distance to the other APs, and will thus be the same. For locations near the border of the coverage area of an AP, the approximation will be less accurate.

Furthermore, the model only considers the interference received from other APs and does not take noise into account. Noise is assumed to be additive and fading effects can be introduced by using Rayleigh or Rician model.

4.1.2 Power and interference

During the simulations of the branch-and-bound algorithm the following parameters are used:

- Transmit power P_t: 100 mW (20 dBm)
- Path loss exponent n: 2.86 (urban area)
- Reference path loss $PL(d_0)$: 40.2 dB (at d_0=1m)

The reference path loss can be calculated with Friis free space equation (considering unity gain) [7]:

$$P_r(d_0) = \frac{P_t \lambda^2}{(4\pi)^2 d_0^2} \tag{4.1}$$

where P_r is the received power at reference distance d_0, P_t the transmit power and λ the wavelength.

With the log-distance path loss model, the total path loss can be determined in the following way [7]:

$$PL(d)[dB] = PL(d_0) + 10n \log_{10} \left(\frac{d}{d_0} \right) \tag{4.2}$$

where $PL(d)$ is the total path loss over distance d, $PL(d_0)$ the path loss at reference distance d_0 and n the path loss exponent.

Finally, the received power can be found after subtracting the path loss from the transmit power [7]:

$$P_r[dBm] = P_t[dBm] - PL[dB] \tag{4.3}$$

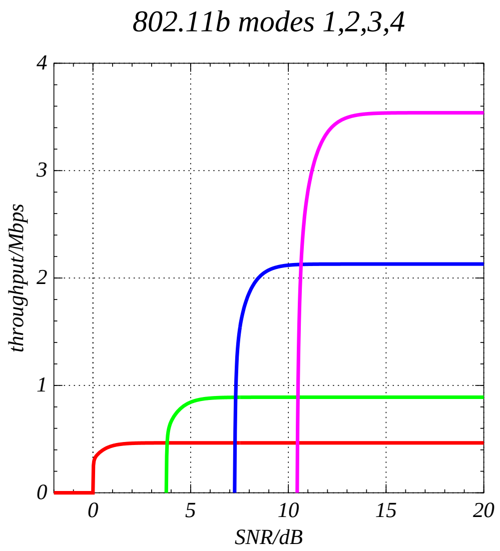
4.1.3 Modulation schemes

In IEEE 802.11b there are four different modulation modes specified, see Figure 4.2. Each modulation scheme requires a different minimal SNR (signal-to-noise ratio). A better SNR allows switching to a higher modulation mode, which increases the throughput. The modulation modes can be used to convert the SNR to actual throughput to obtain coverage areas with specific throughputs.

4.2 Objectives

There are several criteria by which the channel allocation problem can be optimized, the so-called objective functions. In any case the total objective has to be minimized, because all objective functions consider the interference that is received at each access point (AP). There are two main objective functions $f(x)$ that are of interest:

- *avg*: minimize the average interference that is experienced at each AP
- *max*: minimize the maximum interference that is experienced at each AP



Mode	*Saturated throughput*
1	0.45 Mbps
2	0.90 Mbps
3	2.10 Mbps
4	3.55 Mbps

Figure 4.2: Throughput for 4 different modulation schemes in IEEE 802.11b [3]

The optimization problem is given as (see Section 2.1):

$$\min_{x \in S} f(x) \tag{2.1}$$

The total interference at AP$_i$ is given by:

$$I_i(x) = \sum_{j \neq i} I_{i,j}(x) \tag{4.4}$$

where $I_{i,j}(x)$ is the interference at AP$_i$ caused by AP$_j$, as a function of the allocated channel vector x.

4.2.1 Average interference

The *avg* objective is a measure of the average interference that is received at every AP. The average interference per AP can be obtained by dividing the total interference received by all APs by the number of APs (n). The *avg* objective function is:

$$f(x) = \frac{1}{n} \sum_i I_i(x) \tag{4.5}$$

4.2.2 Maximum interference

The *max* objective gives an explicit upper bound for the maximum interference that is received by any AP. The *max* objective function is:

$$f(x) = \max_i I_i(x) \tag{4.6}$$

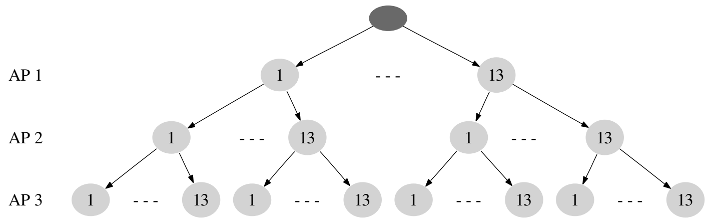


Figure 4.3: Branch-and-bound tree: channel optimization

4.3 Branch-and-bound model

With the objective defined in Section 4.2, a branch-and-bound model can be constructed. In this model the number of depths in the branch-and-bound tree will be equal to the number of APs, instead of the number of cities in the TSP. Furthermore, at each depth a certain channel is assigned to an AP. Therefore, the number of branches emerging from each AP will be equal to the number of channels. Figure 4.3 shows a tree with three APs and 13 channels.

For the channel optimization problem there are no constraints, because each channel can be assigned to every node and every channel can be used multiple times or none. Therefore, the solutions space S in Equation 2.1 consists of all possible combinational channel assignments x, where x is a vector with the channel allocations for each AP.

Summarizing:

- x: vector containing the channels that are assigned to the APs
- $c(x)$: there are no constraints for the channel optimization problem, thus $c(x)$ will always return *true*
- S: the solution space consists of all possible combinations of channel assignments to the APs
- *avg*: this objective function $f(x)$ determines the average interference at a single AP, given a certain $x \in S$
- *max*: this objective function $f(x)$ determines the maximum interference at a single AP, given a certain $x \in S$
- $\min f(x)$: the total objective is to find the $x \in S$ that minimizes $f(x)$
- *time complexity*: the computational time required is $O(c^n)$, with n the number of APs and c the number of channels (13 in most of Europe and the UK)

Chapter 5

Channel optimization - implementation and results

The models discussed in Chapter 4 are used to optimize the channel allocation over all APs, such that the interference received at any AP is minimized. To speed up the simulations several programming tricks are implemented and discussed next in Section 5.1. The simulation results, which show the time dependencies and the achievable interference levels are presented in Section 5.2.

5.1 Implementation

As the size of the channel optimization problem (i.e. the number of nodes n) and its time complexity grows very fast ($O(13^n)$), the branch-and-bound algorithm must be very efficient to compute. Therefore, the algorithms are implemented in the C programming language. C is one of the fastest programming languages, because it is very low-level. This results in little overhead and allows you to control every step in the algorithm.

5.1.1 Branch-and-bound: local best-first search

The branch-and-bound variant used for implementation of the channel optimization problem is the local best-first search, as discussed in Section 3.2.4. This method allows for quick pruning and converges quickly to the optimal solution. Experiments showed that this method has superior pruning for this specific problem compared to the other methods that were discussed in Chapter 3. The pseudo-code for the local best-first search method which minimizes the interference is given in Algorithm 2.

5.1.2 Programming tricks

The channel allocation problem belongs to the NP class and even though the algorithms are implemented in C, the computation still takes a considerable amount of time. There-

Algorithm 2 MINIMIZE()

Parameters:

Integer $depth \geqslant 0$, initialized to 0
Integer n, equal to the number of access points
Vector x of length n, containing the channel assignment
Vector z of length n, remembering the x that gave the best $f(x)$ so far
Integer $num\_channels$, containing the available number of frequency channels
Local matrix $child$ of size $[n \text{ x } 2]$, containing $\alpha(x)$ and channel for evaulated child nodes
Function $\alpha(x)$, representing a lower bound for $f(x)$
Real ω, representing the upper bound, initialized to infinity

```
depth++
at_leaf = false
m = 0
for j = 1 to num_channels do
    if depth == 1 and 2j > num_channels then
        break
    end if
    x[depth − 1] = j
    if α(x) < ω then
        child[m][0] = α(x)
        child[m][1] = x[depth − 1]
        m++
        if depth == n then
            at_leaf = true
            ω = α(x)
            z ← x
        end if
    end if
end for
if !at_leaf then
    permute rows of child so that column 0 is decreasing
    for row in child do
        x[depth − 1] = row[1]
        MINIMIZE()
    end for
end if
depth−−
return
```

	Channel	
	avg	*max*
Complementary solutions	x	x
Channel spacing	x	x
Pre-ordering	x	x
Initial random solution	x	x
Incremental objectives	x	
Symmetric AP distance matrix	x	x

Table 5.1: Programming tricks

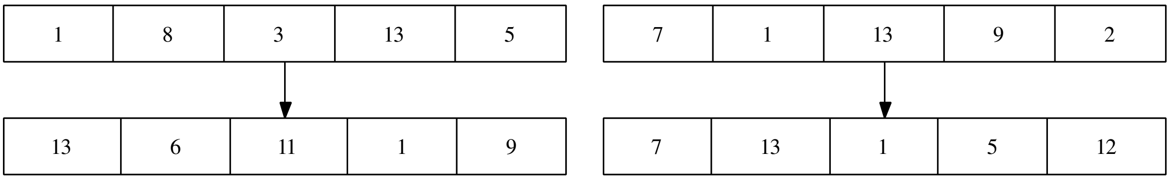


Figure 5.1: Complementary channel assignment schemes: 2 examples

fore, a number of smart tricks are implemented in the algorithm to reduce the computational time even further. All tricks are listed in Table 5.1 and are discussed next.

Complementary solutions

All possible solutions for this optimization problem have a complementary solution. Since the first channel has the same properties as the last channel, the second channel has the same properties as the second-last channel and so on, it makes no difference whether the channel assignment start with the first or the last channel. Therefore, these solutions will be interchangeable. The solutions will give the same objective value and therefore half of the solutions in the tree can be discarded.

In Figure 5.1 two pairs of complementary channel assignment are given. Each pair of channel allocation schemes will result in the same objective value and thus the same amount of interference.

By allowing the first AP only to be assigned the first half of all possible channels, half of the branch-and-bound tree is discarded. The part of the tree that is discarded exactly contains the complementary solutions. All other APs are still allowed to use all possible channels.

Channel spacing

In practical cases not every available channel is used, but for example a channel spacing of 3 is implemented. This means that only channels 1, 4, 7, 10 and 13 can be assigned

Channel spacing	*Channel numbers*	*Total*
1	1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10 - 11 - 12 - 13	13
2	1 - 3 - 5 - 7 - 9 - 11 - 13	7
3	1 - 4 - 7 - 10 - 13	5
4	1 - 5 - 9 - 13	4
5	1 - 6 - 11	3
6	1 - 7 - 13	3

Table 5.2: Channel spacing and their respective channels

to APs. As a larger separation in channels inherently decreases the spectral overlap between channels, these implementations are very useful for scenarios with only a few APs. Figure 4.1 shows that for a channel spacing of 5 or 6, all channels can be considered non-overlapping. An overview for the different channel spacings is given in Table 5.2.

By increasing the channel spacing and decreasing the number of channels, the time complexity reduces. Recall that the time complexity of this problem is given as $O(13^n)$ with its base-number equal to the number of channels. By reducing the number of channels, the time complexity decreases and the optimization performance increases.

Pre-ordering

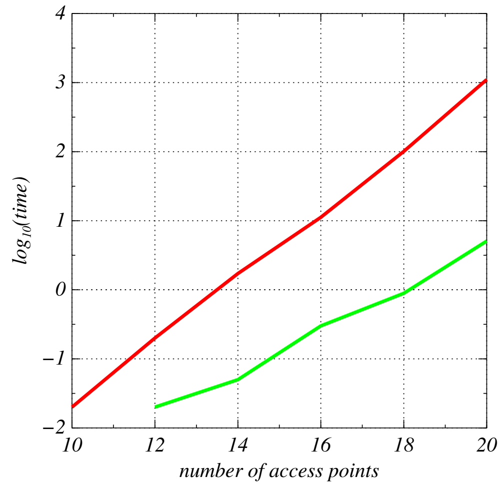
APs that are critical in the optimization process are those that are surrounded by a large number of APs in their neighborhood. These APs have a large chance of receiving much interference and are therefore more important to be assigned to the best possible channels.

These critical APs have a large likelihood to be located in the center of the area. Therefore, by ordering the APs by their distance to the center of mass (of all APs), it is likely that the critical APs are assigned a channel in one of the first steps of the optimization process and will therefore have more freedom in the channel assignment in the top of the tree.

To see the influence of pre-ordering on the computational time needed for an optimization process, an experiment has been performed. In Figure 5.2 the results are shown, together with the configuration used for the experiment. Note that at this stage there is no relation in time between pre-ordering and no pre-ordering that can be extended to general configuration settings. This experiment merely shows that there is an advantage in pre-ordering.

Initial random solution

Starting with a random solution gives an upper bound for the optimization algorithm. In the branch-and-bound method this upper bound is important and useful for pruning (discarding parts of the tree).



Parameters:	
Objective	*max*
Transmit power	20 dBm
Path loss exponent	2.86
Reference path loss (1m)	40.2 dB
Mean separation	75 m
Channel spacing	2
Quasi-random generator	Niederreiter

Figure 5.2: The computational time needed to solve the channel allocation problem to optimality when using pre-ordering and no pre-ordering

By running a number of quick random solutions, a fairly good upper bound can be obtained to start the algorithm with. This improves the pruning process and speeds up the optimization algorithm.

Incremental objectives

When evaluating a possible solution (or node in the tree), the (partial) objective is computed. For every AP the received interference has to be calculated, which is an exhausting process. By remembering the amount of interference in a previous stage and only calculating the interference caused by the newly assigned AP, a lot of time is saved.

This trick can only be applied to the *avg* objective, for which it is only necessary to remember the total amount of interference over all APs. For the *max* objective, the interference for every single AP has to be remembered, which requires too much state information and reduces the speed of the algorithm and declines the effect of this trick.

Symmetric AP distance matrix

The amount of interference received by an AP depends on the distance from the interfering source. These distances are stored in a matrix, together with the resulting interference, which is computed before running the algorithm. Because the distance from AP 1 to AP 2 is the same as the difference between AP 2 and AP 1, the matrix is symmetrical.

Therefore, when computing the objective value only half of the matrix has to be evaluated and the resulting interference can be doubled. This reduces the computational time of the objective function by half and can be applied to all objectives.

Objectives	*avg*, *max*
Number of APs	10 ... 35
Channel spacing	1, 2, 3, 4, 5, 6
Number of channels	13
Transmit power	20 dBm
Path loss exponent	2.86
Reference path loss (1m)	40.2 dB
Mean separation	50 m
Random runs	100
Quasi-random generator	Niederreiter

Table 5.3: Simulation parameters

5.2 Simulation results

With the proposed local best-first search algorithm a large number of simulations were performed. The *avg* and *max* objectives are used for determining an optimal channel allocation scheme. For both objectives it is of interest what the minimum level of interference is that can be reached, as well as the computational time to solve the problem to optimality. The parameters used for the simulations are summarized in Table 5.3.

The simulations are performed with varying numbers of APs, in a range from 10 to 35. The locations of the APs are generated quasi-randomly, to distribute them in a random, but uniform way. More information about quasi-random generators and the used Niederreiter algorithm can be found in Appendix A.

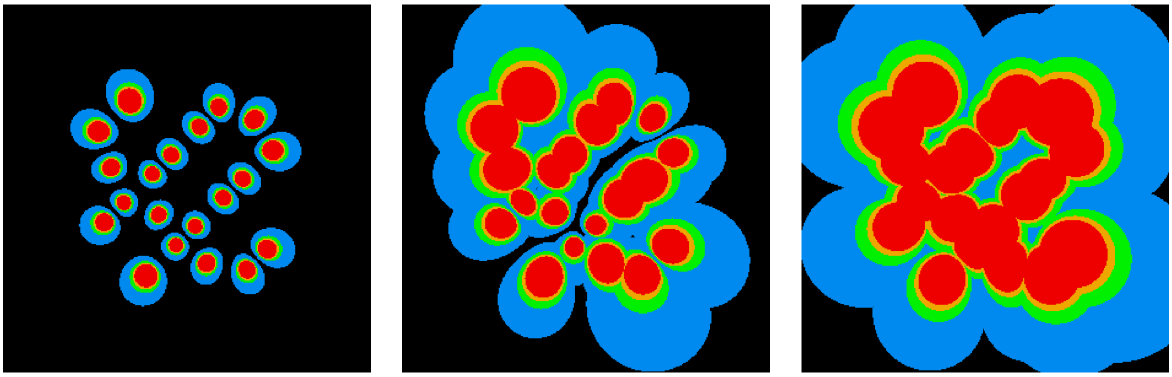
To determine the decline in minimum interference and computational time when increasing the channel spacing, this parameter is also varied. The channel spacings range from 1 to 6, which results in different amount of channels available for allocation, see Table 5.2.

Before each simulation a large number of random channel allocation schemes are generated to start with a good upper bound ω. A sharp value for ω improves the pruning process and reduces the computational time considerably.

An example of how channel optimization can improve the user throughput and coverage area is given in Section 5.2.1. Note that the throughput is not linear related to the objectives of minimizing the interference and is merely illustrative. Section 5.2.2 and 5.2.3 discuss the simulation results using channel optimization. These sections give the dependencies of the received interference and computational time on the number of APs and the channel spacing. Extra graphs are presented in Appendix B for reference.

5.2.1 User throughput and coverage area

By minimizing the received interference at each AP, the goal is to optimize the user throughput and coverage area. These two objectives are not linear related, but a sim-



(a) 20 APs using the same power level and channel

(b) 20 APs with randomly assigned channels

(c) 20 APs using the same power level, but with an optimized channel allocation (13 channels)

Figure 5.3: Comparison in throughput and coverage using a single channel, a random assignment or channel optimization with 13 channels, showing mode 1, mode 2, mode 3, mode 4 and no throughput. The modes are defined in Section 4.1.3

plification is necessary to have a feasible time complexity. Figure 5.3 shows an example where the throughput and coverage area are largely improved by optimizing the channel allocation. After optimization, the throughput at every position is determined by mapping the signal-to-interference ration (SIR) to a modulation mode, see Figure 4.2.

The left figure is obtained by using the same channel for all APs, the middle figure is a result of a random channel assignment with 13 channels and the right figure is obtained by allocating an optimal scheme using all 13 channels.

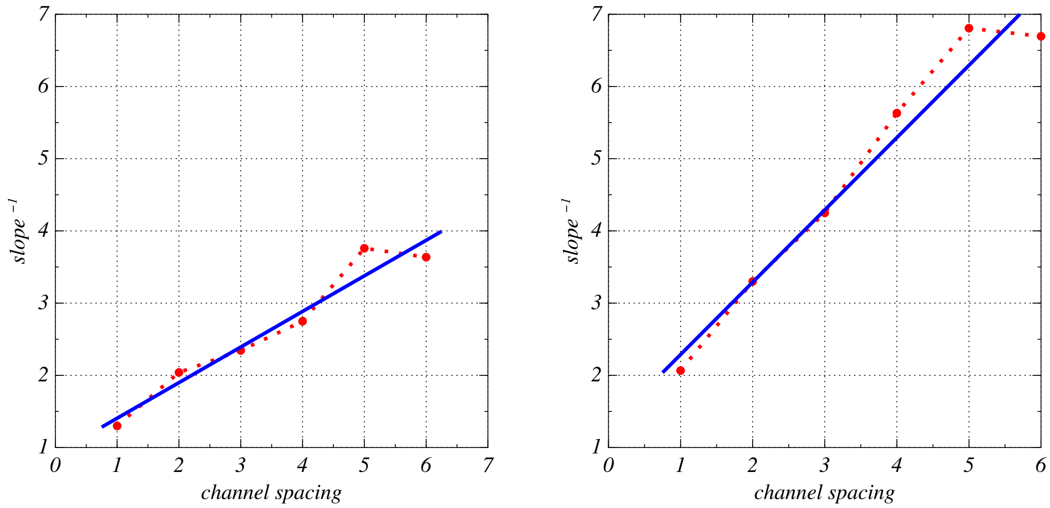
5.2.2 Computational time

The computational time is the time needed for the algorithm to find an optimal solution to the channel allocation problem. The time resolution for the simulations is equal to 10 ms and is measured in seconds. This gives a bad accuracy for a total number of APs around 10, but is sufficient for larger numbers of APs.

Figure B.1 and B.3 in Appendix B shows that the time depends logarithmic on the number of APs. The minimum computational time is given by the time resolution of 10 ms ($\log_{10}(0.01) = -2$).

The time dependency differs for the *avg* and *max* objective and for different values of channel spacing. The dependencies can be represented by the steepness of the curves (linear slope on a logarithmic scale). Figure 5.4a and 5.4b show the relation between the slopes and various values of channel spacing.

The graphs show that there is a linear relation for the inverse of the slope ($slope^{-1}$). A higher value for the inverse slope indicates that computational time increases slower for larger number of APs.



(a) Avg: all spacings

(b) Max: all spacings

Figure 5.4: Relation between *slope*$^{-1}$ and channel spacing, where *slope* is the slope of the computational time versus the number of access points on a logarithmic scale. See also Figure B.1 and B.3

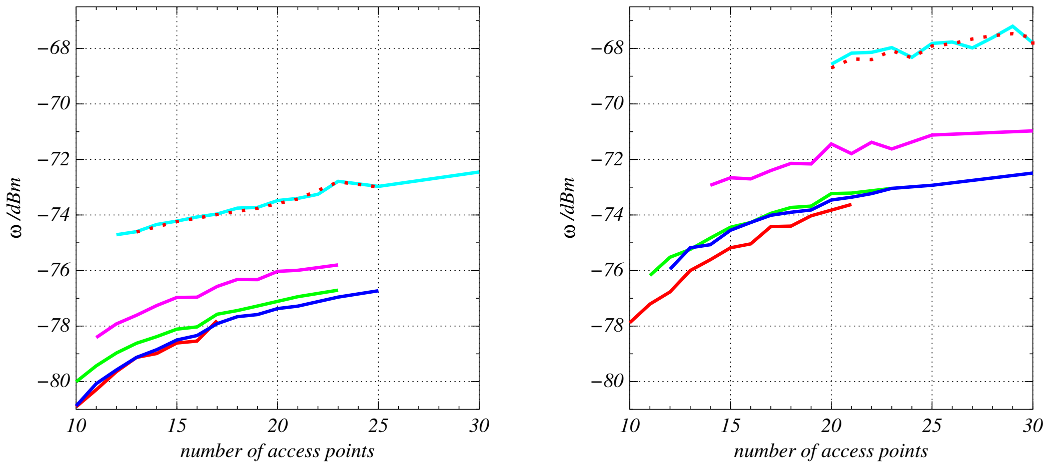
For channel spacing 1 the inverse slope is already larger for the *max* objective than for *avg*, as shown in Figure 5.4, and also increases quicker. This shows that for larger number of APs the computational time increases at a slower rate for the *max* objective, than for the *avg* objective. Note that the performance of channel spacing 5 and 6 is almost equal as they have the same total number of channels available.

Additionally, simulations showed that for different values of the mean separation the computational time is equal. This indicates that the required time to solve a problem to optimality mainly depends on the ratios of individual distances between the APs and not on absolute distances. Appendix A shows that there is a minor, but negligible, influence of the the distribution of the AP locations on the computational time.

5.2.3 Interference

In Figure 5.5a the average interference ω at a single AP is given as a function of the total number of APs, for different channel spacings. The average interference is calculated as the total sum of received interference at all APs divided by the number of APs, as explained in Section 4.2. The maximum interference in Figure 5.5b shows the maximum received interference ω at any AP.

The absolute value of the received interference depends on the mean separation between a pair of closest APs. The interference graphs will shift up (receive more interference) when the mean separation reduces and shift down when the mean separation



(a) Average interference (*avg*) at a single AP, for different channel spacings

(b) Maximum interference (*max*) at a single AP, for different channel spacings

Figure 5.5: Received interference at a single AP, when optimizing the *sum* or *max* objective, for channel spacing 1, spacing 2, spacing 3, spacing 4, spacing 5 and spacing 6 (dotted)

increases. The shape of the graph will be retained.

The received interference ω for all spacings grow monotonically. I.e. more APs introduce more received interference at all other APs. Firstly, it is important to notice that for both the *avg* and *max* objective, the values of ω for spacings 1, 2 and 3 are close together. Secondly, it is even more important to note that for the *avg* objective channel spacing 3 outperform es spacing 2, even though there are less channels available. This also holds for the *max* objective, but the performance is now more comparable with channel spacing 2.

The assigned channel spacings results in a number of available channels for the optimization process. Inherently to the channel spacing size, the available channels for spacing 2 overlap more with each other than the available channels for spacing 3. This can explain the superior performance of channel spacing 3 over spacing 2, even though spacing 2 has more channels available than spacing 3.

Also note that channel spacings 5 and 6 perform equally well. Therefore, it has no apparent disadvantage to use a channel spacing of 5 instead of 6 when 12 different channels are available. A channel spacing of 6 would reduce to 2 available channels for a total of only 12 channels.

Chapter 6

Joint channel and power optimization - introduction

In previous chapters an algorithm has been proposed to optimize the channel allocation in a 'wireless city' problem. This chapter will extent the problem by allowing variations in the transmitter power.

First a problem description will be given, after which the objectives and the branch-and-bound model are discussed.

6.1 Problem description

Changing the transmitting power of an AP, changes its coverage area. As APs are normally not spaced evenly and uniform over an area, it may be advantageous to change the coverage area for each AP, to minimize overlap.

Reducing the overlap of the coverage areas of the APs, reduces the interference at any point in the total area. A reduction in interference is desirable, because this is very likely to increase the SIR inside the coverage area of each AP. The disadvantage of reducing the coverage area of each AP is that the total area that is covered by all APs will also be reduced. This problem can be partially countered using different channels reducing the interference, as discussed in Chapter 4 and 5. In this way overlap of coverage areas will not result in interference and the total coverage area can be maximized.

In certain configurations (i.e. sets of AP locations) it can even be desirable to switch off APs by reducing their transmitting power to zero. As the positioning of the APs is most of the time not optimal, one or more APs can be a bottleneck to the optimization process and omitting them can give better results.

The problem of simultaneous optimization of channel and power allocation has largely the same properties as the channel optimization problem. The aspects discussed in Section 4.1 therefore also concern this section. Additional information to the model description, discussed next, is found in Section 4.1.1.

6.1.1 Model description

As the optimization criterion for channel optimization was solely based on the received interference, it is necessary to use the signal-to-interference ratio (SIR) when optimizing both channel and power allocation.

When only interference is used as a measuring criteria, all APs would simply decrease their powers to their minimum to reduce the total interference. This results in a great reduction in coverage area. Thus, to retain a coverage area that is as large as possible, the signal strength must be measured as well. Also, the interference and signal strength will not be measured at each AP itself, because the signal has infinite strength at this point. Therefore, the SIR, in contrast to solely interference, is measured at a certain range from the AP.

Just as in the case of using only channel optimization, measuring at all possible positions is simply a too slow process for the optimization algorithm. Therefore, the SIR is measured at a selected number of positions. The exact determination of the measuring locations depends on the chosen objective and is discussed in Section 6.2.

Furthermore, the variation in transmitting power of each AP cannot be continuous, as the objective (in this case the SIR) has a non-linear dependency on the power and will result in an infinite number of power levels. Discretizing the powers allows using the branch-and-bound method discussed in Chapter 3. Using discrete power levels is comparable with real-life applications where it is more feasible for APs to have a fixed number of transmitting powers, than a continuous range of power levels.

6.2 Objectives

In contrast to the channel optimization problem, the objectives are now based on the signal-to-interference ratio (SIR), instead of the interference solely. There are several criteria by which the joint channel and power allocation problem can be optimized, but in every case the SIR has to be measured. Therefore, the optimization problem is given as:

$$\max_{x \in S} f(x) \tag{6.1}$$

where x is a matrix containing channel and power allocations for each AP, S is the solution space and $f(x)$ is the objective function.

The objective function $f(x)$ is based on the SIR and two variants are discussed next.

6.2.1 Single measure point per access point

As it is unfeasible to measure at each possible position, the signal-to-interference ratio (SIR) must be measured at selected locations. In Section 5.1.2, pre-ordering of access points showed that the APs close to the center of mass are more critical than the rest. Therefore, with each AP there is a measuring point located towards the center of mass (of all APs) at a certain range. At these points the SIR is calculated:

$$SIR_i(x) = \frac{S_i}{\sum_{j \neq i} I_{i,j}(x)} \tag{6.2}$$

where S_i is the signal from AP$_i$ and $I_{i,j}(x)$ is the interference at AP$_i$'s measuring point received from AP$_j$.

The objective function to be maximized is given as the minimum over all SIRs:

$$f(x) = \min_i SIR_i(x) \tag{6.3}$$

6.2.2 Multiple measure points per access point

The previously discussed objective serves optimize the critical measuring points. This objective serves to minimize the total overlap by measuring the SIR at multiple points for each AP. Each point is located at a certain range towards one of the other APs. The SIR of each AP can be specified in three different ways: the minimum of all points, the maximum of all points and the average of all points. Mathematically the SIR for AP$_i$ is denoted as:

$$SIR_i(x) = \min_k \frac{S_i}{\sum_{j \neq i} I_{i,j,k}(x)} \qquad \text{(minimum)} \tag{6.4}$$

$$SIR_i(x) = \max_k \frac{S_i}{\sum_{j \neq i} I_{i,j,k}(x)} \qquad \text{(maximum)} \tag{6.5}$$

$$SIR_i(x) = \frac{1}{k} \sum_k \frac{S_i}{\sum_{j \neq i} I_{i,j,k}(x)} \qquad \text{(average)} \tag{6.6}$$

where S_i is the signal from AP$_i$ and $I_{i,j,k}(x)$ is the interference at AP$_i$'s kth measuring point received from AP$_j$.

The objective function is again given as the minimum over all SIRs:

$$f(x) = \min_i SIR_i(x) \tag{6.7}$$

6.3 Branch-and-bound model

As with the channel allocation problem described in Chapter 4, the number of depths is still equal to the total number of APs. Each AP now only has to be assigned both a channel and a transmit power. As a result of these multiple power levels the number of nodes at each depth in the branch-and-bound tree is multiplied by the number of power levels. A tree where each AP can select one out of two available channels and one out of two power levels is shown in Figure 6.1. The resulting time complexity is:

$$O((c \cdot p)^n) \tag{6.8}$$

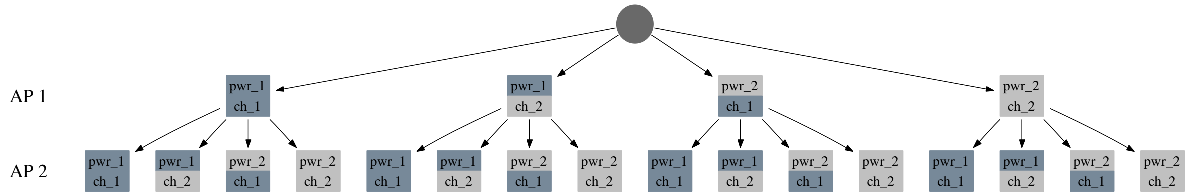


Figure 6.1: Branch-and-bound tree: joint channel and power optimization

where n is the number of APs, c the number of channels and p the number of power levels.

A model summery is given here:

- x: vector containing the channels that are assigned to the APs
- $c(x)$: there are no constraints for the channel optimization problem, thus $c(x)$ will always return *true*
- S: the solution space consists of all possible combinations of channel assignments to the APs
- *single measuring point*: this objective function $f(x)$ determines the signal-to-interference ratio (SIR) towards the center of mass, given a certain $x \in S$
- *multiple measuring points*: this objective function $f(x)$ determines the signal-to-interference ratio (SIR) of each AP from multiple points, given a certain $x \in S$
- $\max f(x)$: the total objective is to find the $x \in S$ that maximizes $f(x)$
- *time complexity*: the computational time required is $O((c \cdot p)^n)$, with n the number of APs, p the number of power levels and c the number of channels (13 in most of Europe and the UK)

Chapter 7

Joint channel and power optimization - implementation and results

In this chapter preliminary results are presented on joint channel and power optimization. That is, some introductory experiments have been performed of which the results will be discussed. These results give an indication of the application of power optimization and can be used for further research.

The implementation of the optimization algorithm is presented first, after which the simulation results are discussed. The simulation results are divided in a general part, for which statistics have been applied on random configurations, and a part about specific configurations.

7.1 Implementation

The objectives discussed in Section 6.2 are implemented in the optimization algorithm. Various number of power levels can be used in the optimization process, but using too many will quickly increase the time complexity and degrade the performance. Two or three power levels already show improvements over a single transmit power.

The absolute transmit powers are generated by taking a starting level and multiplying this with a constant factor for each subsequent level. Both the starting level and the multiplicative factor can be specified.

The range at which the measurement points are located can be set manually, but is recommended to set to about $\frac{1}{3}$ of the mean separation. A range that is too short will result in every AP reducing its power level to the minimum and thus reduces its coverage area, and a range too large will result in the measuring point being outside the coverage area of the AP and therefore disabling the optimization process.

Algorithm 3 MAXIMIZE()

Parameters:

Integer $depth \geqslant 0$, initialized to 0
Integer n, equal to the number of access points
Vector x_{pwr} of length n, containing the transmit power assignment
Vector x_{ch} of length n, containing the channel assignment
Vector $x = x_{ch} : x_{pwr}$ of length $2n$, the catenation of x_{ch} and x_{pwr}
Vector z of length $2n$, remembering the x that gave the best $f(x)$ so far
Integer $num\_channels$, containing the available number of frequency channels
Integer num_power_levels, containing the available number transmit power levels
Vector $power$ of length num_power_levels, containing the actual transmitter powers
Local matrix $child$ of size $[n \text{ x } 3]$, containing $\alpha(x)$, transmit power and channel for evaulated child nodes
Function $\alpha(x)$, representing an upper bound for $f(x)$
Real ω, representing the lower bound, initialized to minus infinity

```
depth++
at_leaf = false
m = 0
for i = 1 to num_power_levels do
    x_pwr[depth − 1] = power[i]
    for j = 1 to num_channels do
        if depth == 1 and 2j > num_channels then
            break
        end if
        x_ch[depth − 1] = j
        if α(x) < ω then
            child[m][0] = α(x)
            child[m][1] = x_pwr[depth − 1]
            child[m][2] = x_ch[depth − 1]
            m++
            if depth == n then
                at_leaf = true
                ω = α(x)
                z ← x
            end if
        end if
    end for
end for
if !at_leaf then
    permute rows of child so that column 0 is decreasing
    for row in child do
        x_pwr[depth − 1] = row[1]
        x_ch[depth − 1] = row[2]
        MAXIMIZE()
    end for
end if
depth−−
return
```

	Channel *avg*	Channel *max*	Joint channel and power
Complementary solutions	x	x	x
Channel spacing	x	x	x
Pre-ordering	x	x	x
Initial random solution	x	x	x
Incremental objectives	x		
Symmetric AP distance matrix	x	x	

Table 7.1: Programming tricks for joint channel and power optimization

7.1.1 Branch-and-bound: local best-first search

The local best-first branch-and-bound method used for the channel optimization problem is also used for the joint optimization problem discussed in this chapter. This model has proved itself in the previous results and the channel optimization algorithm can be easily extended to the joint optimization algorithm. The pseudo-code for the local best-first search method that includes power allocation and which maximizes the SIR is given in Algorithm 3.

7.1.2 Programming tricks

Several tricks discussed in Section 5.1.2 and used for the channel optimization problem are also applicable to the joint channel and power optimization problem. Table 7.1 shows the applicability of each programming trick. Two of the six tricks cannot be applied and are discussed next.

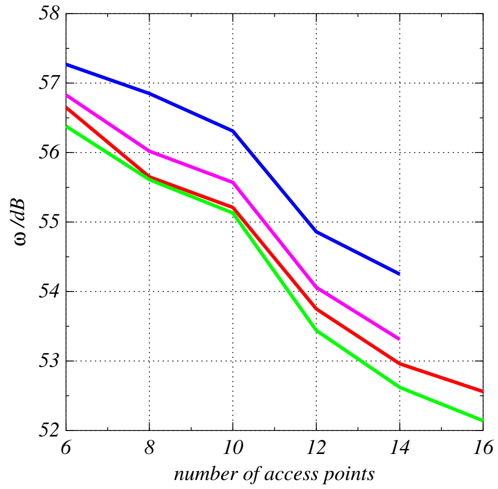
Incremental objectives - the objective functions in Equations 6.3 and 6.7 are unsuitable to update incrementally, because of the state information required. Compare this with the *max* objective for the channel optimization problem in Equation 4.6.

Symmetric AP distance matrix - this matrix was symmetric as a result of equal powers. Now the transmit powers are varied this property is no longer valid, because the interference caused by AP_i to AP_j is not necessarily equal to the interference caused AP_j to AP_i by any more.

7.2 Simulation results

Several simulations have been performed. The objectives with single or multiple measuring points are used to determine an optimal channel and power allocation scheme. For both objectives it is of interest what the maximum signal-to-interference ratio (SIR) is that can be achieved, as well as the computational time to perform the optimization.

In Section 7.2.1 random configurations are discussed, generated by a quasi-random number generator using the Niederreiter algorithm, see Appendix A.



Parameters:	
Transmit power	20 dBm
Power increase factor	0.80
Path loss exponent	2.86
Reference path loss (1m)	40.2 dB
Mean separation	50 m
Measuring range	15 m
Channel spacing	3
Quasi-random generator	Niederreiter

Figure 7.1: Performance of different joint channel and power optimization objectives: using a single measuring point and multiple points (*min*), (*avg*) and (*max*)

In Section 7.2.2 a number of special configurations are tested. Some of these cases consider real-life configurations, while others consider a special case by allowing transmitter powers equal to zero (i.e. turned off).

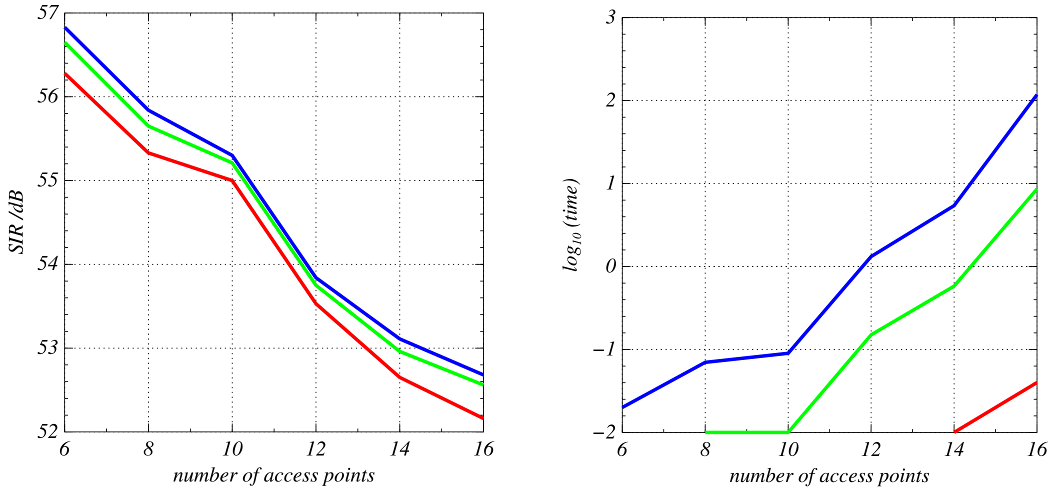
7.2.1 Random configurations

In Figure 7.1 all objectives discussed in Section 6.2 show similar behavior. The objectives using multiple points per AP clearly represent their respective way of determining the SIR, as the *max* has the highest values, *avg* intermediate values and *min* the lowest values. The objective values of using a single point is situated close to the *min*. This shows that using multiple measuring points per AP shows no apparent advantages at this time, but does increase the time complexity.

Figure 7.2a shows the improvement when access points can transmit at multiple power levels. From a theoretical point of view it is expected that when an AP can transmit at multiple powers, the objective will always be equally good or better. Simply, because the same channels and power levels are available as well, and can therefore always choose a solution with one power level.

As the number of APs increases the objective ω show a degradation, which is also seen for using channel optimization solely. In Figure 5.5 the total interference increases slowly with the number of APs, as more APs induce more total interference. Note that the degradation is only a few dB and starts to level off for larger number of APs, as in Figure 5.5.

The computational time related to the time complexity increases when the number of transmit powers increases (see Equation 6.8). Figure 7.2b shows the increase in time with an increasing number of APs for 1, 2 and 3 transmit power levels. The increase in time complexity becomes clear when comparing the computational time for 2 power



(a) Objective value: minimum SIR

(b) Computational time (resolution: 10ms)

Figure 7.2: Averages over quasi-random configurations with three numbers of power levels: 1, 2 and 3

levels with Figures B.1b and B.3a. The times are comparable, but a reduced number of channels (i.e. larger spacing) is used for joint channel and power optimization.

7.2.2 Specific configurations

Soho

One of the locations where BT deployed a number of wireless access points is the Soho area in London. Figure 7.3 shows an overview of this area. As an illustration a small part consisting of 10 APs has been taken to optimize, using joint channel and power allocation.

Figure 7.4a and 7.4b show the change in objective and throughput when multiple power levels are used. Below each figure the objective value is given, together with the coverage area for modulation modes 3 and 4 (see Section 4.1.3). The performance increase is only small as we have seen before for the random configurations (7.2a). The increase is time is however substantial. Even though this example with two fixed power levels shows little improvement, it is possible to improve the objective by more extreme power control, as discussed in the next section.

Extreme power control

An extreme form of power control is reducing the power to zero, i.e. turning the AP off. This form of power control is especially useful if there is a single bottleneck, which prohibits the optimization process from finding a better solution. Depending if the AP

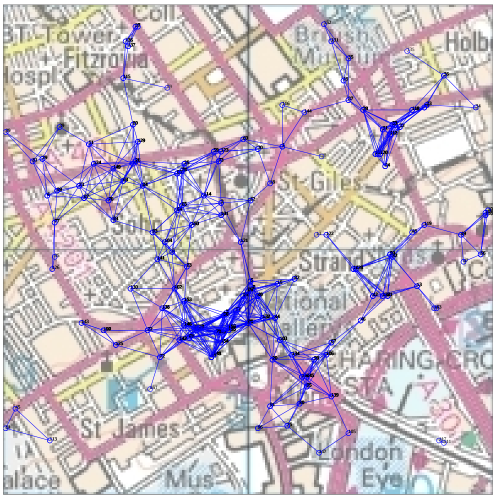
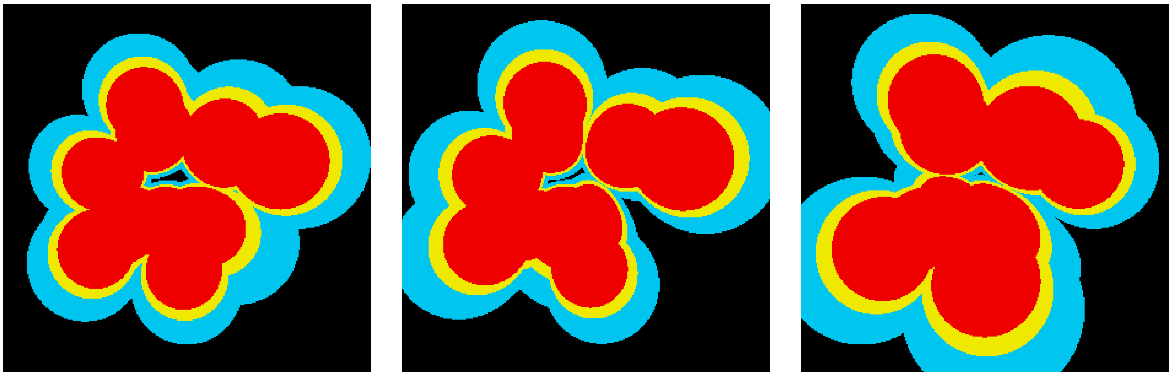


Figure 7.3: Soho interference graph that connect potential interfering access points



(a) 1 transmit power

(b) 2 transmit powers

(c) 2 transmit powers, omitting 1 access point (AP4)

Min. SIR:	24.18 dB	Min. SIR:	24.49 dB	Min. SIR:	26.23 dB
Mode 3 or 4:	0.3625%	Mode 3 or 4:	0.3761%	Mode 3 or 4:	0.4402%
Mode 4:	0.2852%	Mode 4:	0.2920%	Mode 4:	0.3368%
Time:	0.02s	Time:	1.26s	Time:	0.24s

Figure 7.4: Change in coverage area when using an extra power level and using extreme power control

has a critical role in the optimization process the solution may change a lot or not at all. The option to omit certain APs and to allow the optimization algorithm to choose which APs to use can improve the final objective considerably.

This option has not yet been implemented in the optimization algorithm, because this imposes some difficulties. As an AP is turned off its measuring point should not be included when determining the objective value. Otherwise, the objective would be largely degraded, as this point has no signal and therefore a very bad SIR value. On the other hand, when turning off APs the total interference will obviously decrease. Therefore, by turning off all but one AP, the best result in SIR will be achieved, but not in coverage. Thus, a counter measure to retain a large coverage area is necessary.

Still, to show the advantages of turning off one AP, a manually modified experiment has been performed. For the Soho configuration discussed previously one of the APs is turned off. The result in shown in Figure 7.4c, where the increase in throughput and coverage can be seen, with obviously a better value for the minimum SIR as well. Thus by removing the bottleneck, an overall better performance is obtained. A reduction in computational time is found as a result of having to optimize one less access point.

Chapter 8

Conclusions

This report shows a study on simultaneous channel and power optimization. This combinatorial optimization (CO) problem is tackled in a structured way by first solving the simpler problem of channel optimization solely. The problem is then extended to joint channel and power optimization allocation.

An optimization algorithm has been created to solve both the channel and joint channel and power optimization problems. A large number of options have been implemented to make it a general purpose algorithm, applicable to a large variety of configurations. The main options are: generating a quasi-random configuration or using locations from a file, the log-distance path loss model, adjustable transmit power levels and adjustable number of channels and channel spacing.

Local best-first search

The problem of channel or joint channel and power optimization is a NP CO problem and therefore increases quickly in size when the number of access points (AP) increase. A modified branch-and-bound algorithm is proposed to solve the problem as quickly and efficient as possible. This local best-first algorithm is a combination of a depth-first and a best-first search. The proposed algorithm combines the properties of finding a feasible solution quickly and proceeding with the best (partial) solution. A quick feasible solution improves the pruning process, and continuing with the best solution allows the algorithm to find the optimal solution as quick as possible.

In the process of implementing the optimization algorithm, a number of speed-up tricks are used to increase the performance even further. These tricks are: omitting complementary solutions, increase the channel spacing, pre-ordering of the APs, using random initial solutions, using incremental objectives and using a symmetric AP distance matrix. All these tricks are described extensively in Chapters 5 and 7 and also show in an applicability matrix for all discussed objective functions.

Channel optimization

The problem of channel allocation is based on the amount of received interference at each AP. Two objectives have been discussed, which are to minimize the average interference (*avg*) and to minimize the maximum interference (*max*).

For both algorithms, channel spacings 1, 2 and 3 offer little difference in objective value, but do differ in time complexity. Moreover, a problem that uses a larger channel spacing can be solved in less time. An interesting result is that for both objectives a channel spacing of 3 outperforms a channel spacing of 2. This makes it advantageous to implement a channel spacing of 3, which results in only a little degradation in performance compared to channel spacing 1, but greatly reduces the computational time. By increasing the channel spacing further than 3, the performance degrades substantially.

The mean separation, which is considered 50 meters, between the two closest APs has no influence on the optimization process. The final interference levels change in absolute value, but the resulting channel allocation schemes are equal. This is a result of equal degradation in both signal and interference level, i.e. the ratio stays the same.

The simulation results show that configurations up to 35 access points can be optimized within reasonable time. That is, depending on the objective and channel spacing an optimal solution can be found within a day on a 64-bit 2.2 GHz desktop computer.

Joint channel and power optimization

In contrast to the channel optimization problem, when optimizing channel and power simultaneously, both the amount of interference and the signal level are important. For this problem the signal-to-interference ratio (SIR) has to be maximized.

Both objectives that have a single measuring point per AP and objectives that have multiple measuring points per AP have been implemented. The results show that using more than one measuring point has no apparent advantages over a single measuring point. Most of the times the objectives are equal and only differ in the computational time. Note that these results apply to the specific objective functions that have been implemented.

Optimization using a small number of transmit power levels (i.e. 2 or 3) shows only a minor improvement in the SIR for each objective. It is however possible to obtain larger improvements by allowing a transmit power of zero. This indicates that more radical power control or having more flexibility in transmit power levels (i.e. a wider range or more discrete values) is able to improve the objective substantially. The exact implementation and consequences for the time complexity are still to be determined.

Using fixed measuring points results in SIR values that are optimized at these specific locations. Therefore, the positions that are most likely to receive the most interference (which are the measuring points) are optimized. At this time the results show no guarantee that optimizing these locations improves the overall SIR (over the whole map). Mapping SIR values to user throughput maps shows also that there is no linear relation between the objective functions and the overall throughput performance and can change independently of the objective value. However, using a more advanced or extreme control of power can substantially improve the performance, as seen when omitting APs.

Recommendations

The problem of channel optimization has almost completely been covered in this report. However, for joint channel and power optimization still a lot of research that can be done. There are a number of directions that are interesting to pursue and will be discussed next.

The current optimization algorithm has fixed measuring points, which are not related to the transmitter power. This prohibits in some way the algorithm to choose power levels that are decreased a lot, because this results in a decreased signal at the measuring point decreases. By dynamically changing the measuring range as a function of the transmit power a better algorithm is obtained.

For a large number of problems the channel allocation scheme that is obtained using joint channel and power optimization is equal to the one obtained by channel optimization solely. In this case the time complexity can be reduced largely by splitting the channel and power optimization process. An optimal channel allocation scheme can be obtained using channel optimization solely and in the next step the objective can be improved further by varying the transmitter power levels.

Another way of splitting up the joint optimization process is by first optimizing the power allocation in such a way that the area covered by each access point is equal. This has the advantage of distributing the load equally over all APs. After that an optimal channel allocation scheme can be found using the channel optimization process. By splitting up the process the time complexity is again reduced largely, compared to a joint objective.

The last recommendation concerns both channel and joint channel and power optimization. For both optimization algorithms the simulations showed that when the number of access points increased the objective value starts to level off. This is related to the mean separation between the two closest access points. When saturating the area with access points, but retaining the mean separation it is probable that there is a bound on the objective value. Knowing and relating this bound to different values of the mean separation allows to give a good prediction on configurations with a known mean separation.

Appendix A

Quasi-random number generators

Quasi-random number generators are used to generate access point (AP) locations in a square area. Random number generators are useful, because it allows us to do statistics over a large number of simulations.

APs are normally not located close together, as this would have no advantage and only cause a lot of interference to each other. Therefore, pseudo-random generators are a little too random. A scatter plot of a pseudo-random number generator has places that are relatively undersampled and other places have clusters of points. The quasi-random number generator tries to maintain a uniform density of coverage and is therefore more useful for this application.

A.1 GNU Scientific Library

The GNU Scientific Library (GSL) offers a number of quasi-random number generators that can be implemented in a C program [5]. These generators are: Niederreiter [2], Sobol [1], Halton [6, 8] and reverse Halton [6, 8].

Figure A.1 shows three scatter plots with AP locations, using the Niederreiter algorithm (used for all simulations). The average separation between the two closest APs is approximately 50 meters.

A.2 Algorithm perfomance

The algorithms that specify the quasi-random number generators, influence the computational time and final result for any application and in this case the channel optimization problem. A simulation shows the influence on the computational time and minimum amount of received interference, see Figure A.2. The simulation parameters are given in Table A.1.

The figures show little change when using different algorithms and therefore indicates that the optimization process is roughly independant of the AP distribution.

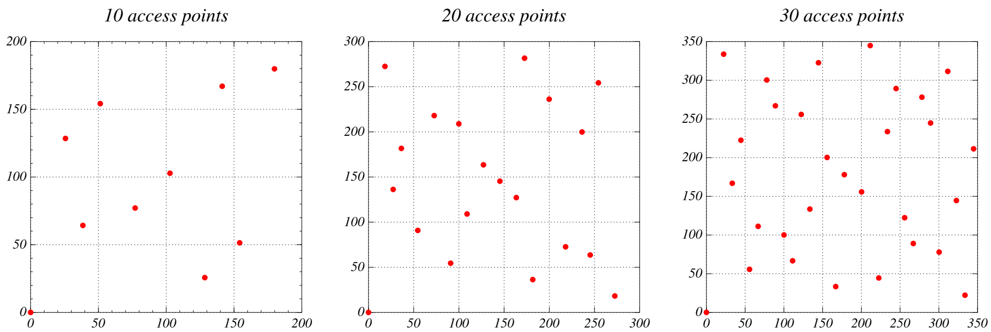


Figure A.1: Quasi-random generated access points locations, using the Niederreiter algorithm

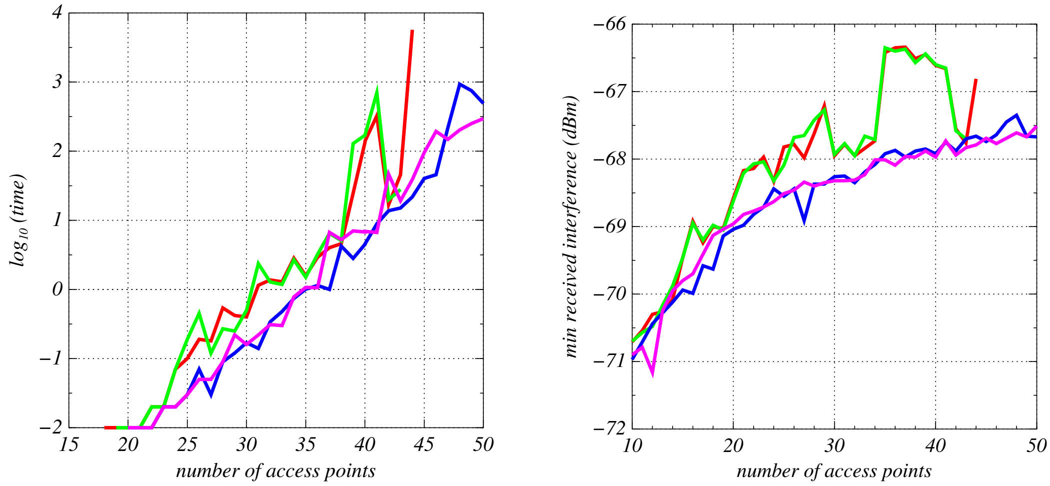


Figure A.2: The computational time and minimum received interference in a channel allocation problem for 4 different quasi-random algorithms: Niederreiter, Sobol, Halton and reverse Halton.

Objective	*max*
Transmit power	20 dBm
Path loss exponent	2.86
Reference path loss (1m)	40.2 dB
Mean separation	50 m
Channel spacing	6
Pre-ordering	*true*

Table A.1: Simulation parameters for quasi-random number generators

Appendix B

Graphs channel optimization

In Chapter 5 the results for the channel allocation problem for wireless cities are discussed. The simulation results have been combined into a smaller number of figures to show the essential graphs.

This appendix shows extra graphs for the channel allocation problem. For both the *avg* and *max* objective the results on computational time and received interference are shown for every value of channel spacing.

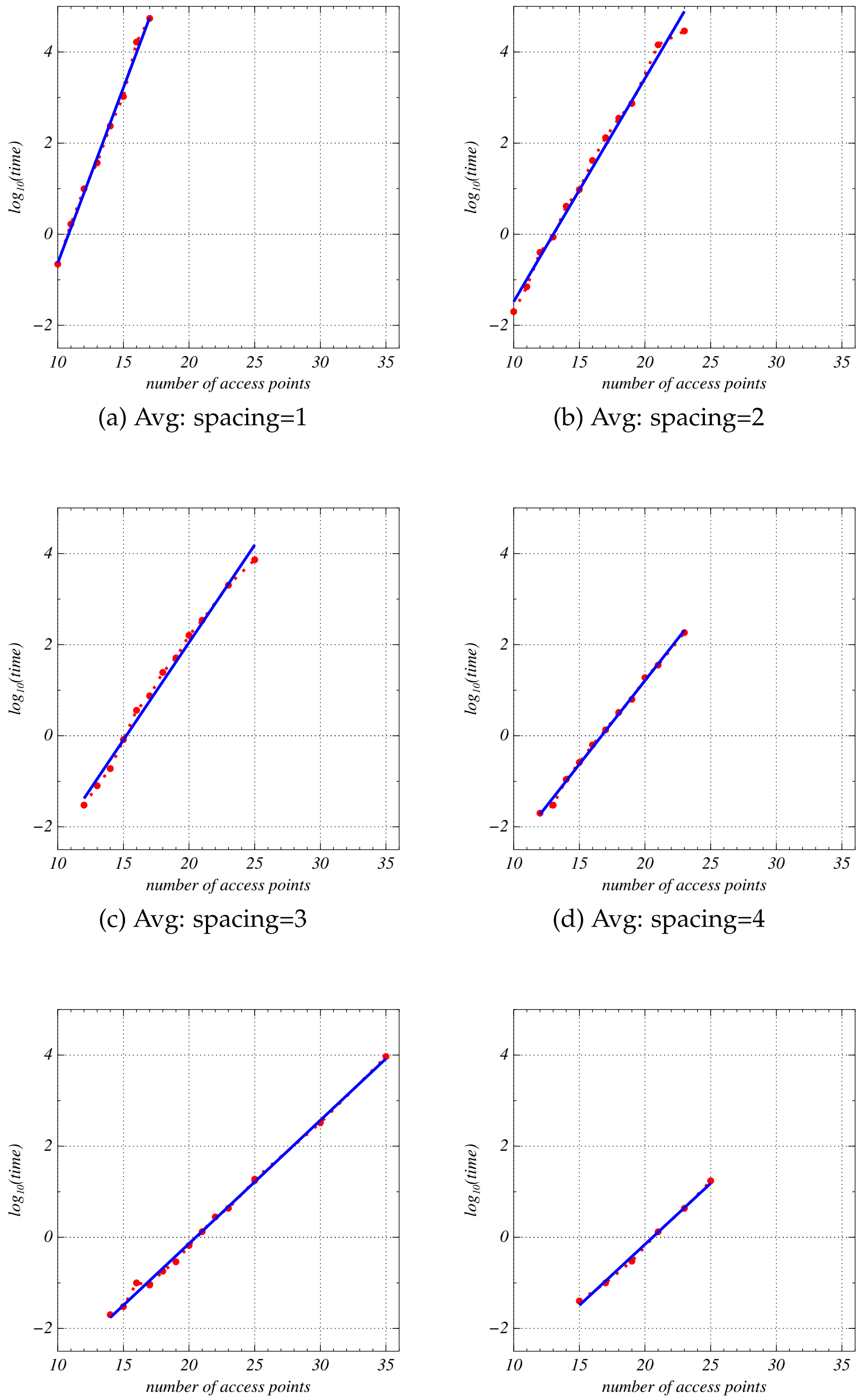
Each figure will show the graphs for channel spacing 1 to 6:

	time	*omega*
avg	Figure B.1	Figure B.2
max	Figure B.3	Figure B.4

Figure B.1 and B.3 show that the computational time depends logarithmically on the number of access points (APs). When the channel spacing increases the time graphs start at a lower point and the slope is less steep, also explained in Section 5.2.2.

Figure B.3f shows a very wild behaviour which indicates that for higher values of the channel spacing (i.e. few channels available) the AP locations heavily influence the performance of the algorithm.

The received interference in Figure B.2 and B.4 increases quickly in the beginning when the number of access opints increases, but levels off for higher number of APs. The received interference tends to saturate when the number of APs keeps increasing with a mean separation of 50 meters between the two closest APs.



(a) Avg: spacing=1

(b) Avg: spacing=2

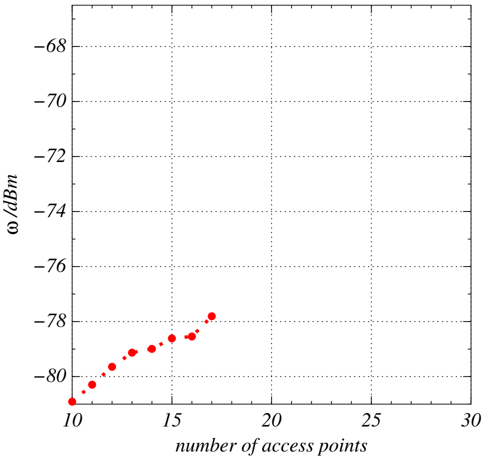
(c) Avg: spacing=3

(d) Avg: spacing=4

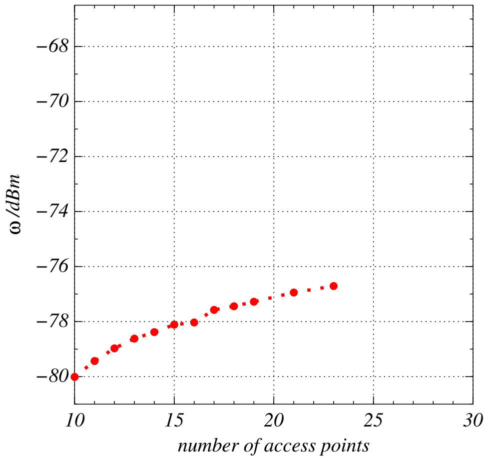
(e) Avg: spacing=5

(f) Avg: spacing=6

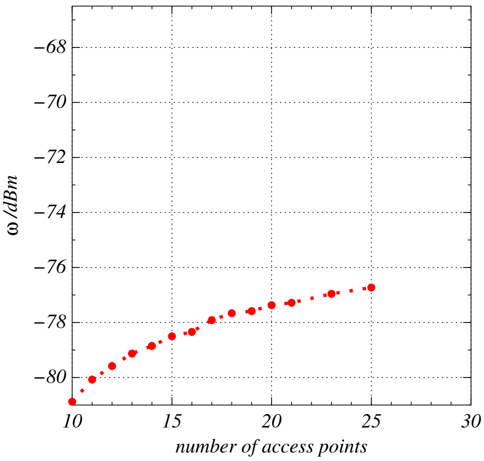
Figure B.1: Channel optimization *avg*: computational time



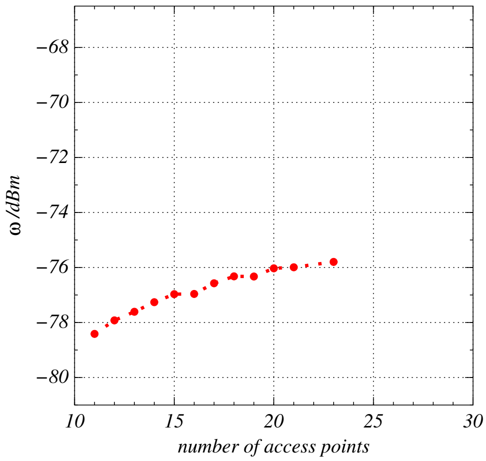
(a) Avg: spacing=1



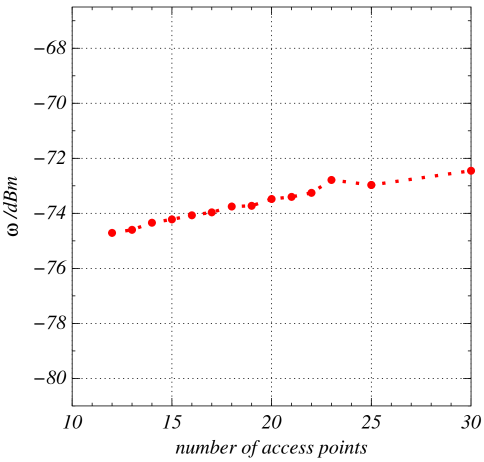
(b) Avg: spacing=2



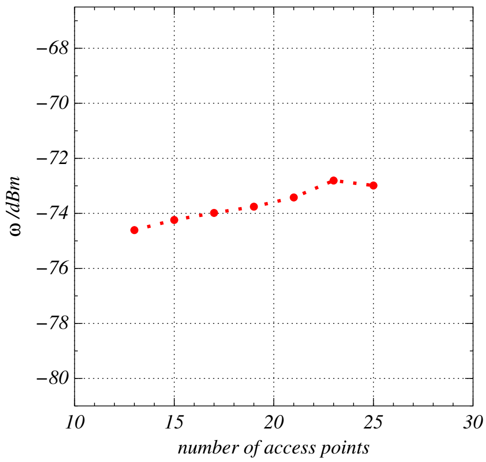
(c) Avg: spacing=3



(d) Avg: spacing=4

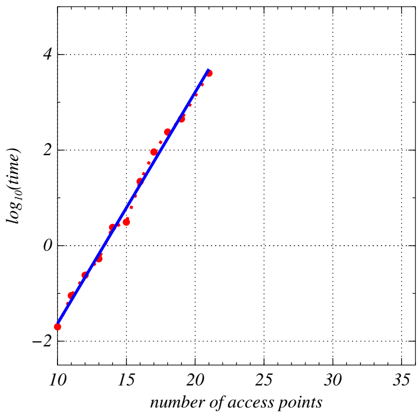


(e) Avg: spacing=5

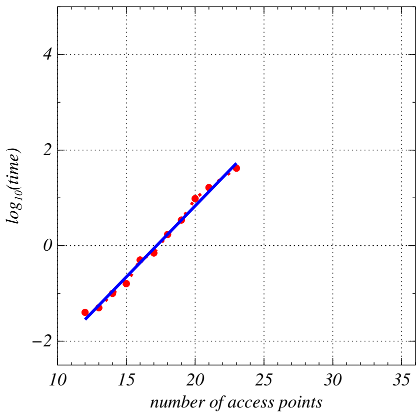


(f) Avg: spacing=6

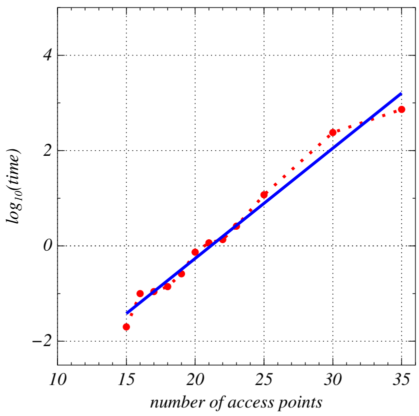
Figure B.2: Channel optimization *avg*: received interference



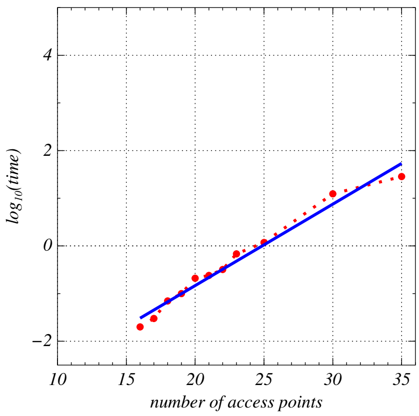
(a) Max: spacing=1



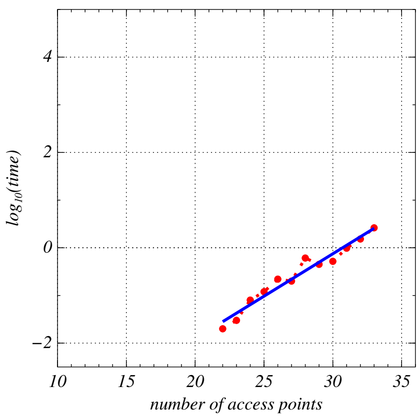
(b) Max: spacing=2



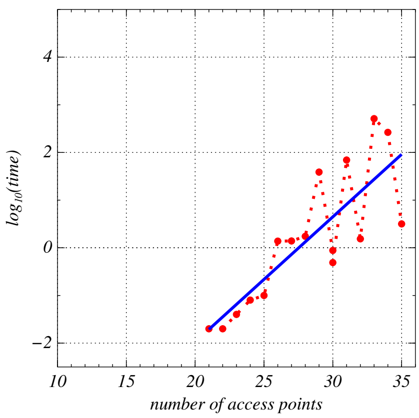
(c) Max: spacing=3



(d) Max: spacing=4

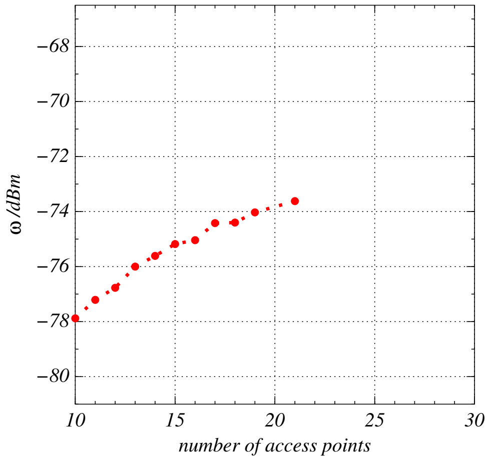


(e) Max: spacing=5

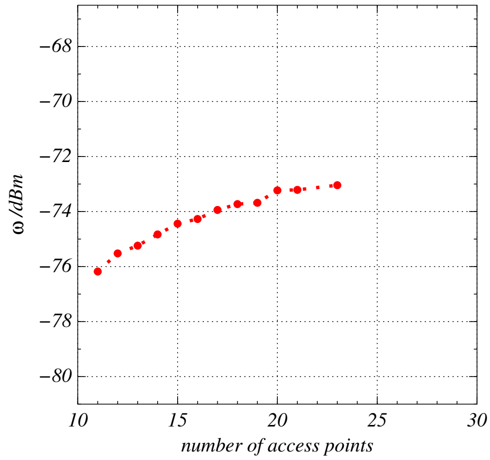


(f) Max: spacing=6

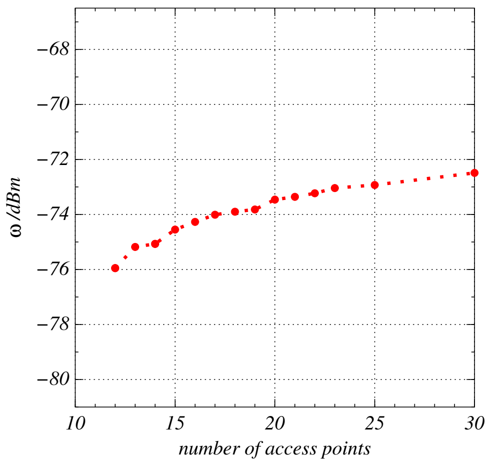
Figure B.3: Channel optimization *max*: computational time



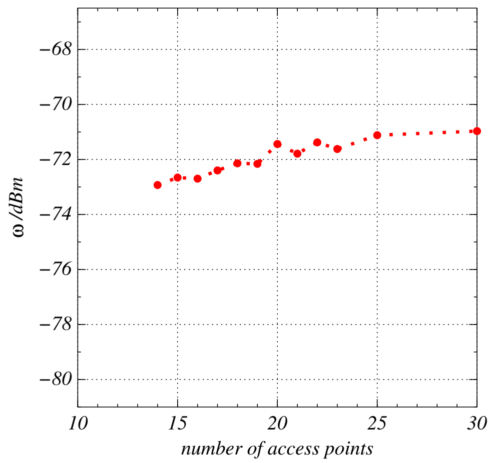
(a) Max: spacing=1



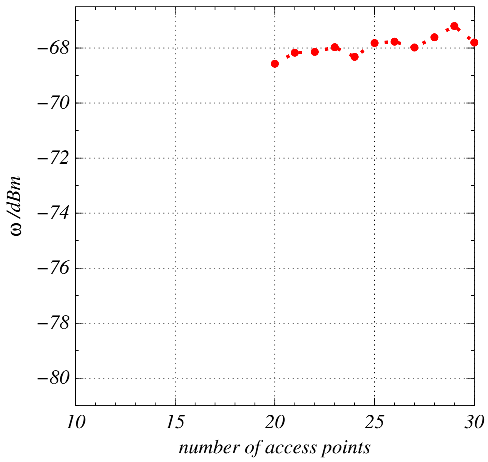
(b) Max: spacing=2



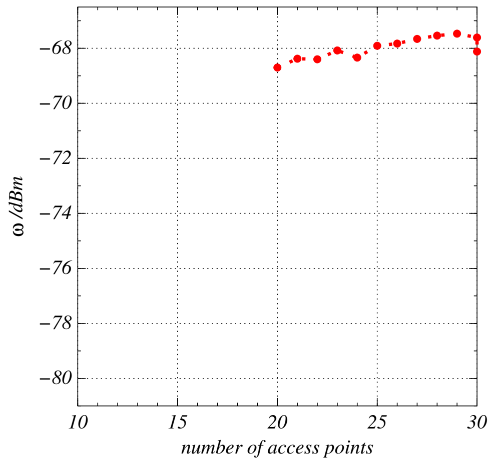
(c) Max: spacing=3



(d) Max: spacing=4



(e) Max: spacing=5



(f) Max: spacing=6

Figure B.4: Channel optimization *max*: received interference

Bibliography

[1] I. A. Antonov and V. M. Saleev. An economic of computing LP-sequences. *USSR Comput. Math. Math. Phys*, 19:252–256, 1979.

[2] P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy point sets. *ACM Trans. on Modeling and Comp. Simul.*, 2:195–213, 1992.

[3] K. M. Briggs. Rate control for 802.11 systems. Unpublished report, April 2008. Mobility Research Centre, BT Research (Martlesham, UK).

[4] K. M. Briggs and A. Ainsworth. Optimal channel allocation for wireless cities. In *VTC2008 Spring*, 2008.

[5] Inc. Free Software Foundation. Quasi-random sequences. Website, May 2008. http://www.gnu.org/software/gsl/manual/html_node/Quasi_002dRandom-Sequences.html.

[6] J. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2, 1960.

[7] T. S. Rappaport. *Wireless communications, principles and practice*. Prentice Hall, Inc., 2nd edition, 2002.

[8] B. Vandewoestyne and R. Cools. Good permutations for deterministic scrambled Halton sequences in terms of L2-discrepancy. *Computational and Applied Mathematics*, 189(1:2):341–361, 2006.