Approximation Algorithms for Optimization Problems in Future Cellular Networks

David Amzallag
Approximation Algorithms for Optimization Problems in Future Cellular Networks

Research Thesis

Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

David Amzallag

Submitted to the Senate of the Technion - Israel Institute of Technology

Shvat 5768 Haifa January 2008
This research thesis was done under the supervision of

Prof. Seffi Naor and Prof. Danny Raz

Department of Computer Science
Technion - Israel Institute of Technology
Haifa 32000, Israel
Acknowledgements

I am indebted to Prof. Seffi Naor and Prof. Danny Raz for their guidance and support throughout my graduate studies. Seffi and Danny gave me the opportunity to follow my personal interests and to establish a theoretical as well as practical research, and always been there when needed with warm advise or clever idea (or both).

Special thanks are due to my following coauthors, colleagues, and friends for taking time to contribute, read manuscripts, and give their valuable comments: Eyal Ackerman, Tomer Armarnik, Reuven Bar-Yehuda, Roee Engelberg, Michael Livschitz, Avi Owshanko, and Gabi Scalosub.

Financial support was provided by the Technion and by REMON - Israel 4G Mobile Consortium (sponsored by Magnet Program of the Chief Scientist Office in the Ministry of Industry and Trade of Israel). I am grateful to both for their help. The participation in REMON was valuable, enrichment, and gave me a stage for evaluate results in a practical, industry-oriented, point of view.

Last, but not least, is my wife Sarit, and my children Shira, Alona, and Yiftach. This work would never have been finished (and started) without their help and encouragement. It is a great honor for me to be a part of this family.
# Contents

Abstract 1

Abbreviations and Notations 3

1 Introduction 5
   1.1 Coping with NP-hard problems on practice 5
   1.2 On discrete location problems in combinatorial optimization 7
   1.3 Beyond 3G and 4G cellular networks 8
   1.4 Organization of this thesis 10

2 Cell Planning of Future Cellular Networks 13
   2.1 Introduction 13
      2.1.1 Formulation and background 14
      2.1.2 How to model the interference? 15
      2.1.3 Our contributions 17
   2.2 Previous work 18
   2.3 The budgeted cell planning problem 20
      2.3.1 Inapproximability 21
      2.3.2 The $k4k$-budgeted cell planning problem 22
      2.3.3 The structure of BCPP solutions 22
      2.3.4 The client assignment problem 26
      2.3.5 The budgeted maximum assignment problem 28
      2.3.6 Approximating $k4k$-BCPP 31
   2.4 The minimum-cost cell planning problem 32
   2.5 Simulation results and practical aspects 35
      2.5.1 On greedy algorithms for minimum-cost cell planning 36
      2.5.2 An LP-rounding based $O(W\sqrt{\log n})$-approximation algorithm 37
      2.5.3 Planning “greenfield” networks 39
      2.5.4 Planning UMTS network in Helsinki 40
   2.6 Conclusions and open problems 41

3 Algorithmic Aspects of Radio Access Networks Design 43
   3.1 Introduction 43
### 3.1 Definitions and background
- 3.1.1 Definitions and background
- 3.1.2 Our contribution

### 3.2 Related work

### 3.3 The bounded-degree minimum routing cost spanning tree problem
- 3.3.1 When the degree constraints are the lowest possible
- 3.3.2 An $O(\log n)$-approximation algorithm
- 3.3.3 Heuristic algorithms

### 3.4 Extensions

### 3.5 Simulation results

### 3.6 Conclusions and open problems

### 4 Cell Selection in Future Cellular Networks
- 4.1 Introduction
- 4.1.1 Our Contribution
- 4.2 Model and definitions
- 4.3 Related work
- 4.4 Approximating AoNDM and $r$-AoNDM problems
  - 4.4.1 A cover-by-one $\frac{1-r}{2-r}$-approximation algorithm
  - 4.4.2 A cover-by-many $(1 - r)$-approximation algorithm
- 4.5 Simulation results
- 4.5.1 Methodology
- 4.5.2 Results
- 4.6 Conclusions and open problems

### 5 Discussion

### References
# List of Figures

1.1 Range versus data rates for each of the generations ........................................ 9  
2.1 Planning future networks ................................................................................. 40  
2.2 Building contours map of the city center of Helsinki ........................................ 41  
3.1 RAN architecture in current 3G systems ............................................................ 43  
3.2 Tree-topology access network .......................................................................... 44  
3.3 Approximating BDRT ....................................................................................... 51  
3.4 A tight example for the $O(\log n)$-approximation algorithm .............................. 52  
3.5 A bad example for GA$_1$ .................................................................................. 53  
3.6 Sample execution of the five algorithms. ............................................................ 55  
3.7 Simulation results of BDRT .............................................................................. 56  
3.8 GBDRT simulation ........................................................................................... 57  
4.1 Bad behavior of the best detected SNR algorithm in high-loaded capacitated network ................................................................................................................ 62  
4.2 Expected profit as a function of the number of clients for different values of $r$ 77  
4.3 Results for the second and the third set of simulations ........................................ 78
List of Tables

2.1 Previous work on cell planning of cellular networks .................. 19
2.2 Cell planning in Helsinki ........................................... 41

3.1 Adjacency matrix for the sample execution ....................... 54
3.2 Algorithms for solving BDRT .................................... 59
3.3 Algorithms for solving GBDRT .................................... 59
This page was intentionally left blank
Abstract

The main goal of this research is to develop new approaches specifically addressing several fundamental optimization problems associated with the upcoming future cellular networks. This research rigorously studies algorithmic aspects of these optimization problems, incorporates the anticipated future new technologies into the studied framework, and presents new methods for solving these problems. Our new techniques are based on novel modeling of technology dependent characterizations and using approximation algorithms in order to provide provable good solutions. In addition, methods presented in this work are also applicable to various current network radio technologies.

We start by studying the problem of planning future cellular networks. In general, cell planning includes choosing a network of base stations that can provide the required coverage of the service area with respect to current and future traffic requirements, available capacities, interference, and the desired QoS. We rigorously study both the budget limited cell planning problem and the minimum-cost cell planning problem.

Next we address a challenging problem that arises in the design of a new topology for radio access network (RAN) for future cellular networks. In this problem we study algorithmic aspects of replacing the commonly used star based architecture, in which a Radio Network Controller (RNC) is connected to a set of base stations via direct links, with a more complex tree structure, in which a base station can be connected to an RNC via other base stations.

Finally, motivated by algorithmic characterizations of the new OFDMA multiple access technique standard (as defined by the IEEE 802.16e-2005), we design a new approach for the cell selection mechanism. Cell selection is the process of determining the cell(s) that provide service to each mobile station. Optimizing these processes is an important step towards maximizing the utilization of current and future cellular networks.

In this thesis, we provide the first approximation algorithms for all of the above problems. Our results indicate that a theoretical study together with optimized practical implementations outperform previous commonly used techniques, and achieve a close-to-optimal solutions on real cellular networks.
This page was intentionally left blank
## Abbreviations and Notations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I$</td>
<td>A set of possible base station configurations (BS)</td>
</tr>
<tr>
<td>$I'$</td>
<td>A set of base station configurations selected for opening</td>
</tr>
<tr>
<td>$J$</td>
<td>A set of clients (or <em>mobile stations</em>, or <em>bins</em>) or RNCs</td>
</tr>
<tr>
<td>$w_i$</td>
<td>The capacity of a base station configuration, $i \in I$</td>
</tr>
<tr>
<td>$c_i$</td>
<td>The installation cost of a base station configuration, $i \in I$</td>
</tr>
<tr>
<td>$S_i$</td>
<td>The set of clients admissible to be covered by base station $i$, $S_i \subseteq J$</td>
</tr>
<tr>
<td>$d_j$</td>
<td>The traffic demand of client $j$, $j \in J$</td>
</tr>
<tr>
<td>$Q(i,j)$</td>
<td>The contribution of BS $i$ to client $j$ after incorporating all interference</td>
</tr>
<tr>
<td>$P$</td>
<td>The interference matrix (also known as <em>impact matrix</em>)</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>The minimum required coverage rate of a client</td>
</tr>
<tr>
<td>$x_{ij}$</td>
<td>The fraction of the capacity of BS $i$ supplied to client $j$, $i \in I$, $j \in J$</td>
</tr>
<tr>
<td>$I_j$</td>
<td>The set of BSs participating in the coverage of client $j$, $j \in J$</td>
</tr>
<tr>
<td>$W$</td>
<td>The largest capacity over all base stations selected for opening</td>
</tr>
<tr>
<td>$w(u,v)$</td>
<td>The symmetric connection cost between base stations $u$ and $v$</td>
</tr>
<tr>
<td>$b(u)$</td>
<td>The maximum number of neighbors allowed for a vertex $u$</td>
</tr>
<tr>
<td>$d_T(r,v)$</td>
<td>The routing cost between a root $r$ and a vertex $v$ on a tree $T$</td>
</tr>
<tr>
<td>$p(j)$</td>
<td>The profit for satisfaction the demand of a client $j$, $j \in J$</td>
</tr>
</tbody>
</table>
This page was intentionally left blank
Chapter 1

Introduction

It is anticipated that the next 5 years about 350 million people will be connected to the cellular communications network, which means approximately 200,000 new subscribers every day or 140 new subscribers every minute. For most of them it will not only be their first mobile phone, it will be their first phone, and for at least half of them it will be their first camera, their first music player and, of course, their first Internet access.

With each passing day, the maturity level of the mobile user and the complexity level of the cellular network reaches a new high. Current networks are no longer “traditional” GSM networks, but a complex mixture of 2G, 2.5G, and 3G technologies. Furthermore, but new technologies beyond 3G (e.g., 3.5G or HSPA) are being utilized in these cellular networks. The very existence of all these technologies in one cellular network has brought the work of designing and optimization of the networks to be viewed from a different perspective. Gone are the days of planning GSM, GPRS or WCDMA networks individually. Now the cellular network business is about dimensioning for new and advanced technologies, planning and optimizing 4G networks, while upgrading 3G/3.5G networks.

In this thesis we develop new approaches for solving fundamental optimization problems that appears in the future cellular networks. This research rigorously studies algorithmic aspects of several optimization problems, incorporates the anticipated future new technologies into their framework, and presents new methods for solving these problems.

We start this introductory part by reviewing some of the most common approaches for coping with NP-hard optimization problems in practice, which is the main purpose of this research. We then overview (Section 1.2) the family of discrete location problems in combinatorial optimization and introduce a “new” type of problem for this important family, modeling the problem of planning a cellular network. Finally, we overview (Section 1.3) the background, the reasons, and the goals for developing future cellular networks.

1.1 Coping with NP-hard problems on practice

An important goal of the theory of algorithms is to produce efficient algorithms that solve computationally difficult problems. When considering the class of NP-hard combinatorial optimization problems, this goal is beyond our reach. As these problems need to be solved
routinely, we seek practical ways of coping with NP-completeness. Perhaps the most common approaches are the following:

– **Easy special cases.** Do not solve the problem in its full generality. Identify properties of the input instances that make the problem easier, and design an algorithm that makes use of these properties.

– **Somewhat efficient exponential algorithms.** Design an algorithm that always solves the problem whose running time is not polynomial, but still much faster than exhaustive search. This approach may be useful for inputs of moderate size.

– **Approximation algorithms.** Sacrifice the quality of the solution so as to obtain more efficient algorithms. Instead of finding the optimal solution, settle for a near optimal solution. Hopefully, this makes the problem easier.

– **Heuristics.** Design algorithms that work well on many instances, though not on all instances. This is perhaps the approach most commonly used in practice.

In this thesis we use three of the approaches mentioned above for coping with the optimization problems arising in future cellular networks. To illustrate these ideas we next sketch the story of one of the important problems in the design of future radio access networks (RAN).

**The bounded-degree minimum routing cost spanning tree problem**

Consider the *bounded-degree minimum routing cost spanning tree problem* discussed in detail in Chapter 3. We are given a complete graph $G = (V, E)$ and let $r \in V$ be a special vertex. Every vertex has a degree-constraint $b(v)$, and every edge $e \in E$ is associated with a cost $c(e)$, such that the edge costs obey the triangle inequality. The bounded-degree minimum routing cost spanning tree problem (BDRT) is to find a spanning tree $T$ of $G$, rooted at $r$, that minimizes the sum of the costs from $r$ to all other vertices in $G$, while every vertex $v$ has no more than $b(v)$ neighbors in $T$. Practically speaking, this models the problem of designing a tree-topology radio access network (RAN), rather than a star-based RAN, which is considered to be much more suitable for future cellular networks.

From a theoretical viewpoint, we are not aware of any previous study of this NP-hard problem and, in particular, no approximation algorithm is known for this problem. An interesting observation is that the objective of this problem can be far from the objective of the bounded-degree MST problem by a factor of $\Omega(n)$ (as in the case of unit-weight complete graph on $n$ vertices, all have a degree bound of two). So, these are different problems.

Now notice that when $b(r)$ is relatively large (e.g., $b(r) \geq n - 1$) then the shortest-path tree (SPT) of $G$, rooted at $r$, is the optimal solution for this polynomial-time solvable case. On the other hand, when $b(v) = 2, \forall v \in V$ this case is exactly the 2-repairman problem, a “notorious” NP-hard version of TSP that have two salespersons instead of one.

Using the SPT as a lower bound for every other solution of the problem, we designed a “non-intuitive” greedy algorithm for the cases for which $b(v) \geq 3, \forall v \in V$, and we show that
this is an $O(\log n)$-approximation algorithms. We present a tight example for this algorithm showing that the approximation factor cannot be further improved using this algorithm. Then, we discuss a family of “intuitive” greedy heuristics for this problem and show that in general, these fast techniques can be as far as $\Omega(n)$ from the optimal solution (meaning that these are “very bad” algorithms in a worst-case viewpoint). Surprisingly enough, these heuristics performed significantly better than the approximation algorithm in practice. Our suggested solution combines the best of the greedy heuristics and the proven approximation algorithm, generating a solution which is close to optimal in practical scenarios, yet can be efficiently computed for sufficiently large network sizes.

1.2 On discrete location problems in combinatorial optimization

Facility location problems are among the most well-studied problems in combinatorial optimization (see [73] for an excellent survey). In the traditional version of the facility location problem we wish to find optimal locations for facilities in order to serve a given set of client locations; we are also given a set of locations in which facilities may be built, where building a facility in location $i \in I$ incurs a cost of $f_i$; each client $j \in J$ must be assigned to one facility, thereby incurring a cost of $c_{ij}$, proportional to the distance between locations $i$ and $j$; the objective is to find a solution of minimum total (assignment + opening) cost. In the $k$-median problem, facility costs are replaced by a constraint that limits the number of facilities to be $k$ and the objective is to minimize the total assignment costs. These two classical problems are min-sum problems, i.e., the sum of the assignment costs goes into the objective function. The $k$-center problem is the min-max analogue of the $k$-median problem: one builds facilities in $k$ locations out of a given possible number of locations, so as to minimize the maximum distance from a given location to the nearest selected location.

A new “type” of discrete location problems is introduced in this thesis (Chapter 2). We call it cell planning problems. In a cell planning problem, every client $j \in J$ has a positive demand $d_j$ and every facility $i \in I$ has also a hard capacity $w_i$ and a subset $S_i \subseteq J$ of clients admissible to be satisfied by it. Two interesting situations can happen when satisfying the demand of a client. The first case is when multiple coverage is possible, meaning that several facilities are allowed to participate in the satisfaction of a single client, while in the second case clients can be satisfied only by a single facility. In addition, a penalty function is introduced, capturing the “interference” between radio channels of neighboring base stations (or facilities). In this model, for example, when the demand of a client is satisfied by two facilities, their “net” contribution is less than or equals the sum of their supplies. The minimum-cost cell planning problem is to find a subset $I' \subseteq I$ of minimum cost that satisfies the demands of all the clients (while taking into account interference for multiple satisfaction).

Observe that this “new” model cannot be seen as a special case of any of the known min-sum discrete location problems (e.g., there is no connection cost between facilities and clients), or as a “special NP-hard type” of a minimum-cost flow problem (e.g., how to model the penalties for multiple satisfaction?).
Our model captures the problem of planning a network of base stations that provides a full coverage of the service area with respect to current and future traffic requirements, available capacities, interference, and the desired QoS. Practically speaking, to the best of our knowledge, this is the first time the problem of planning a cellular network is modeled (and solved) while taking into account capacities of base stations - a crucial parameter in the optimization of future cellular networks.

1.3 Beyond 3G and 4G cellular networks

With the introduction of the first cellular networks in 1982, 2G in 1992 and 3G around 2004, the upcoming deployment of the fourth generation cellular networks is most likely to take place in the next 5–7 years. In the last few years there has been much discussion around 4G, with conferences and books published. Some people thought that it was appropriate to start discussion of 4G in 2002 given the likely ten years it would take to complete the standard and have equipment developed. An example of this was the Japanese authorities who announced a research program aimed at producing a system capable of delivering 100 Mbits/s to end users [71]. Others argued that 3G had failed, or was inappropriate, and 4G should be introduced rapidly in its place. As might be expected, some of the key proponents of a rapid introduction of 4G were manufacturers with products that they classified as being 4G. However, during the early part of 2003, the ITU\(^1\), mindful perhaps of the slow and somewhat uncertain introduction of 3G, decided to put on hold any discussions about 4G systems, thus delaying the introduction of 4G to 2012, if not later. Even without examining the technical issues, Figure 1.1\(^2\) suggests why 4G is likely to be different from 3G, and indeed perhaps may not even emerge.

Figure 1.1 illustrates that there is always a trade-off between range and data rate. Broadly speaking, higher data rates require more spectrum. More spectrum can only be found higher in the frequency band but higher frequency signals have a lower propagation range. Each generation has accepted a shorter range in return for a higher data rate. But, as Figure 1.1 shows, the next “step” in the process, where 4G might logically fit, is already taken by a mix of 3G enhancements, WiMax and WiFi. Indeed, interestingly, NTT DoCoMo plans [71] for a 4G technology talk of an OFDM-based solution in the 36 GHz band providing up to 100 Mbits/s of data. This is almost identical to the specification for the latest WLAN system, 802.11a, which is already available and exceeded by proposals for 802.11n. Figure 1.1 also tells us much about the structure of future communications networks. Higher data rate systems are preferred but with their shorter range can be economically deployed only in high-density areas. So we might expect a network where 2G is used to cover rural areas, 3G to cover urban and suburban areas along with key transport corridors, and WLANs providing very high data rates in selected hotspot locations. Finally, Figure 1.1 also illustrates another constraint that of backhaul. Many WiFi cells are connected back into the network using ADSL. This has

\(^1\)See International Telecommunication Union (ITU) website http://www.itu.int/home/index.html.
\(^2\)Figure 1.1 and its analysis are partially based on W. Webb. Wireless Communications: The Future, John Wiley & Sons. Ltd., 2007.
data rates in the region of 12 Mbits/s at present, much lower than the rate that the WLAN air interface can provide. A further increase in air interface data rate in such a situation is useless. Instead, research needs to be focused on better types of backhaul. Hence, a popular argument is that there may not be another generation of cellular technology because there may not be sufficient economic justification for the development of a completely new standard. Instead, we might expect to see enhancements to all the different standards making up the complete communications network, with perhaps some new niche standards emerging in areas such as ultra-short range communications. This has not stopped countries and organizations from claiming that they already have a 4G technology, or like the Japanese, that they are working on one. Of course, since there is no widely agreed definition of what comprises a new generation, anyone is free to claim that their technology meets whatever criteria they regard as important, and with the definition of 3G already confused, perhaps we should expect 4G to be even more opaque!

Despite this controversy next generation networks will probably have, among other things, the following characteristics:

**Performance and bandwidth.** 4G systems are intended to provide high quality video services providing data transfer speeds of 100 Mbit/s (peak rate in mobile environment) to 1 Gbit/s (fixed indoors) with a 5 MHz frequency bandwidth. Such bandwidth enabling high bandwidth services within the reach of local area network (LAN) hotspots, installed in airports, homes, coffee shops and offices. The system capacity is expected to be at least 10 times larger than current 3G systems. In addition, these objectives should be met together with a drastic reduction in the cost (1/10 to 1/100 per bit) [see [71], for example].

**Interoperability.** The existence of multiple standards for 3G made it difficult to roam and incorporate across networks. There is therefore a need for a global standard providing global mobility and service portability so that the single-system vendors of proprietary equipment do not bind the customers.

**Technology.** Rather than being an entirely new standard, 4G will basically resemble a conglomeration of existing technologies and will be a convergence of more than one technology.
OFDMA (Orthogonal Frequency Division Multiple Access) is considered to be the leading radio technology for the upcoming 4G cellular infrastructure. OFDMA is a multi-user version of the popular OFDM (Orthogonal Frequency Division Multiplexing) digital modulation scheme. Multiple access is achieved in OFDMA by assigning subsets of subcarriers (subchannels or slots) to individual users. This allows a multiple simultaneous low data rate transmissions to several users.

1.4 Organization of this thesis

The first set of problems we are studying in this thesis is the set of cell planning problems, discussed in Chapter 2. Optimal planning of cellular networks is an important ingredient in the effort to provide advanced cellular services at a reasonable cost. Cell planning includes planning a network of base stations to provide the required coverage of the service area with respect to current and future traffic requirements, available capacities, interference, and the desired QoS. Cell planning for future networks requires new approaches, combining appropriate modeling of the technologies dependent characterizations and novel algorithm techniques.

We provide such an approach; we rigorously model important aspects of the future networks, define the budget limited cell planning problem and the minimum-cost cell planning problem, and provide the first approximation algorithms for these problems. We then show how the theoretical results can be used to derive practical planning of cellular networks. Our results indicate that a theoretical study together with optimized practical implementations outperform previous commonly used techniques, and achieve a close-to-optimal planning of real cellular networks.

From a theoretical point of view, we generalize the well-known studies on the budgeted maximum coverage problem [44] and the hard-capacitated set cover problem [21] in the sense that our problems contains, among others, capacities as well as non-uniform demands.

Preliminary versions of the results presented in this chapter appear in [12, 13], and in [10].

Next, in Chapter 3, we study a challenging problem that arises in the design of new topologies for radio access network for future cellular networks. Since the forthcoming cellular systems will provide broadband wireless access to a variety of advanced data and voice services, networks will have a significantly larger number of base stations and a much higher bandwidth demand from their radio access networks. This will motivate operators to replace the commonly used star based architecture, in which a Radio Network Controller (RNC) is connected to a set of base stations via direct links, with a more complex tree structure, in which a base station can be connected to an RNC via other base stations.

We address algorithmic aspects of this challenging design problem, in which tree-topology is used to connect base stations and RNCs. We formulate the problem as an optimization problem and prove that it is NP-hard to approximate it in the general case. For the metric case, however, we develop an $O(\log n)$-approximation algorithm, and we study the performance of this algorithm and several other heuristics in practical scenarios. Our results indicate that a combination of a certain greedy heuristic and the proven approximation algo-
algorithm, generates a solution that produces close to optimal results in practical scenarios and can be efficiently computed for sufficiently large network sizes.

A preliminary version of the results presented in this chapter appear in [14].

Motivated by algorithmic characterizations of the new OFDMA multiple access technique standard (as defined by the IEEE 802.16e-2005), in Chapter 4 of this thesis we design new approach for the cell selection mechanism. Cell selection is the process of determining the cell(s) that provide service to each mobile station. Optimizing these processes is an important step towards maximizing the utilization of current and future cellular networks. In this paper we study the potential benefit of global cell selection versus the current local mobile SNR-based decision protocol. In particular, we study the new possibility that is feasible in OFDMA-based systems, of satisfying the minimal demand of a mobile station simultaneously by more than one base station.

We formalize the problem as an optimization problem, called the all-or-nothing demand maximization problem, and show that when the demand of a single mobile station can exceed the capacity of a base station, this problem is not only NP-hard but also cannot be approximated within any reasonable factor. In contrast, under the very practical assumption that the maximum required bandwidth of a single mobile station is at most an \( r \)-fraction of the capacity of a base station, we present two different algorithms for cell selection. The first algorithm guarantees a satisfaction of at least a \( 1 - r \) fraction of an optimal assignment, where a mobile station can be covered simultaneously by more than one base station. The second algorithm guarantees a satisfaction of at least a \( \frac{1-r}{2-r} \) fraction of an optimal assignment, while every mobile station is covered by at most one base station. Using an extensive simulation study we show that the cell selections determined by our algorithms achieve a better utilization of high-loaded capacity-constrained future networks than the current SNR-based scheme. Specifically, our algorithms are shown to obtain up to 20% better usage of the network’s capacity, in comparison with the current cell selection algorithms.

Preliminary versions of the results presented in this chapter appear in [8, 9].

Finally in Chapter 5 we summarize and short discuss our results.
This page was intentionally left blank
Chapter 2

Cell Planning of Future Cellular Networks

2.1 Introduction

Cell planning is an important step in the design, deployment, and management of cellular networks. Cell planning includes planning a network of base stations that provides a (full or partial) coverage of the service area with respect to current and future traffic requirements, available capacities, interference, and the desired QoS. Cell planning is employed not only when a new network is built or when a modification to a current network is made, but also (and mainly) when there are changes in the traffic demand, even within a small local area (e.g., building a new mall in the neighborhood or opening new highways). Cell planning that is able to respond to local traffic changes and/or to make use of advance technological features at the planning stage is essential for cost effective design of future systems.

As described earlier (page 8) future systems will be designed to offer very high bit rates with a high frequency bandwidth. Such high frequencies yield a very strong signal degradation and suffer from significant diffraction resulting from small obstacles, hence forcing the reduction of cell size (in order to decrease the amount of degradation and to increase coverage), resulting in a significantly larger number of cells compared to previous generations. Future systems will have cells of different sizes: picocells (e.g., an in-building small base station with antenna on the ceiling), microcells (e.g., urban street, up to 1km long with base stations above rooftops at 25m height), and macrocells (e.g., non-line-of-sight urban macro-cellular environment). Each such cell is expected to service users with different mobility patterns, possibly via different radio technologies. Picocells can serve slow mobility users with relatively high traffic demands. They can provide high capacity coverage with hot-spot areas coverage producing local solutions for these areas. Even though these cells do not have a big RF impact on other parts of the network, they should be taken into consideration during the cell planning stage since covering hot-spot areas may change the traffic distribution. At the same time, microcells and macrocells can be used to serve users with high mobility patterns (highway users) and to cover larger areas. Hence, it is important to be able to choose appropriate locations for potential base stations and to consider different radio technologies, in
order to achieve maximum coverage (with low interference) at a minimum cost.

The increased number of base stations, and the variable bandwidth demand of mobile stations, will force operators to optimize the way the capacity of a base station is utilized. Unlike in previous generations, the ability of a base station to successfully satisfy the service demand of all its mobile stations would be highly limited and will mostly depend on its infrastructure restrictions, as well as on the service distribution of its mobile stations. To the best of our knowledge, no cell planning approach, known today, is taking the base station capacity into account.

Base stations and mobile terminals are expected to make extensive use of adaptive antennas and smart antennas. In case the system will have the ability to distinguish between different users (by their geographic positions), adaptive antennas will point a narrow lobe to each user, reducing interference while at the same time, maintaining high capacity. Smart antenna systems combine an antenna array with a digital signal-processing capability, enabling base stations to transmit and receive in an adaptive, spatially sensitive manner. In other words, such a system can automatically change the directionality of its radiation patterns in response to its signal environment. This can dramatically increase the performance characteristics (such as capacity) of a wireless system. Hence, future methods for cell planning should be able to include a deployment of smart antennas and adaptive antennas in their optimization process. Note that current advanced tools for cell planning already contains capabilities for electrical modifications of tilt and azimuth.

This chapter rigorously studies algorithmic aspects of cell planning problems, incorporates the anticipated future technologies into the cell planning, and presents new methods for solving these problems. Our new techniques are based on novel modeling of technologies dependent characterizations and approximation algorithms that provide provable good solutions. In addition, methods presented in this chapter are also applicable to current networks and various radio technologies.

2.1.1 Formulation and background

Consider a set $I = \{1, 2, \ldots, m\}$ of possible configurations of base stations and a set $J = \{1, 2, \ldots, n\}$ of clients. Each base station configuration (abbreviated base stations) containing the geographical location, typical antenna pattern (as well as its adopted model for propagation), azimuth, tilt, height and any other relevant parameters of a base station antenna that together with the technology determine the coverage area of the antenna and the interference pattern (for example, two such configurations with the same parameters except the tilt will be considered as different). Each base station $i \in I$ has capacity $w_i$, installation cost $c_i$, and every client $j \in J$ has a demand $d_j$. The demand is allowed to be simultaneously satisfied by more than one base station. Each base station $i$ has a coverage area represented by a set $S_i \subseteq J$ of clients admissible to be covered (or satisfied) by it; this base station can satisfy at most $w_i$ demand units of the clients in $S_i$. Computing $S_i$, as preprocessing stage to

---

1 Notice that when planning cellular networks, the notion of “clients” sometimes means mobile-stations and sometimes it represents the total traffic demand created by a cluster of mobile-stations at a given location. In this chapter we support both forms of representations.
the cell planning itself, is based on the physical properties of the antenna, the power setting, terrain information, and the corresponding area-depending propagation models. One optional way to build the $S_i$’s collection is to list the set of clients who “see” base station $i$ as their “best” server (or either “best” or “secondary”). Such computations are usually done using simulators and are outside the scope of this chapter.

When a client belongs to the coverage area of more than one base station, interference between the servicing stations may occur. These interference are modeled by a penalty-based mechanism and may reduce the contribution of a base station to a client. We denote by $Q(i, j)$ the net contribution of base station $i$ to client $j$, for every $j \in J, i \in I$, after incorporating all relevant interference. We formulate cell planning problems using the abstract notation of $Q(i, j)$ to be independent of the adopted model of interference. Interference models are discussed in Section 2.1.2.

Using this formulation, we define two cell planning problems. The budgeted cell planning problem (BCPP) asks for a subset of base stations $I' \subseteq I$ whose cost does not exceed a given budget $B$ and the total number of fully satisfied clients is maximized. That is, an optimal solution to BCPP needs to maximize the number of clients for which $\sum_{i \in I} Q(i, j) \geq d_j$. The minimum-cost cell planning problem (CPP) is to find a subset $I' \subseteq I$ of minimum cost that satisfies at least $\gamma$ of the demands of all the clients, for a given constant $0 < \gamma \leq 1$.

Many special cases of both cell planning problems have been extensively studied in the literature. In most cases the problem is NP-hard and finding an optimal solution for it (when applied to real networks) is infeasible in a reasonable running time. Thus, much of the work deals with only a limited aspect of the cell planning problem, and different heuristics solutions. In this chapter we address this hardness using approximation algorithms.

### 2.1.2 How to model the interference?

Interference handling is an important issues in planning and management of cellular networks. Basically, interference is caused by simultaneous signal transmissions in different cells (inter-cell). In this section we overview interference in the forthcoming cellular systems and present a new approach of incorporating interference in cell planning.

In narrowband systems (e.g., IS-136, GSM), transmissions within a cell are restricted to separate narrowband channels. Furthermore, neighboring cells use different narrowband channels for user transmissions. This requires splitting of the total bandwidth and reduces the frequency reuse in the network. However, the network can now be simplified and approximated by a collection of point-to-point non-interfering links, and the physical-layer issues are essentially point-to-points ones. Notice that since the level of interference is kept minimal, the point-to-point links typically have high signal-to-interference-plus-noise ratios (SINRs).

Wideband systems designs have a contrasting strategy: all transmissions are spread over the entire bandwidth. The key feature of these systems is universal frequency reuse, that is, the same spectrum is used in every cell. However, simultaneous transmissions can now interfere with each other and links typically operate at low SINRs. Two of the systems that differ, by a design choice, in how the users’ signal are spread are CDMA and OFDM. The Code Division Multiple Access (CDMA) system is based on direct-sequence spread-spectrum.
Here, users’ information bits are coded at a very low rate and modulated by pseudonoise sequences. In such systems (e.g., IS-95), the simultaneous transmissions, intra-cell and inter-cell, cause interference. In the Orthogonal Frequency Division Multiplexing (OFDM) system, on the other hand, users’ information is spread by hopping in the time-frequency grid. Here, the transmissions within a cell is kept orthogonal but adjacent cells share the same bandwidth and inter-cell interference still exists. This system has the advantage of the full frequency reuse of CDMA while retaining the benefits of the narrowband system where there is no intra-cell interference (see Chapter 4 in [69]).

Interference is typically modeled, for cell planning proposes, by an interference matrix which represents the impact of any base station on other base stations, as a result of simultaneous coverage of the same area (see Appendix 6B in [17]). Next we generalized this behavior to also include the geographic position of this (simultaneous) coverage.

Let $P$ be an $m \times m \times n$ matrix of interference, where $p(i_1, i_2, j) \in [0, 1]$ represents the fraction of $i_1$’s service which client $j$ loses as a result of interference with $i_2$ (defining $p(i, i, j) = 0$ for every $i \in I$, $j \in J$, and $p(i, i', j) = 0$ for every $j \not\in S_i$)\(^2\). This means that the interference caused as a result of a coverage of a client by more than one base station depends on the geographical position of the related “client” (e.g., in-building coverage produces a different interference than a coverage on highways using the same set of base stations). As defined above, $Q(i, j)$ is the contribution of base station $i$ to client $j$, taking into account the interference from all relevant base stations. We describe here two general models for computing $Q(i, j)$.

Let $x_{ij}$ be the fraction of the capacity $w_i$ of base station $i$ that is supplied to client $j$. Recall that $I' \subseteq I$ is the set of base stations selected to be opened, the contribution of base station $i$ to client $j$ is defined to be

$$Q(i, j) = w_ix_{ij} \cdot \prod_{i' \neq i : i' \in I'} (1 - p(i, i', j)). \quad (2.1)$$

Notice that, as defined by the above model, it is possible that two distinct base stations, say $\alpha$ and $\beta$ interfere with each other “in” a place $j$ (i.e., $p(\alpha, \beta, j) > 0$) although $j \not\in S_\beta$. In general, each of these base stations “interferes” base station $i$ to service $j$ and reduces the contribution of $w_ix_{ij}$ by a factor of $p(i, i', j)$.

Since (2.1) is a high-order expression we use the following first-order approximation, while assuming that the $p$’s are relatively small,

$$\prod_{i' \in I'} (1 - p(i, i', j)) = (1 - p(i, i_1', j)) (1 - p(i, i_2', j)) \ldots \approx 1 - \sum_{i' \in I'} p(i, i', j). \quad (2.2)$$

Combining (2.1) and (2.2) we get

$$Q(i, j) \approx \begin{cases} w_ix_{ij}(1 - \sum_{i' \neq i \in I'} p(i, i', j)), & \sum_{i' \in I'} p(i, i', j) < 1, \\ 0, & \text{otherwise}. \end{cases} \quad (2.3)$$

\(^2\)For simplicity, we do not consider here interference of higher order. These can be further derived and extended from our model.
Consider, for example, a client $j$ belonging to the coverage areas of two base stations $i_1$ and $i_2$, and assume that just one of these base stations, say $i_1$, is actually participating in $j$’s satisfaction (i.e., $x_{i_1j} > 0$ but $x_{i_2j} = 0$). According to the above model, the mutual interference of $i_2$ on $i_1$’s contribution ($w_{1}x_{1j}$) should be considered, although $i_2$ is not involved in the coverage of client $j$.

In most cellular wireless technologies, this is the usual behavior of interference. However, in some cases a base station can affect the coverage of a client if and only if it is participating in its demand satisfaction. The contribution of base station $i$ to client $j$ in this case is defined by

$$Q(i, j) \approx \begin{cases} w_{i}x_{ij}(1 - \sum_{i' \neq i \in I_{j}} p(i, i')), & \sum_{i' \neq i \in I_{j}} p(i, i') < 1 \\ 0, & \text{otherwise.} \end{cases}$$

(2.4)

where $I_{j}$ is the set of base stations that participate in the coverage of client $j$, i.e., $I_{j} = \{i \in I : x_{ij} > 0\}$. Notice that in this model the interference function does not depend on the geographic position of the clients.

### 2.1.3 Our contributions

In this chapter we study two of the most important cell planning problems in the perspective of future cellular networks: BCPP and CPP. We combine a theoretical study of approximation algorithms for these problems with simulations study of the practical effectiveness of these algorithms when compared to current state of the art.

**The budgeted cell planning problem.** We present the first study of the BCPP. We show, in Section 2.3.1, that approximating BCPP is NP-hard. Then we define a restrictive version of BCPP, the $k4k$-budgeted cell planning, by making additional assumption that is valid in most practical scenarios. The additional property is that every set of $k$-opened base stations can fully satisfy at least $k$ clients, for every integral value of $k$. In Section 2.3.2 we show that this problem remains NP-hard and, through a rigorous study of the optimal solution, we present, in Section 2.3.6, an $e^{-1}(\approx 0.368)$ factor approximation algorithm for this problem.

From a theoretical point of view, our study contains two independent results on the client assignment problem (CAP) and the budgeted maximum assignment problem (BMAP). Given a set of base stations with its capacities, and a set of clients with its demands, the client assignment problem asks for the maximum number of clients that can be satisfied each by exactly one base station. Notice that in this problem we consider the set of base stations as (already) opened and no installation cost is specified. We show, in Section 2.3.4, that this problem is NP-hard and can be approximated within a factor of 1/2 of the optimum. The second problem, BMAP, is a generalization of CAP, and assume that every base station has an installation cost, and a general budget is also given. A $\frac{1}{2}(1 - \frac{1}{e})$-approximation algorithm is presented, in Section 2.3.5, for this problem. This algorithm generalizes the result of the well-known study on the budgeted maximum coverage problem [44] in the sense that the problem dealt in [44] does not include capacities nor non-uniform demands.

**The minimum-cost cell planning problem.** We present (Section 2.4) an $O(\log W)$-approximation algorithm for the non-interference (i.e., $P$ is the zero matrix) case of the CPP,
where $W$ is the largest capacity over all base stations selected for opening. To the best of our knowledge this is the first approximation algorithms for this special case of the problem. Theoretically speaking, our algorithm (Section 2.4) generalizes the well-known result on the hard-capacitated set cover problem [21] in the sense that the problem, in our case, contains capacities as well as non-uniform demands.

**Simulation results and practical aspects.** In order to verify that our algorithms for CPP perform well in practice, two different simulation sets were conducted with scenarios relevant to future technologies (Section 2.5). Each of these simulations have the objectives of minimization the total cost and minimization the total number of antennas (sectors). In both simulation sets our results indicate that the practical algorithms derived from our theoretical scheme can generate solutions that are very close to the optimal solutions and much better than the proved worst-case theoretical bounds. Moreover, our algorithms achieves a significant lower value of the solution cost than of the commonly used greedy approach [57, 70].

### 2.2 Previous work

Cell planning is one of the most studied problems in relation to cellular networks optimization. Previous works dealt with a wide variety of special cases (e.g., cell planning without interference, frequency planning, uncapacitated models, antenna-type limitations, and topological assumptions regarding coverage) and the objectives are mostly of minimum-cost type. The techniques used in these works range from meta-heuristics (e.g., genetic algorithms, simulated annealing, etc.) [7, 24, 41, 47, 49, 50, 52, 53, 55, 68, 75] and greedy approaches [2, 3, 56, 70, 78], through exponential-time algorithms that compute an optimal solution [48], to approximation algorithms for special cases of the problem [1, 30, 33, 48]. A comprehensive survey of various works on cell planning problems appears in [12] and a comparison between optimization methods for cell planning of 3G systems appears in Chapter 14 of [57]. Table 2.1 summarizes the approaches taken by several relatively recent papers in this area, and examines planning objectives, technological supported assumptions or constrains, applicability for 3G/4G, and if the worst-case performance of the solution suggested is rigorously analyzed and the distance from the optimal solution is provably bounded.

A solution for an uncapacitated version of CPP has been presented in [41]. In his chapter, Hurley use local improvements simulated annealing procedures to optimize multi-objective function (full coverage, minimum total cost of sites, lower bound on the total satisfied traffic demands, minimum interference level, and upper bound on the number of handovers permitted). However, base stations assumed to have unlimited capacity and no theoretical analysis or guarantee on the quality of the solutions is given.

A restricted approach for cell planning is described in a series of papers (e.g., [52, 55], and [53]). These papers are characterized by an integer programming formulation for various optimization problems: planning location for a given number of base stations, penalizing multiple coverage, minimizing interference between base stations, counting reused frequencies, minimizing the number of blocked channels, minimizing the number of blocked downlink connections, and maximizing the uniquely served traffic. Optimization of these objectives
is done via simulated annealing and branch-and-bound heuristics. Unfortunately most of solutions for these problems cannot be applied for B3G/4G technologies.

The problem of planning location for base stations (BSLP) is one of the most studied case of the CPP. In this problem clients are usually assumed to have unit demand and base stations assumed to have an installation cost and ability to cover any number of clients (from their coverage area). The objective is, in general, to minimize the total cost of opened base station in order to provide connectivity to all the clients. Local improvements procedure using a naive simulated annealing is used in \([68]\). In their work, no specific model is illustrated, and interference are not supported (although the work is addressing wideband cellular technologies). Genetic algorithms are used in \([47, 49]\), and \([75]\) to solve this problem. In these papers multiobjective optimization is performed, no interference are assumed \([47]\), and wideband cellular technologies are not supported \([49, 75]\). Tabu search approach is used in \([50]\) and \([7]\) for an integer programming formulation of this problem.

Greedy algorithms for combinatorial formulations of BSLP are presented in \([56, 70, 78]\). In these papers interference are not supported, no traffic demands are assumed (i.e., clients assumed to have unit demand), and no performance guarantee is analyzed (or claimed).

Approximation algorithms for BSLP are used in \([1, 30]\), and \([33]\). A PTAS for two combinatorial problems are presented in \([30]\) and \([33]\): Maximizing the number of totally satisfied mobile clients, and minimizing the number of installed base stations, while a bicriteria \(O(\log n, \log n)\)-approximation algorithm is described in \([1]\) for BSLP. In these papers many restricted assumptions are assumed: The model is limited to metric space only (which is not

### Table 2.1: Previous work on cell planning of cellular networks

<table>
<thead>
<tr>
<th>Parameters</th>
<th>([41])</th>
<th>([68])</th>
<th>([47])</th>
<th>([75])</th>
<th>([50])</th>
<th>([7])</th>
<th>([53])</th>
<th>([55])</th>
<th>([24])</th>
<th>([70])</th>
<th>([2])</th>
<th>([30])</th>
<th>([33])</th>
<th>([1])</th>
<th>([48])</th>
<th>our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location planning for base stations</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Frequency allocation plan</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Planning under budget limitation</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>Planning using exists infrastructure</td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Supporting some interference model</td>
<td>+ ±</td>
<td>+ ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>+ ±</td>
<td></td>
</tr>
<tr>
<td>Supporting user-defined impact matrix</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Handling capacities of base stations</td>
<td>+ ±</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Traffic maps supported model</td>
<td>+ ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td></td>
</tr>
<tr>
<td>Antenna patterns availability</td>
<td>+ ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Flexible topological assumptions</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td>+ +</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Smart antennas friendly</td>
<td>+ ±</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Responding to traffic changes</td>
<td>+ ±</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3G friendly</td>
<td>+ +</td>
<td>+ ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td>± ±</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4G friendly</td>
<td>+ ±</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Worst-case analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Technion - Computer Science Department - Ph.D. Thesis PHD-2008-02 - 2008
necessary true in practice), no traffic demands are supported, and base stations are assumed to have an unlimited capacity of coverage. In addition, in [30] and [33] it is assumed that only omnidirectional pattern antennas are used, and no location of base stations is allowed below a minimal distance between these base stations (which is also not practical as in the case of installing several antennas on one site).

Exact $2^{O(\sqrt{m \log m})}$-time algorithm is described in [48] for the base station location problem. No interference are assumed by this somewhat efficient exponential algorithm, while topological restrictions were also adopted (geographical distances are lie on an Euclidean space and omnidirectional patterns of the same radius are assumed for all antennas).

Base station location problem together with frequency planning is addressed in [24]. This problem is solved via Lagrangian relaxation based heuristic, an integer programming formulation and then performing local improvements heuristics. No traffic demands are assumed in this work, interference are via reuse only, and only frequency assignment-based cellular technologies are supported.

The main contribution of this chapter, as indicated in Table 2.1, is that we provide provable performance algorithms in a model that captures many of the important aspects of future cellular networks. Moreover, as shown in Section 2.5, these algorithms also performs well over real data.

2.3 The budgeted cell planning problem

In this section we study the problem of cell planning under budget constraint. To the best of our knowledge, despite the extensive research of non-budgeted cell planning problems (i.e., CPP; See for example Table 2.1 on page 19), there is no explicit study in the literature of the BCPP (in both theoretical and, surprisingly, also in practical settings).

BCPP is closely related to the well-known budgeted maximum coverage problem. Given is a budget $B$ and a collection of subsets $S$ of a universe $U$ of elements, where each element in $U$ has a specified weight and each subset has a specified cost. The budgeted maximum coverage problem asks for a subcollection $S' \subseteq S$ of sets, whose total cost is at most $B$, such that the total weight of elements covered by $S'$ is maximized. This problem is the “budgeted” version of the set cover problem in which one wishes to cover all the elements of $U$ using a minimum number of subsets of $S$. The budgeted maximum coverage problem is a special case of BCPP in which elements are clients with unit demand, every set $i \in I$ corresponds to a base station $i$ containing all clients in its coverage area $S_i \subseteq J$, and $w_i = |S_i|$ for all base stations in $I$. In this setting, budgeted maximum coverage is the case (in the sense that a solution for BCPP is optimal if and only if it is optimal for the budgeted maximum coverage) when there are no interference (i.e., $P$ is the zero matrix). For the budgeted maximum coverage problem, there is a $(1 - \frac{1}{e})$-approximation algorithm [5, 44], and this is the best approximation ratio possible unless NP = P [27, 44]. Interestingly enough, we show in the next section that our generalization makes this problem hard to approximate.

BCPP is also closely related to the budgeted unique coverage version of set cover. In the budgeted unique coverage problem elements in the universe are uniquely covered, i.e., appear
in exactly one set of $S'$. As with the budgeted maximum coverage problem, this problem is a special cases of BCPP. In this setting, budgeted unique coverage is when the interference is taking to be the highest (i.e., the impact matrix is a matrix of 1’s, except the diagonal where elements are equals to zero). For the budgeted unique coverage problem, there is an $\Omega(1/\log n)$-approximation algorithm \cite{23} and, up to a constant exponent depending on $\epsilon$, $O(1/\log n)$ is the best possible ratio assuming NP $\not\subseteq$ BPTIME $(2^n)$ for some $\epsilon > 0$.

Let $U = \{1, \ldots, n\}$, let $c_u, u \in U$ be a set of nonnegative weights, and let $B$ be a nonnegative budget. Another closely related problem to BCPP is the problem of maximizing nondecreasing submodular set function with budget constraint, namely,

$$\max_{S \subseteq U} \left\{ f(S) : \sum_{u \in S} c_u \leq B \right\},$$

where $f(S)$ is a nonnegative nondecreasing submodular polynomially computable set function (a set function is submodular if $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$ for all $S, T \subseteq U$ and nondecreasing if $f(S) \leq f(T)$ for all $S \subseteq T$). For this problem, there is a $(1-\frac{1}{e})$-approximation algorithm \cite{66}, and as this problem is a generalization of the budgeted maximum coverage ($c_u = 1$, for all $u \in U$, and $f(S)$ denotes the maximum weight that can be covered by the set $S$), this ratio is the best achievable. Although this problem seems, at least from a natural perspective, to be closely related to BCPP, observe that set (covering) functions are not submodular, in general, when interference are involved. Consider, for example, an instance of BCPP in which $I = \{1, 2, 3\}$ with $w_1 = w_2 = 1$ and $w_3 = 1/4$, a single client with $d = 2$ that can be satisfied by all base stations, and symmetric interference $p(1, 3) = p(2, 3) = 1/2$, while $p(1, 2) = 0$. Taking $S = \{1\} \cup \{3\}$ and $T = \{2\} \cup \{3\}$ we have $f(S) + f(T) \neq f(S \cup T) + f(S \cap T)$, where $f(S)$ is defined to be the maximum number of fully satisfied clients that can be covered by the set $S$ of base stations.

### 2.3.1 Inapproximability

As mentioned earlier, the budgeted maximum coverage problem is a special case of BCPP. Khuller, Moss, and Naor \cite{44} showed, in 1999, that the greedy method of picking at each step the most effective set until either no element is left to be covered or the budget limitation is exceeded, results, together with a combination of the enumeration technique, in a $(1-\frac{1}{e})$-approximation algorithm for this problem. Unfortunately, a natural attempt to adopt the ideas from \cite{44} to the more general setting of BCPP fails, as stated by the next theorem.

**Theorem 2.3.1.** It is NP-hard to find a feasible solution to the budgeted cell planning problem.

**Proof.** The proof is via a reduction from the subset sum problem. Given an instance of the subset sum problem, i.e., a set of natural numbers $A = \{a_1, a_2, \ldots, a_n\}$ and an additional natural number $T = \frac{1}{2} \sum_{i=1}^{n} a_i$. We build an instance of BCPP with $I = \{1, 2, \ldots, n\}, |J| = 1$ and $w_i = c_i = a_i$ for every $i \in I$; the budget and the single client’s demand are $B = d = T$ and no interference are assumed.
It is easy to see that the client is satisfied if and only if there exists $S \subseteq A$ with $\sum_{i \in S} a_i = T$. Since there is only a single client, any polynomial-time approximation algorithm must produce a full coverage, solving the subset sum problem in polynomial time. 

2.3.2 The $k4k$-budgeted cell planning problem

In light of the above inapproximability result, we turn to define a restrictive version of BCPP which is general enough to cover all interesting practical cases. In order to do that, we use the fact that in general, the number of base stations in cellular networks is much smaller than the number of clients. Notice that when planning cellular networks, the notion of “clients” sometimes means mobile-clients and sometimes it represents the total traffic demand created by many mobile-clients at a given location. Our models support both forms of representations. Moreover, when there is a relatively large cluster of antennas in a given location, this cluster is usually addressed to meet the traffic requirements of a high-density area of clients. Thus for both interpretations of “clients” the number of satisfied clients is always much bigger than the number of base stations. Followed by the above discussion, we define the $k4k$-budgeted cell planning problem ($k4k$-BCPP) to be BCPP with the additional property that every set of $k$ base stations can fully satisfy at least $k$ clients, for every integer $k$ (and we refer to this property as “$k4k$ property”). First, we show that this problem remains hard.

**Theorem 2.3.2.** The $k4k$-budgeted cell planning problem is NP-hard.

**Proof.** Via a reduction from the budgeted maximum coverage problem. Consider an instance of the budgeted maximum coverage problem, that is, a collection of subsets $S = \{S_1, \ldots, S_m\}$ with associated costs $\{c_i\}_{i=1}^m$ over a domain of elements $X = \{x_1, \ldots, x_n\}$, and a budget $L$.

We can construct an instance of $k4k$-BCPP such that an optimal solution to this problem gives an optimal solution to the budgeted maximum coverage problem. First, we construct a bipartite graph of elements vs. sets, derived from the budgeted maximum coverage instance: there is an edge $(x_i, S_j)$ if and only if element $x_i$ belongs to the set $S_j$. The instance of $k4k$-BCPP is as follows: the set of clients is $\{x_1, \ldots, x_n\} \cup \{y_1, \ldots, y_m\}$, where each of the $x_i$'s has a unit demand and each of the $y_r$’s has zero demand. The set of potential base stations is $\{S_1, \ldots, S_m\}$, each with opening cost $c_i$ and capacity $w_i = |S_i|$. The admissible clients of base station $S_i$, are the elements of $S_i$ and all the $y$’s, the budget is $B = L$, and there is no interference.

Clearly, a solution to $k4k$-BCPP is optimal if and only if the corresponding solution of the budgeted maximum coverage instance is optimal. 

In the remainder of this section we assume that the interference model is the one defined in Equation (2.4).

2.3.3 The structure of BCPP solutions

Our algorithm is based on a combinatorial characterization of the solution set to BCPP (and in particular to $k4k$-BCPP). The following lemma is a key component in the analysis of our
approximation algorithm.\textsuperscript{3}

\textbf{Lemma 2.3.3.} Every solution to the \(k4k\)-budgeted cell planning problem can be transformed to a solution in which the number of clients that are covered by more than one base station is at most the number of opened base stations. Moreover, this transformation leaves the number of fully satisfied clients as well as the solution cost unchanged.

\textit{Proof.} Consider a solution \(\Delta = \{I', J', x\}\) to the \(k4k\)-BCPP, where \(I' \subseteq I\) is the set of base stations selected for opening, \(J' \subseteq J\) is the set of fully satisfied clients, \(x_{ij}\)'s are the base station-client coverage rates, and \(J'' \subseteq J'\) is the set of clients that are satisfied by more than one base station. Without loss of generality we may assume that every client has a demand greater than zero, since there is no need for “covering” clients with zero demand.

We associate the weighted bipartite graph \(G_{\Delta} = (I' \cup J', E)\) with every such solution. In this graph, \((i, j) \in E\) has weight \(w(i, j) = w_i x_{ij}\) if and only if \(x_{ij} > 0\), and \(w(i, j) = 0\), otherwise.

Two cases need to be considered:

1. If \(G_{\Delta}\) is acyclic then we are done (i.e., no transformation is needed); in this case \(|J''| < |I'|\). To see this, let \(T\) be a forest obtained from \(G_{\Delta}\) by fixing an arbitrary base station vertex as the root (in each of the connected components of \(G_{\Delta}\)) and trimming all client leaves. These leaves correspond to clients who are covered, in the solution, by a single base station. Since the distance, from the root, to every leaf of each tree is even, the number of internal client-vertices is at most the number of base station-vertices, hence \(|J''| < |I'|\).

2. Otherwise, we transform \(G_{\Delta} = (I' \cup J', E)\) into an acyclic bipartite graph \(G_{\Delta'} = (I' \cup J', E')\) using a cycle canceling algorithm. For simplicity, we first describe the following algorithm for the no interference case.

\textbf{Algorithm A \textit{[cycle canceling without interference].}} As long as there are cycles in \(G_{\Delta}\), pick a cycle \(C\) and let \(\sigma\) be the weight of a minimum-weight edge on this cycle. Take a minimum-weight edge on \(C\) and, starting from this edge, alternately, in clockwise order along the cycle, decrease and increase the weight of every edge by \(\sigma\).

It is easy to verify that at the end of the algorithm every client receives, and every base station supplies, the same amount of demand units as before. Moreover, the only changes here are the values of the \(x_{ij}\)'s. Hence, Algorithm A preserves the number as well as the identity of the satisfied clients. Since at each iteration at least one edge is removed, \(G_{\Delta'}\) is acyclic, thus yielding \(|J''| < |I'|\) as in the former case.

When interference exist, using Algorithm A to remains the number (as well as the identity) of the satisfied clients unchanged is possible only when a modification of the \(x_{ij}\)'s does not affect the \(Q(i, j)\) of any client. Otherwise, this algorithm can no longer guarantee that the number of the satisfied clients will remains the same.

\textsuperscript{3}This Lemma is also true for non-\(k4k\) versions of the BCPP.
To overcome this problem we generalize the method of cycle canceling. Consider a cycle $C = (v_1, \ldots, v_k = v_1)$ in $G_\Delta$, such that odd vertices correspond to base stations. Let $v_i$ be any client-vertex in $C$. Now suppose the base station which corresponds to $v_{i-1}$ increases its supply to $v_i$ by $\alpha$ units of demand. The basic idea of the generalization is to compute the exact number of demand units the base station which corresponds to $v_{i+1}$ must subtract from its coverage, in order to preserve the satisfaction of that client, taking into account all the demand (with its interference) supplied by base station vertices which are outside the cycle.

Notice that increasing a certain $w(v_i, v_{i+1})$ does not necessary increase the supply to client $v_i$. When interference are considered, it could actually happen that increasing $w(v_i, v_{i+1})$ decreases the supply to $v_i$ (if the new interference penalties outweigh the increased supply). Similarly, decreasing some $w(v_i, v_{i+1})$ could actually increase the supply to $v_i$. However, one can assume for optimal solutions that these cases do not occur (as the solution could be transformed into an equivalent solution where such edges have $w(v_i, v_{i+1}) = 0$).

To demonstrate the idea of canceling cycles when interference exist let us assume, for simplicity, that there is only a single base station which is not on the cycle, denoted by $v_o$, which participates in the coverage of client $v_i$. Then, the total contribution of base stations $v_{i-1}, v_{i+1},$ and $v_o$ to the coverage of client $v_i$ is, by (2.1),

$$\delta(v_i) = Q(v_0, v_i) + Q(v_{i+1}, v_i) + Q(v_{i-1}, v_i).$$

Given that the supply of base station $v_{i-1}$ to client $v_i$ is increased by $\alpha$ units of demand (i.e., $w'(v_{i-1}, v_i) = w(v_{i-1}, v_i) + \alpha$, where $w'$ is the updated weight function of the edges), base station $v_{i+1}$ must decrease its supply to this client by $\beta$ units of demand (i.e., $w'(v_{i+1}, v_i) = w(v_{i+1}, v_i) - \beta$) in order to preserve the satisfaction of client $v_i$ (assuming $v_o$’s supply remains the same). Then, the value of $\beta$ can be computed via a solution to the following equation (in variable $\beta$),

$$\delta'(v_i) = Q'(v_0, v_i) + Q'(v_{i+1}, v_i) + Q'(v_{i-1}, v_i) = \delta(v_i).$$

Notice that our cycle canceling algorithms are used for the proof of existence and such computations are not necessary for the execution of our approximation algorithm.

**Algorithm B [cycle canceling with interference].** As long as there are cycles in $G_\Delta$, pick a cycle $C = (v_1, \ldots, v_k = v_1)$ where odd vertices represent base stations. As before, every edge $e_i = (v_i, v_{i+1})$ on the cycle has a weight $w(v_i, v_{i+1})$ associated with it, representing the amount of demand supplied by the base-station-vertex in $e_i$ to the client-vertex in $e_i$. For simplicity, let $d'_i$ denote this value.

We recursively define a sequence of weights $\{y_i\}_{i=1}^{k-1}$, with alternating signs which represent a shift in the demand supply of base stations to clients along
the cycle. Start by setting $y_1 = \epsilon$, representing an increase of $\epsilon$ to the demand supplied by the base-station-vertex $v_1$ to the client-vertex $v_2$. This increase of supply may not all be credited to vertex $v_2$ due to interference, some of which are possibly due to base stations which are outside the cycle, which contribute to $v_2$’s satisfaction. Set $y_2$ to be the maximal decrease in the demand supplied by base-station-vertex $v_3$ to client-vertex $v_2$, such that $v_2$ remains satisfied. This amount of demand is now “available” to base-station-vertex $v_3$, hence we allow $v_3$ to supply this surplus to client-vertex $v_4$. We continue in this manner along the cycle.

If, by repeating this procedure, we end up with $|y_{k-1}| \geq \epsilon$, then we say the cycle is $\epsilon$-adjustable. Otherwise, redefine the values of $\{y_i\}_{i=1}^{k-1}$ in a similar manner, but in reverse order, i.e., starting from $y_{k-1}$ and ending with $y_1$. However, it is easy to verify that at least one direction, the cycle is $\epsilon$-adjustable, for some value of $\epsilon$.

Let $\epsilon_{\text{max}}$ be the largest value for which the cycle is adjustable, and consider its corresponding values of $y_i$, $i = 1, \ldots, k-1$. Note that the $y_i$’s have alternating signs, and for any client-vertex $v_i$, $y_i = -y_{i-1}$. Define the quotients $z_i = d'_i/y_i$ for every $i = 1, \ldots, k-1$, and let $z_{\text{min}} = \min_{z_i < 0} |z_i|$. Now increase the amount of demand supplied on every edge on the cycle to be $w'(v_i, v_{i+1}) = y_i \cdot z_{\text{min}}$, where $w'$ is the updated weight function of the edges, as before.

Two important invariants are maintained throughout our cycle-canceling procedure. The first is that $w'(i, j) \geq 0$ for every edge $(i, j)$ of the cycle. The second is that there exists at least one edge $e = (i, j)$ on the cycle for which $w'(i, j) = 0$. Therefore Algorithm B preserves the number and the identity of the satisfied clients and $G_{\Delta'}$ is also a solution. Since at each iteration at least one edge is removed, $G_{\Delta'}$ is acyclic and $|J''| < |I'|$ as before.

\[ \square \]

**An \( \frac{\epsilon-1}{3\epsilon-1} \)-approximation algorithm**

The main difference between $k4k$-BCPP and other well-studied optimization problems is the existence of interferences. In order to overcome the difficulties caused by interferences, we use Lemma 2.3.3. We distinguish between two cases according to whether or not the optimal solution has the following property: there are many clients that are covered by more than one base station. If the optimal solution has this property, then by opening the maximum number of base stations and applying the $k4k$ property, we get a good approximation. Otherwise, we reduce the problem to the problem of finding a feasible set of base stations such that the number of clients that can be covered, each by exactly one base station, is maximized. Although this problem is still NP-hard, we show how to approximate it using the greedy approach and ideas similar to the ideas of [44].
2.3.4 The client assignment problem

Prior to using the greedy approach to solve the $k4k$-BCPP one must answer the question: how many clients can be covered by a set $S$ of opened base stations, and how many more can be covered if another base station $i$ is to be opened next? Formally, for a given set of base stations, $I'$, let $N(I')$ be the number of clients that can be satisfied, each by exactly one base station (we assume no interference, or interference of the second kind). We refer to the problem of computing $N(\cdot)$ as the Client Assignment Problem (CAP).

**Lemma 2.3.4.** The function $N(\cdot)$ is not submodular.

*Proof.* Consider the following example: $I = \{1, 2, 3\}$, $J = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $S_1 = J$, $S_2 = \{1, 2, 3\}$, $S_3 = \{4, 5, 6\}$. The demands are: $d_1 = d_2 = d_3 = d_4 = d_5 = d_6 = 4$, $d_7 = 3$, $d_8 = d_9 = d_{10} = 9$, and the capacities are: $w_1 = 30$, $w_2 = w_3 = 12$.

Let $S = \{1, 2\}$ and $T = \{1, 3\}$. One can verify that $N(S) = N(T) = 8$, $N(S \cap T) = 7$ and $N(S \cup T) = 10$. □

**Theorem 2.3.5.** CAP is NP-hard.

*Proof.* We reduce the PARTITION problem to CAP. Given is a set of positive integers $\{a_1, \ldots, a_n\}$ whose sum is $T$, where $T$ is even, we construct the following instance of CAP: $I = \{i_1, i_2\}$, $J = \{1, \ldots, n, n + 1, n + 2\}$, $w_1 = w_2 = \frac{T + 1}{2}$, $S_1 = J \setminus \{n + 2\}$, $S_2 = J \setminus \{n + 1\}$, and for $j \leq n$, $d_j = a_j$, and $d_{n+1} = d_{n+2} = 0.5$. It is easy to verify that there exists a partition of $\{a_1, \ldots, a_n\}$ into two subsets whose sum is $T/2$ if and only if it is possible to cover all the clients in the above CAP instance. □

Lemma 2.3.4 and Theorem 2.3.5 indicate the difficulty in applying the greedy approach to solve $k4k$-BCPP. Informally speaking, submodularity guarantees that a greedy choice that is made at one step stays the greedy choice even when taking into account the later steps. Without submodularity it is not clear whether greediness is the right approach. Moreover, Theorem 2.3.5 implies that we cannot efficiently compute the best (in the greedy sense) base station to open in a single step. These two difficulties prevent us from using the generalization of [44] proposed by Sviridenko [66] to approximate $k4k$-BCPP, as the algorithm of Sviridenko can be used to approximate only submodular polynomially computable functions.

In order to overcome these problems, we use an approximation for CAP. We note that CAP is a special case of the well studied General Assignment Problem (GAP), which can be approximated up to a constant factor (see [29] for a $\frac{3}{2} - \frac{1}{e}$-approximation). Nevertheless, for our application we will need an approximation algorithm with submodularity-like properties, and hence we use the following simple greedy algorithm. The algorithm gets as an input an ordered set of base stations $I' = \{i_1, \ldots, i_k\}$.

Denote by $N_A(I')$ the number of clients covered by Algorithm 2.1 when it is given $I'$ as an input. In what follows we use $(I', i)$ (for $i \notin I'$) to denote an input to Algorithm 2.1 in which base station $i$ is ordered last, that is, after all the base stations in $I'$. We also denote by $BS_{I'}(j)$ the base station that covers client $j$ when Algorithm 2.1 gets $I'$ as its input (let $BS_{I'}(j) = \emptyset$ when client $j$ is not covered).
Algorithm 2.1 GREEDY APPROXIMATION FOR CAP
1: for all clients in a non-decreasing order of their demand do
2: Let $j$ be the current client.
3: Find the first base station in the given order that can cover $j$.
4: if it exists then
5: Assign $j$ to this base station.
6: else all base stations cannot cover $j$ due to capacity constraints
7: Leave client $j$ uncovered.
8: end if
9: end for

Theorem 2.3.6. Algorithm 2.1 is a $\frac{1}{2}$-approximation to CAP, that is, for every $I'$, $N_A(I') \geq \frac{N(I')}{2}$.

Proof. Let $OPT$ denote the optimal solution. Fix a base station $i$, and let $J_{OPT}$ denote the set of clients covered by $i$ in $OPT$, but are not covered at all by Algorithm 2.1. Order the clients in $J_{OPT}$ in a non-decreasing order of their demand, and let $j_1, j_2, \ldots, j_h$ be this order. Denote the clients that Algorithm 2.1 chooses to cover using base station $i$ by $j'_1, j'_2, \ldots, j'_h$ (where these are ordered according to the order they are covered by the algorithm). Notice that $d_{j'_h} \leq d_{j_1}$ and $d_{j_1} + \sum_{\ell=1}^{h} d_{j'_\ell} > w_i$, since otherwise the algorithm would cover $j_1$ using $i$.

For each client $j_{t} \in J_{OPT}$, let

$$z(j_{t}) = \min \left\{ \ell \left| \sum_{r=1}^{\ell} d_{j'_{r}} + \sum_{r=1}^{t} d_{j_{r}} > w_i \right. \right\}.$$

One can think of $z(j_r)$ as the reason that $j_r$ is not covered by $i$. Feasibility of $OPT$ implies that $\sum_{r=1}^{k} d_{j_{r}} \leq w_i$, and hence $z(j_{r})$ is well-defined. Moreover, since $d_{j'_{h}} \leq d_{j_{1}}$, for every $j_{t}$ and $j_{s}$, $r \neq s$, we have that $z(j_{t}) \neq z(j_{s})$. This implies that $k \leq h$, and the theorem follows.

We need the following two properties of Algorithm 2.1.

Lemma 2.3.7 (Monotonicity of Algorithm 2.1). For every set of base stations $I'$ and every base station $i \not\in I'$, $N_A(I' \cup \{i\}) \geq N_A(I')$.

Proof. Assume to the contrary that $N_A(I' \cup \{i\}) < N_A(I')$ and consider the first client $j$ during the execution with the input $I' \cup \{i\}$ that is not covered by $BS_{I'}(j)$ or by a base station that precedes $BS_{I'}(j)$ in the given order of base stations. During the execution with the input $I'$, base station $BS_{I'}(j)$ had at least $d_{j}$ unused capacity when Algorithm 2.1 considered client $j$. During the execution with the input $I' \cup \{i\}$, base station $BS_{I'}(j)$ had at most $d_{j} - 1$ unused capacity when Algorithm 2.1 considered client $j$. Hence, there exists at least one client $j'$ with demand at most $d_{j}$ that is covered by $BS_{I'}(j)$ during the execution with the input $I' \cup \{i\}$ and that is not covered by $BS_{I'}(j)$ during the execution with the input $I'$. Since $j'$ is considered by the algorithm before client $j$, it could be covered by $BS_{I'}(j)$ during the execution with $I' \cup \{i\}$.
Algorithm 2.2 BUDGETED MAXIMUM ASSIGNMENT

1: For every ordered $I' \subseteq I$, $c(I') \leq B$ and $|I'| < 3$, compute $N_A(I')$. Let $I_1$ be the subset with the highest value of $N_A$ computed.
2: for every ordered $I' \subset I$, $c(I') \leq B$ and $|I'| = 3$ do
3: \begin{verbatim}
    U ← I \ I'
    repeat
    Select $i \in U$ such that maximizes $\frac{N_A(I', i) - N_A(I')}{c_i}$.
    if $c(I') + c_i \leq B$ then
    I' ← (I', i)
    end if
    U ← U \ {i},
    until $U = \emptyset$
4: end for
5: if $N_A(I') > N_A(I_1)$ then
6: I_1 ← I'
7: end if
8: Output $I_1$.

the input $I'$ (we have at least $d_j$ units of capacity available that are later used to cover $j$). Hence we conclude that that $BS_{I'}(j')$ precedes $BS_{I'}(j)$, which is a contradiction to the way we chose $j$. \hfill \Box

Lemma 2.3.8. For every set of base stations $I'$ and every base stations $i_1, i_2 \notin I'$,

$$N_A(I', i_1, i_2) - N_A(I', i_1) \leq N_A(I', i_2) - N_A(I').$$

Proof. Note that since $i_2$ is the last base station in the order, it cannot change the clients that are covered by the rest of the base stations. Hence, $N_A(I', i_1, i_2) - N_A(I', i_1)$ is simply the number of clients covered by $i_2$ when the algorithm is given $(I', i_1, i_2)$ as an input, and $N_A(I', i_2) - N_A(I')$ is the number of clients covered by $i_2$ when the algorithm is given $(I', i_2)$ as an input. By the definition of the algorithm, the set of clients that the algorithm will try to cover by $i_2$ when $i_1$ is available is a subset of the clients that the algorithm will try to cover by $i_2$ when $i_1$ is not available. Hence, the lemma follows from the greediness of the algorithm with respect to $i_2$. \hfill \Box

2.3.5 The budgeted maximum assignment problem

In this section we present an approximation algorithm for the following problem: find a subset $I'$ of base stations whose cost is at most the given budget and that maximizes $N(I')$. We refer to this problem as the budgeted maximum assignment problem (BMAP). Algorithm 2.2 and its analysis generalize the ideas of [44] in the sense that the problem dealt in [44] does not include capacities nor non-uniform demands. Notice that we use here Algorithm 2.1 to compute $N_A(\cdot)$, and hence all subsets of base stations are ordered.
If the optimal solution has less than three base stations, we will consider it in the first step of the algorithm, and get at least \( \frac{1}{2} \) of its value. It is left to take care of the case that the optimal solution has at least three base stations. In this case, we order the base stations in \( OPT \) by selecting at each step the set in \( OPT \) that maximizes the difference in the value of \( N_A(\cdot) \). Let \( Y \) be the first three base stations according to this order, and let \( Y' \) be the set of base stations that are added to \( Y \) when the algorithm considers \( Y \) as its initial set of opened base stations. For the rest of the discussion, reorder \( OPT \setminus Y \) according to the order the base stations were considered by Algorithm 2.2.

Let \( \ell \) be the number of base stations opened by the algorithm before the first base station from \( OPT \setminus Y \) is considered but not added to \( Y' \) because its addition would violate the budget constraint. Denote these base stations by \( i_1, \ldots, i_\ell \), and let \( i_{\ell+1} \) be the first base station from \( OPT \setminus Y \) is considered but not added to \( Y' \). In what follows, let \( G_h = \bigcup_{k=1}^{h} i_k \) (in this order).

Notice that by fixing \( Y \) to be the three first base stations in the input to Algorithm 2.1, we also fix the clients that are going to be assigned to them by the algorithm, regardless of what other base stations are opened. Hence, in what follows we abuse notation and for \( I' \) such that \( Y \cap I' = \emptyset \) we denote by \( N_A(I') \) the number of clients that are covered by \( I' \) when Algorithm 2.1 is given the input \( Y, I' \).

**Lemma 2.3.9.** For each \( i_k, k = 1, \ldots, \ell + 1 \) we have:

\[
\frac{c_{i_k}}{B} \cdot (N_A(OPT \setminus Y) - N_A(G_{k-1})) \leq N_A(G_k) - N_A(G_{k-1}).
\]

**Proof.** For each base station \( i \in OPT \setminus (Y \cup G_{k-1}) \), we have

\[
\frac{N_A(G_{k-1}, i) - N_A(G_{k-1})}{c_i} \leq \frac{N_A(G_{k-1}, i_k) - N_A(G_{k-1})}{c_{i_k}}.
\]

Hence,

\[
(N_A(G_{k-1}, i) - N_A(G_{k-1})) \leq c_i \cdot \frac{N_A(G_k) - N_A(G_{k-1})}{c_{i_k}}.
\]

Summing up over all \( i \in OPT \setminus (Y \cup G_{k-1}) \) we get:

\[
\sum_{i \in OPT \setminus (Y \cup G_{k-1})} (N_A(G_{k-1}, i) - N_A(G_{k-1})) \leq \sum_{i \in OPT \setminus (Y \cup G_{k-1})} c_i \cdot \frac{N_A(G_k) - N_A(G_{k-1})}{c_{i_k}}.
\]

Since \( \sum_{i \in OPT \setminus (Y \cup G_{k-1})} c_i \leq B \),

\[
\sum_{i \in OPT \setminus (Y \cup G_{k-1})} (N_A(G_{k-1}, i) - N_A(G_{k-1})) \leq B \cdot \frac{N_A(G_k) - N_A(G_{k-1})}{c_{i_k}}.
\]

By Lemma 2.3.8 we get:

\[
N_A(G_{k-1}, OPT \setminus (Y \cup G_{k-1})) \leq B \cdot \frac{N_A(G_k) - N_A(G_{k-1})}{c_{i_k}},
\]

and by Lemma 2.3.7 we get

\[
N_A(OPT \setminus Y) \leq B \cdot \frac{N_A(G_k) - N_A(G_{k-1})}{c_{i_k}},
\]

which completes the proof. \( \square \)
Lemma 2.3.10. For each \( i_k, k = 1, \ldots, \ell + 1 \) we have:

\[
N_A(G_k) \geq \left[ 1 - \prod_{h=1}^{k} \left( 1 - \frac{c_{ih}}{B} \right) \right] \cdot N_A(OPT \setminus Y)
\]

Proof. By induction on \( k \). For \( k = 1 \), \( \frac{N_A(i_1)}{c_{i_1}} \geq \frac{N_A(OPT \setminus Y)}{B} \) by Lemma 2.3.9. Suppose the lemma is true for \( k - 1 \), we show it holds for \( k \).

\[
N_A(G_k) = N_A(G_{k-1}) + [N_A(G_k) - N_A(G_{k-1})] \\
\geq N_A(G_{k-1}) + \frac{c_{ik}}{B} \cdot (N_A(OPT \setminus Y) - N_A(G_{k-1})) \\
= \left( 1 - \frac{c_{ik}}{B} \right) \cdot N_A(G_{k-1}) + \frac{c_{ik}}{B} \cdot N_A(OPT \setminus Y) \\
\geq \left( 1 - \frac{c_{ik}}{B} \right) \cdot \left[ 1 - \prod_{h=1}^{k-1} \left( 1 - \frac{c_{ih}}{B} \right) \right] \cdot N_A(OPT \setminus Y) + \frac{c_{ik}}{B} \cdot N_A(OPT \setminus Y) \\
= \left[ 1 - \prod_{h=1}^{k} \left( 1 - \frac{c_{ih}}{B} \right) \right] \cdot N_A(OPT \setminus Y).
\]

Here, the first inequality follows from Lemma 2.3.9, and the second inequality follows from the induction hypothesis.

\[\Box\]

Theorem 2.3.11. Algorithm 2.2 is a \( \frac{e-1}{2e} \)-approximation for BMAP.

Proof. Notice that for positive \( c_1, \ldots, c_r \) such that \( \sum_{i=1}^{r} c_i = A \), the function \( (1 - \prod_{i=1}^{r} (1 - \frac{c_i}{A})) \) is minimized when \( c_1 = \cdots = c_r = \frac{A}{r} \). From Lemma 2.3.10 and the fact that the total cost of \( G_{\ell+1}, c(G_{\ell+1}) \), is more than \( B \), we get:

\[
N_A(G_{\ell+1}) \geq \left[ 1 - \prod_{h=1}^{\ell+1} \left( 1 - \frac{c_{ih}}{B} \right) \right] \cdot N_A(OPT \setminus Y) \\
\geq \left[ 1 - \prod_{h=1}^{\ell+1} \left( 1 - \frac{c_{ih}}{c(G_{\ell+1})} \right) \right] \cdot N_A(OPT \setminus Y) \\
\geq \left[ 1 - \left( 1 - \frac{1}{\ell + 1} \right)^{\ell+1} \right] \cdot N_A(OPT \setminus Y) \\
\geq \left( 1 - \frac{1}{e} \right) \cdot N_A(OPT \setminus Y).
\]

From Lemma 2.3.8 we know that \( N_A(G_{\ell}) + N_A(i_{\ell+1}) \geq N_A(G_{\ell+1}) \), hence we get:

\[
N_A(G_{\ell}) + N_A(i_{\ell+1}) \geq \left( 1 - \frac{1}{e} \right) \cdot N_A(OPT \setminus Y).
\]

By the way we ordered \( OPT \) and by Lemma 2.3.8 we have that

\[
N_A(i_{\ell+1}) \leq \frac{1}{3} N_A(Y).
\]
Algorithm 2.3 k4k-BUDGETED CELL PLANNING

1: Let $I_1$ be the output of Algorithm 2.2.
2: Let $I_2$ be a set of base stations of maximum size, having a total opening cost less than or equal to $B$.
3: if $N_A(I_1) < |I_2|$ then
4: Output $I_2$ and a set of
5: $|I_2|$ clients that can be covered using the oracle.
6: else
7: Output $I_1$ and the clients covered by Algorithm 2.1 for these base stations.
8: end if

We combine the two last inequalities and get:

$$N_A(Y \cup G_\ell) = N_A(Y) + N_A(G_\ell) \geq N_A(Y) + \left(1 - \frac{1}{e}\right) \cdot N_A(OPT \setminus Y) - N_A(i_{\ell+1})$$
$$\geq N_A(Y) + \left(1 - \frac{1}{e}\right) \cdot N_A(OPT \setminus Y) - \frac{1}{3} N_A(Y)$$
$$\geq \left(1 - \frac{1}{3}\right) N_A(Y) + \left(1 - \frac{1}{e}\right) \cdot N_A(OPT \setminus Y)$$
$$\geq \left(1 - \frac{1}{e}\right) \cdot N_A(OPT)$$
$$\geq \frac{e-1}{2e} \cdot N(OPT),$$

where the last inequality follows from Theorem 2.3.6. This completes the proof. 

\[\square\]

2.3.6 Approximating k4k-BCPP

We are now ready to present a $\frac{e-1}{3e-1}$-approximation algorithm for k4k-BCPP.

**Theorem 2.3.12.** Algorithm 2.3 is a $\frac{e-1}{3e-1}$-approximation algorithm for the k4k-budgeted cell planning problem.

**Proof.** Let $\tilde{n}$ be the number of covered clients in the solution obtained by Algorithm 2.3, and let $n^*$ be the maximum number of satisfied clients as obtained by the optimal solution. In the latter, let $n_1^*$ denote the number of clients that are satisfied by a single base station, and $n_2^*$ denote the number of clients satisfied by more than one base station. Let $I^*$ denote the set of base stations opened (by the optimal solution) for satisfying these $n^* = n_1^* + n_2^*$ clients.

Let $N(OPT)$ denote the value of the optimal solution for the BMAP instance. It holds that $N(OPT) \geq n_1^*$. For the solution $I_1$ we know that

$$\tilde{n} \geq N_A(I_1) \geq \frac{e-1}{2e} N(OPT) \geq \frac{e-1}{2e} n_1^*.$$ (2.5)
We get:

\[
\frac{3e-1}{2e} \tilde{n} = \tilde{n} + \frac{e - 1}{2e} \tilde{n}
\]  

\[
\geq \tilde{n} + \frac{e - 1}{2e} |I^*|
\]  

\[
\geq \frac{e - 1}{2e} n_1^* + \frac{e - 1}{2e} n_2^*
\]

\[
= \frac{e - 1}{2e} n^*
\]

(2.6) (2.7) (2.8) (2.9)

where Inequality (2.7) follows from the fact that \(\tilde{n} \geq |I_2| \geq |I^*|\) and the \(k4\k\) property, and Inequality (2.8) is based on (2.5) and Lemma 2.3.3. \(\square\)

2.4 The minimum-cost cell planning problem

Recall that the minimum-cost cell planning problem asks for a subset of base stations \(I' \subseteq I\) of minimum cost that satisfies at least \(\gamma\) of the demands of all the clients, for a given constant \(0 < \gamma \leq 1\), while maintaining the capacity of every base station.

Let \(z_i\) denote the indicator variable of an opened base station, i.e., \(z_i = 1\) if base station \(i \in I\) is selected for opening, and \(z_i = 0\) otherwise. Consider the following integer program for this problem (IP1).

In the first set of constraints (2.10) we ensure that at least \(\gamma\) of the demand \(d_j\) of every client \(j\) is satisfied, while the second set (2.11) ensure that the ability of every open base station to satisfy the demands of the clients is limited by its capacity (and that clients can be satisfied only by opened base stations). The contribution \(Q(i, j)\) of base station \(i\) to client \(j\), taking into account interference from other base stations, can be modeled as in (2.3) or (2.4), or any other predefined behavior of interference. However, because of the way \(Q(i, j)\)'s are computed, the integer program (IP1) is not linear when interference exists. Without loss of generality we may assume that every client in the input has demand at least 1, as the used units can be scaled accordingly and there is no need for “covering” the clients with zero demand. Lastly, we use the following assumption as well.

Assumption 1. The values \(\{w_i\}_{i \in I}\) and \(\{\gamma d_j\}_{j \in J}\) are integers.

\[
\min \sum_{i \in I} c_i z_i \quad \text{(IP1)}
\]

s.t. \[ \sum_{i \in I} Q(i, j) \geq \gamma \cdot d_j, \quad \forall j \in J \quad \text{(2.10)} \]

\[
\sum_{j \in J} x_{ij} \leq z_i, \quad \forall i \in I \quad \text{(2.11)}
\]

\[
0 \leq x_{ij} \leq 1, \quad \forall i \in I, j \in S_i \quad \text{(2.12)}
\]

\[
x_{ij} = 0, \quad \forall i \in I, j \notin S_i
\]

\[
z_i \in \{0, 1\}, \quad \forall i \in I \quad \text{(2.13)}
\]
When there are no interference, IP$_1$ becomes much simpler. (LP$_2$) is its linear programming relaxation, in which the last set of integrality constraints (2.13) is relaxed to allow the variables $z_i$ to take rational values between 0 and 1.

$$
\begin{align*}
\text{min} & \sum_{i \in I} c_i z_i \\
\text{s.t.} & \sum_{i \in I} w_i x_{ij} \geq \gamma \cdot d_j, \quad \forall j \in J & (2.14) \\
& \sum_{j \in J} x_{ij} \leq z_i, \quad \forall i \in I & (2.15) \\
& 0 \leq x_{ij} \leq 1, \quad \forall i \in I, j \in S_i & (2.16) \\
& x_{ij} = 0, \quad \forall i \in I, j \notin S_i & (2.17)
\end{align*}
$$

In fact, LP$_2$ is a minimum-cost flow problem. To see that, consider the network $(G, u, c')$, which is defined as follows.

- The graph $G = (V, E)$, where $V = I \cup J \cup \{s\}$ and $E = \{(i, j) \mid i \in I, j \in S_i\} \cup \{(s, i) \mid i \in I\} \cup \{(j, s) \mid j \in J\}$.

- The vertex capacity function $u$, where $u(s) = \infty$, $u(i) = w_i$ for $i \in I$ and $u(j) = \gamma d_j$ for $j \in J$.

- The vertex cost function $c'$, where $c'(i) = \frac{c_i}{w_i}$ for $i \in I$, $c'(j) = 0$ for $j \in J$ and $c'(s) = -1 - \max_{i \in I} c'(i)$.

Accordingly, Assumption 1 yields that there is an optimal solution to the above flow problem, in which the flow in every edge is integral (specifically, any open base station $i$ that serves a client $j$ contribute at least one unit of the client’s demand). Moreover, this solution can be computed efficiently using the known algorithms for minimum-cost flow [6]. We denote the solution to LP$_2$ which correspond to that flow by $\{\bar{z}, \bar{x}\}$. Let $I_j = \{i \in I : \bar{x}_{ij} > 0\}$, for every client $j \in J$. Note that by this definition it follows that for every $i \in I_j$ we have that

$$w_i \bar{x}_{ij} \geq 1. \quad (2.18)$$

Next we introduce our approximation algorithms for CPP with no interference. Our algorithm is based on the greedy approach and achieves an approximation of $O(\log W)$, where $W = \max_{i \in I} \{w_i\}$. Unlike the criterion used by other known greedy heuristics, our greedy algorithm chooses to open a base station which maximizes the increase in the maximum demand that can be satisfied by the entire set of the opened base stations.

**A greedy $O(\log W)$-approximation algorithm**

In this section we present a greedy algorithm for CPP with no interference. It generalizes the algorithm of Chuzhoy and Naor [21] for set cover with hard capacities. For the sake of
Algorithm 2.4 MINIMUM-COST CELL PLANNING (GREEDY)

1: $I' \leftarrow \emptyset$
2: while $f(I') < \gamma \sum_{j \in J} d_j$ do
3: \hspace{1em} Let $i = \arg \min_{i \in I : f_{I'}(i) > 0} \frac{c_i}{f_{I'}(i)}$
4: \hspace{1em} $I' \leftarrow I' \cup \{i\}$.
5: end while
6: return $I'$.

For a subset of base stations, $H \subseteq I$, let $f(H)$ denote the maximum total demand (in demand units, where the clients need not to be fully covered) that can be satisfied by the base stations in $H$. For $i \in I$, define $f_H(i) = f(H \cup \{i\}) - f(H)$. Note that when there are no interference, we can calculate $f(H)$ using the following linear program:

$$
\begin{align*}
\text{max} & \sum_{j \in J} \sum_{i \in H} w_i x_{ij} & \quad \text{(LP)} \\
\text{s.t.} & \sum_{i \in H} w_i x_{ij} \leq \gamma \cdot d_j, & \forall j \in J \\
& \sum_{j \in J} x_{ij} \leq 1, & \forall i \in H \\
& 0 \leq x_{ij} \leq 1, & \forall i \in H, j \in S_i \\
& x_{ij} = 0, & \forall i \in H, j \notin S_i
\end{align*}
$$

We need the following lemma.

Lemma 2.4.1. Let $H \subseteq I$ be a subset of the base stations, and let $H_1$ and $H_2$ be a partition of $H$ into two disjoint sets. Then, there exists a solution to $\mathbf{IP}_1$ in which the base stations in $H_1$ satisfy a total of $f(H_1)$ demand units.

Proof. Assume that we are given a solution to $\mathbf{IP}_1$, $\{\tilde{z}, \{\tilde{x}\}\}$, such that the base stations in $H_1$ satisfy a total of less than $f(H_1)$ demand units. Let $x$ be an optimal solution to $\mathbf{LP}_2$ for $H_1$. Iteratively, update $\{\tilde{z}, \{\tilde{x}\}\}$ as follows: while $\sum_{i \in H_1} \sum_{j \in J} w_i \tilde{x}_{ij} < f(H_1),$

1. Let $i \in H_1$ be a base station such that $\sum_{j \in J} \tilde{x}_{ij} < \sum_{j \in J} x_{ij}$ (notice there must exist such $i$).
2. Let $j \in J$ be a client such that $\tilde{x}_{ij} < x_{ij}$.
3. Let $\Delta = w_i \cdot \min \{x_{ij} - \tilde{x}_{ij}, \sum_{j \in J} x_{ij} - \sum_{j \in J} \tilde{x}_{ij}\}$.
4. If there exists a base station $i' \in H_2$ such that $\tilde{x}_{i'j} > 0$, let $\delta = \min \{w_{i'j} \tilde{x}_{i'j}, \Delta\}$ and set $\tilde{x}_{ij} \leftarrow \tilde{x}_{ij} + \frac{\delta}{w_i}$ and $\tilde{x}_{i'j} \leftarrow \tilde{x}_{i'j} - \frac{\delta}{w_{i'j}}.$
5. Else, there exists a base station \( i' \in H_1 \) such that \( \tilde{x}_{i'j} > x_{ij} \). Let \( \delta = \min \{ w_{i'} \cdot (\tilde{x}_{i'j} - x_{i'j}), \Delta \} \) and set \( \tilde{x}_{ij} \leftarrow \tilde{x}_{ij} + \frac{\delta}{w_i} \) and \( \tilde{x}_{i'j} \leftarrow \tilde{x}_{i'j} - \frac{\delta}{w_{i'}} \).

One can easily verify that the above process halts with a feasible solution with the desired property. \( \square \)

Let \( i_1, i_2, \ldots, i_k \) be the base stations that were chosen by Algorithm 4 to the solution, in the order they were chosen. Let \( I'_\ell \) be the solution at the end of iteration \( \ell \) of the algorithm. Let OPT be a set of base stations that comprises an optimal solution, \( \{ \tilde{z}, \tilde{x} \} \).

Next, we inductively define for each iteration \( \ell \) and \( i \in \text{OPT} \setminus I'_\ell \) a value \( a_\ell(i) \), so that the following invariant holds: it is possible to cover all the clients using the base stations in OPT also pays for it. Otherwise, we charge each base station in OPT \( \setminus I'_\ell \) with the capacities \( a_\ell(i) \) for \( i \in \text{OPT} \setminus I'_\ell \) and \( w_i \) for \( i \in I'_\ell \).

Let \( a_0(i) = \sum_{j \in J} w_i x_{ij} \). The invariant holds trivially. Consider the \( \ell \)th iteration. By the induction hypothesis and Lemma 2.4.1, there exists a solution \( \{ \tilde{z}, \tilde{x} \} \) of IP such that the base stations in \( I'_\ell \) satisfy a total of exactly \( f(I'_\ell) \) demand units and each base station \( i \in \text{OPT} \setminus I'_\ell \) satisfies at most \( a_{\ell-1}(i) \) demand units. For each \( i \in \text{OPT} \setminus I'_\ell \) let \( a_\ell(i) = \sum_{j \in J} w_i x_{ij} \).

In what follows, we charge the cost of the base stations that are chosen by Algorithm 2.4 to the base stations in OPT. If \( i \in \text{OPT} \), we do not charge any base station for its cost, since OPT also pays for it. Otherwise, we charge each \( i \in \text{OPT} \setminus I'_\ell \) with \( \frac{c_i}{f_{\ell-1}'(i_\ell)} \cdot (a_{\ell-1}(i) - a_\ell(i)) \). Notice that the total cost of \( i \) is indeed charged.

Consider a base station \( i \in \text{OPT} \). If \( i \in I'_\ell \), let \( h \) denote the iteration in which it was added to the solution. Else, let \( h = k + 1 \). For \( \ell < h \), it follows from the definition of \( a_{\ell-1}(i) \) that \( f_{\ell-1}'(i) \geq a_{\ell-1}(i) \). By the greediness of Algorithm 4 it holds that:

\[
\frac{c_i}{f_{\ell-1}'(i_\ell)} \leq \frac{c_i}{f_{\ell-1}'(i)} \leq \frac{c_i}{a_{\ell-1}(i)},
\]

and the total cost charged upon \( i \) is:

\[
\sum_{\ell=1}^{h-1} \frac{c_i}{f_{\ell-1}'(i_\ell)} \cdot (a_{\ell-1}(i) - a_\ell(i)) \leq c_i \sum_{\ell=1}^{h-1} \frac{(a_{\ell-1}(i) - a_\ell(i))}{a_{\ell-1}(i)}
\]

\[
= c_i \cdot H(a_0(i))
\]

\[
= c_i \cdot O(\log a_0(i))
\]

\[
= c_i \cdot O(\log w_i),
\]

where \( H(r) \) is the \( r \)th harmonic number. This completes the analysis.

### 2.5 Simulation results and practical aspects

In the previous sections, we proved theoretical bounds on the performance of our algorithms. In this section we discuss the practicality of the models and evaluate our algorithms through simulations. The simulation results are comprised of two separate sets of simulations: planning “greenfield” networks and planning UMTS networks in a real urban environment.
2.5.1 On greedy algorithms for minimum-cost cell planning

As mentioned earlier, various algorithmic approaches have been studied for the minimum-cost cell planning problem. Methods include genetic algorithms, tabu search, branch-and-bound, and simulated annealing. These methods do not have a guaranteed polynomial running time and the quality of their solutions depend on the duration of the execution. For our comparisons, we would like to consider algorithms with polynomial running time. The greedy technique (e.g., [56, 70, 78]) is a good practical (and very popular in real-life planning [57, Chapter 14]) candidate for such a comparison (the works of [56] and [78] uses methods that are very similar to [70]).

The SCBPA (Set Cover Base stations Positioning Algorithm) model due to Tutschku [70] is perhaps the most natural approach. Its aim is to provide sufficient coverage for a planning area using as few base stations as possible. Usually, a set \( \mathcal{S} \) of potential base stations is prescribed, each of them serves a certain sub-area \( S_i \subseteq J, i \in \mathcal{S} \), of the planning area \( J \), and each has an installation cost \( c_i \). The goal is to cover the entire area \( J \) with a minimum-cost subset of base stations. This is essentially the minimum-weight set cover problem.

The model for the minimum-weight set cover problem does not support capacities for base stations neither it supports client demands. Hence, this model cannot be used for solving cell planning problems for future networks. Therefore, we define a natural generalization of the Tutschku’s SCBPA algorithm specifically designated for capacitated versions of cell planning problems; we call this generalization Extended-SCBPA algorithm (ESCBPA). The ESCBPA greedy algorithm picks at each step a base station that maximizes the ratio between the amount of non-covered demands this station can satisfy and its cost. The algorithm stops when all the clients are fully satisfied (or alternatively, if not enough capacity is left). Notice that when there is no restriction on the capacity of the base stations, this algorithm becomes identical to the one described in [70].

From a computationally point of view the SCBPA is an \( O(\log n) \)-approximation algorithm for an \( n \)-clients CPP [22, 72], while its generalization, ESCBPA, can be arbitrarily bad. To see this, consider an instance with three base stations \( I = \{1, 2, 3\} \) of unit capacity and opening costs of \( c_1 = 1, c_2 = C, \) and \( c_3 = \epsilon (C > 1 \text{ and } \epsilon > 0) \), and a set of two clients \( J = \{1, 2\} \), each has a unit demand. In addition, assume that \( S_1 = \{1\}, S_2 = \{2\}, \) and \( S_3 = \{1, 2\} \). First, the ESCBPA algorithm will select to open the third base station and client 1, for example, is the one who is satisfied. Then, the second base station will be picked and client 2 will be satisfied. The total cost of this solution is \( C + \epsilon \). However, by opening the first base station (and satisfying client 1), and the third (now satisfying client 2), the optimal solution achieves a full coverage with a total cost of \( 1 + \epsilon \). Hence the performance guarantee of the ESCBPA algorithm is unbounded. Moreover, one can show that ESCBPA might not produce a feasible solution to CPP (that is, to satisfy all the demand of the clients), although such a solution exists.

Our greedy algorithm (described in Section 2.4) has a different greedy criterion: it chooses to open a base station which maximizes the increase in the maximum demand that can be satisfied by the entire set of the so-far-opened base stations. To emphasize the different behavior of these two greedy approaches, consider the following example. We have two base stations...
Algorithm 2.5 \textsc{Minimum-cost cell planning (lp rounding)}

1: Calculate \{\bar{z}, \bar{x}\} as explained above.
2: \( W \leftarrow \max_{i : \bar{z}_i > 0} \{w_i\} \); \( \lambda \leftarrow \Theta(W \sqrt{\log n}) \).
3: for all \( i \in I \) do
4: \( z_i \leftarrow 1 \) with probability \( \min\{1, \lambda \cdot \bar{z}_i\} \).
5: for all \( j \in J \) do
6: \( x_{ij} \leftarrow \frac{\bar{x}_{ij}}{\bar{z}_i} z_i \).
7: end for
8: end for
9: return \{z\} and \{x\}.

stations \( I = \{1, 2\} \) with capacities \( w_1 = 1 + 3\epsilon \) and \( w_2 = 1 + 2\epsilon \), for a given \( \epsilon > 0 \). The opening costs of these two base stations are taken to be the same. There are five clients \( J = \{1, \ldots, 5\} \) with \( d_1 = d_2 = 1 + \epsilon, d_3 = 1, \) and \( d_4 = d_5 = \epsilon \). In addition, \( S_1 = \{1, 2, 3\} \) while \( S_2 = \{3, 4, 5\} \). The ESCBPA will pick the base station with the largest amount of net capacity (base station 1), yielding a satisfaction of at most one single client; our greedy algorithm chooses the base station that can satisfy the largest amount of non-covered demand (base station 2), resulting in a full coverage of all his 3 clients. However, when there are no upper limitations on the capacity of the base stations, both the ESCBPA and our greedy algorithm are the same.

2.5.2 An LP-rounding based \( O(W \sqrt{\log n}) \)-approximation algorithm

Many NP-hard combinatorial optimization problems can be formulated as an integer programming problem, which can be subsequently relaxed into a linear programming problem (whose solution is a lower bound of the integer program). However, the optimal solution to the linear programming problem in general does not coincide with the solution to the initial integer programming problem. A basic approach is to solve the linear program and then convert the fractional solution obtained into an integer solution, trying to ensure that in this process the cost does not increase too much.

In this section we present an LP-rounding based approximation algorithm for the non-interference variant of CPP. Our algorithm is based on solving the LP-relaxation (\( \text{LP}_3 \)), and randomly rounding the fractional solution to an integer solution. The integer solution it produces is within a factor of \( O(W \sqrt{\log n}) \) of the optimum, where \( W = \max_{i \in I} \{w_i\} \), as before.

Notice that from a worst-case viewpoint, our \( O(\log W) \)-approximation algorithm (Section 2.4) is much better than the algorithm presented in this section. The purpose of discussing Algorithm 2.5 is to examine how this algorithm is performed in practice.

\textbf{Lemma 2.5.1.} The expected value of the cost of the solution produced by Algorithm 5 is at most \( \lambda \cdot \text{OPT} \), where \( \text{OPT} \) is the value of the optimal solution to \( \text{LP}_3 \).

\textbf{Proof.}

\[
E(\sum_{i \in I} c_i z_i) = \sum_{i \in I} c_i \cdot \Pr(z_i = 1) \leq \sum_{i \in I'} c_i \cdot \lambda \bar{z}_i \leq \lambda \cdot \text{OPT}.
\]
Lemma 2.5.2. For every client \( j \in J \), the probability that \( j \) is not covered is at most \( \frac{1}{2n} \).

Proof. Consider a client \( j \in J \). We divide the set of base stations participating in its coverage, \( I_j \), into two (disjoint) sets: those with large values of \( z_i \), and those with smaller \( z_i \)'s; Namely, \( I_j = I_j^L \cup I_j^S \), where \( I_j^L = \{ i : \lambda z_i \geq 1, x_{ij} > 0 \} \) and \( I_j^S = I_j \setminus I_j^L \).

Followed by this partition, the total contribution of \( I_j^L \) and \( I_j^S \) to the coverage of client \( j \) can be described by

\[
\chi_j^L = \sum_{i \in I_j^L} w_i x_{ij} \quad \text{and} \quad \chi_j^S = \sum_{i \in I_j^S} w_i x_{ij},
\]

respectively. Accordingly, define

\[
\tilde{\chi}_j^L = \sum_{i \in I_j^L} w_i \bar{x}_{ij} \quad \text{and} \quad \tilde{\chi}_j^S = \sum_{i \in I_j^S} w_i \bar{x}_{ij}.
\]

Hence,

\[
\Pr(\text{client } j \text{ is not covered}) = \Pr(\chi_j^L + \chi_j^S < \gamma d_j) = \Pr(\chi_j^S < \gamma d_j - \chi_j^L) \leq \Pr(\chi_j^S < \bar{\chi}_j^S)
\]

where the last inequality follows as \( \bar{\chi}_j^L \leq \chi_j^L \) (by the rounding procedure) and \( \bar{\chi}_j^L + \bar{\chi}_j^S \geq \gamma d_j \) (by the feasibility of \( \{ \bar{z}, \bar{x} \} \)). For \( i \in I_j^S \), the random variables \( \theta_{ij} = w_i x_{ij} \) are independent over the \( i \)'s, taking their values from \([0, w_i]\), and have mean

\[
E(\theta_{ij}) = \lambda w_i \bar{x}_{ij} \quad \text{and} \quad \sum_{i \in I_j^S} E(\theta_{ij}) = \lambda \bar{\chi}_j^S.
\]

Using the above discussion, Hoeffding’s bound\(^4\) [39], Inequality (2.18), and by recalling the value of \( \lambda \), we get

\[
\Pr(\chi_j^S < \bar{\chi}_j^S) = \Pr\left( \sum_{i \in I_j^S} \theta_{ij} < \bar{\chi}_j^S \right) = \Pr\left( E\left[ \sum_{i \in I_j^S} \theta_{ij} \right] - \sum_{i \in I_j^S} \theta_{ij} > (\lambda - 1) \bar{\chi}_j^S \right) \leq \exp\left( -\frac{2(\lambda - 1)^2 (\bar{\chi}_j^S)^2}{\sum_{i \in I_j^S} w_i^2} \right) \leq \exp\left( -\frac{2(\lambda - 1)^2 |I_j^S|^2}{|I_j^S| W^2} \right) \leq \exp\left( -2 ((\lambda - 1)/W)^2 \right) \leq \frac{1}{2n},
\]

\(^4\)Hoeffding bound [39]. Let \( X_1, \ldots, X_n \) be independent random variables with finite first and second moments, whose values in \([a_i, b_i]\), respectively. Then for any \( \epsilon > 0 \),

\[
\Pr\left\{ \sum_{i=1}^n X_i - E\left( \sum_{i=1}^n X_i \right) \geq \epsilon \right\} \leq \exp\left( -\frac{\epsilon^2 n}{2 \sum_{i=1}^n (b_i - a_i)^2} \right).
\]
as required.

Thus, by the naive union bound, the probability that there is an uncovered client \( j \) is at most \( \frac{1}{2} \). Thus, the output of Algorithm 2.5 covers \( J \) with high probability, and we have the following theorem

**Theorem 2.5.3.** Algorithm 2.5 finds a subset \( I' \subset I \) of base stations that satisfies (2.10)-(2.13) with high probability, and whose expected cost is no more than \( O(W \sqrt{\log n}) \) times the optimal cost, where \( W = \max_{i \in I} \{w_i\} \).

**On the integrality gap of LP3**

Given is an LP-relaxation for a minimization problem, the *integrality gap* of the relaxation is the supremum of the ratio between the optimal integral and fractional solutions. It is not hard to see that if the cost of the solution found by an algorithm is compared directly with the cost of an optimal fractional solution, the best approximation factor one can hope to prove is the integrality gap of the relaxation.

Unfortunately, it is not possible to design an approximation algorithm for CPP (even without interference) which is based solely on LP3 and whose approximation factor is independent of \( W \), since the integrality gap of LP3 is \( \Omega(W) \). To see this, consider an instance with a single client and one base station with some fixed cost \( c > 0 \). The capacity of the base station is \( W \) while the demand of the client is 1. Obviously, a fractional solution can have \( z = \frac{1}{W} \) and cost of \( \frac{c}{W} \), while the optimal integer solution must open the base station and has cost of at least \( c \). Thus, we have an integrality gap of at least \( W \).

In the following two sets of simulations, we compare ESCBPA algorithm and our techniques (Sections 2.4–2.5.2) to the value of the optimal solution of the linear program LP2 (which is a lower bound on the optimal solution for the minimum-cost cell planning problem).

### 2.5.3 Planning “greenfield” networks

In this set of simulations we built a network consisting of an \( n \times n \)-grid of clients’ locations (demand points that we considered as *bins*). Each bin is assumed to have traffic demand according to the Erlang distribution with parameter 1/30 (which is realistic in urban areas in busy hours). We consider a random set of 10% of all grid points to be possible locations for positioning base stations. For each such possible location, we consider the set of 8 optional sectorized antennas (with horizontal azimuth values of 0°, 45°, . . . , 315° and with an open angle of 30°, 60° and 120°). The radius of each antenna was chosen up to 5 bins length, but under considerations of urban areas (i.e., relatively small cells). The height and the tilt of each antenna are taken to be fixed. So, for the \( n \times n \)-grid of clients we have a set of \( 24 \times 0.1n^2 \) possible configurations of antennas. Each antenna has a coverage area reflected from its geometric parameters. Both the capacity and the installation cost of each antenna are assumed to be a linear function of the sum of demands of all of the clients within its coverage area. Notice that we assume that no interference is exist in this scenarios. The cell planning task here is to choose a minimum-cost subset of antenna-configurations in order to satisfy the traffic demands of all the clients.
Figure 2.1: Planning future networks

Figure 2.1(a) depicts the ratio between the solution cost and the LP cost as a function of network size of each of the three methods. We ran our simulation on network size of $25 - 144$ bins (clients). Our results show that our approximation algorithms achieve a significant lower value of the solution cost than the greedy heuristic. Among our two algorithms, the $O(\log W)$-approximation algorithm was better (up to a factor of 50%) than the $O(W\sqrt{\log n})$-approximation algorithm, where $W$ is equal here to the maximum number of bins associated with a base station (e.g., $\max_{i \in I} |S_i|$). Moreover, we are far away from the corresponding theoretical worst-case behavior of $O(\log W)$, and in fact, all simulation results are within a factor of 2.5 of the optimum lower bound. Since the $O(\log W)$-approximation algorithm is expected to produce a close-to-optimal solutions when the capacities are relatively low, our results indicate that such a behavior is also apparent when the capacities of the base stations is proportional to the sum of the traffic demand of their clients (under the anticipated RF conditions derived from the way the $S_i$’s are created in the pre-processing stage).

Figure 2.1(b) depicts the number of sectors selected for deployment as a function of the network size for the three algorithmic methods. We see that our $O(\log W)$-approximation algorithm selects approximately 15%-20% of the number of sectors selected by the ESCBPA algorithm. Notice that when the objective was the minimize the total number of sectors, both approximation algorithms gave relatively similar results.

2.5.4 Planning UMTS network in Helsinki

In the second simulation set we study the theoretical bounds of our approximation algorithms on real networks. We compare our algorithm to the optimal solution of the corresponding LP on microcellular and picocellular UMTS networks in a real urban environment.

We analyze an area of $800m^2$ in the city center of Helsinki, Finland (a square area with $(386200, 6674800)$ as the lower left corner and $(387000, 6675600)$ as the upper right corner (Fig-
Table 2.2: Cell planning in Helsinki

<table>
<thead>
<tr>
<th></th>
<th>Extended Tutschku</th>
<th>Greedy algorithm</th>
<th>Randomized rounding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution cost / LP₃ cost</td>
<td>2.987</td>
<td>1.683</td>
<td>1.886</td>
</tr>
<tr>
<td>Number of sectors</td>
<td>32</td>
<td>15</td>
<td>21</td>
</tr>
</tbody>
</table>

Table 2.2 shows that our $O(\log W)$-approximation algorithm was better than the ESCBPA algorithm and our $O(W^{1/2} \log n)$-approximation algorithm (in both objectives). In this real-data scenario our $O(\log W)$-approximation algorithm reached the same level of coverage with approximately half the number of sectors ESCBPA algorithm did. Notice that all three algorithms output relatively close-to-optimal solutions (1.68 to 2.98 times the lower bound on the optimal solution).

2.6 Conclusions and open problems

In this work, we describe new techniques for solving two important cell planning problems: the budgeted cell planning (BCPP) and the minimum-cost cell planning problems (CPP). As far as we know, previous work do not provide performance guarantee for these problems. Our formulation is very flexible and it allows the incorporation of several 4G cellular networks characterizations (such as smart antennas, capacities for base stations, interference,

---

5 Although the data in [46] does not contain capacities for base stations, base stations in this scenario were limited in their capacities in a similar way as in Simulation A.
and different-size-cells). Due to the NP-hardness of the problems, we resorted to polynomial-time approximation algorithms.

We show that although BCPP is NP-hard to approximate, we can still cover all useful scenarios by adopting a very practical assumption, called the $k+4k$-property, satisfied by every real cellular network, and we give a fully combinatorial $\frac{e}{e-1}$-approximation algorithm for this problem.

We obtained a combinatorial $O(\log W)$-approximation algorithm addressed a non-interference version of CPP, where $W$ is the largest capacity over all base stations selected for opening.

We simulated our algorithm to the CPP and compared its performance to the results produced by previous works (and some new suggested LP-rounded based technique). Two different simulation sets were conducted with scenarios relevant to future technologies. In both sets of simulation our results indicate that practical algorithms derived from the theoretical scheme can provide solutions that are close to the optimal solution and much better than our proved theoretical bound.

The main open problem is a development of an approximation algorithm for CPP (while taking into account also the interference).

Another open problem is the minimum-cost site planning problem. In this problem, in addition to what was given in CPP, we also have a set of potential geographic locations (sites) for installing a cluster of antennas (or base stations). Usually, such a cluster comprises of antennas of different pattern, power, and direction, and these are addressed to service a large area with relatively several different behavior subareas (e.g., a close high-density mall in one side and a farther low-density neighborhood, at the other side of the site). The problem is motivated by the operator’s need to reduce the number of geographical sites in which its equipment is installed since their rental is relatively high. Followed by the above discussion, we can define the minimum-cost site planning problem as the problem of finding a subset of sites and a subset of base stations in such a way that the demands of all the clients is satisfied, and the total cost of installing sites and base stations (on open sites) is minimized.

The minimum-cost site planning problem can be seen as two-level CPP for which the “sites” can be seen as the first level and the base stations, selected to be installed on these sites, as the second level. Since this problem is a natural generalization of CPP, we believe that solving CPP will make the solution for the former much more accessible.
Chapter 3

Algorithmic Aspects of Radio Access Networks Design

3.1 Introduction

Third generation network elements are functionally grouped into the Radio Access Network (RAN, or UTRAN, in UMTS systems) that handles all radio-related functionality; Core Network (CN), which is responsible for switching and routing calls and data connections to external networks, and the User Equipment (UE) that uses air interface to communicate with the base stations (see [51, Chapter 5]).

RAN architecture in current 3G systems consists of one or more Radio Network Sub-systems (RNSs) as depicted in Figure 3.1. Each RNS is a sub-network within the RAN, comprising of one Radio Network Controller (RNC) and one or more base stations (BSs). The RNC owns and controls radio resources in its RNS; it is the service access point for all services provided by the RAN to the CN over the Iu interface. It then communicates with the BSs in its RNS (over the lub interface) which are in charge of the communication to the UE over the WCDMA radio interface. RNCs may be connected to each other via an Iur interface (i.e., the open interface that allows soft handover between RNCs from different manufactures).

![RAN Architecture Diagram](image-url)

Figure 3.1: RAN architecture in current 3G systems
RAN in future systems is expected to be considerably different from current RANs. First, as mentioned before, in 4G systems the cell size is expected to be smaller than in 3G systems [45]. Therefore, the 4G RAN will contain many more base stations. A careful design must be used in order to handle this large number of base stations without a significant increase in the number of RNCs (i.e., RNC dimensioning). Second, the larger number of base stations (resulting in more frequent handover in the system) and the expected higher bit rate will result in a heavier load on the links between the RNCs and base stations. Finally, these changes should be made in a cost-efficient way.

In the current traditional star topology radio access networks (e.g., in UMTS systems), all base stations are directly connected to RNCs. When a tree topology is deployed (rather than a star), a base station is allowed to be connected to another base station rather than its RNC. However, base stations have no routing capabilities and they simply forward all received data towards their corresponding RNC and from the RNC to the corresponding base station. For example, in Figure 3.2, if a mobile user inside the coverage area of base station $B_1$ wishes to communicate with a mobile user inside the coverage area of base station $B_2$, their data traffic will be sent through RNC $R$ (RNCs nodes are colored in black while base-stations are whites). Therefore, the link connecting the parent of $B_1$ and $B_2$ to its parent, on the way to the RNC, must be capable of carrying large enough amount of traffic to handle both its own traffic as well as the traffic originating from its three children. Thus, the “heaviest” links are designed to be those that are connected directly to the RNC. These links must be able to handle all the traffic in their subtree. In the case of tree topologies, the constraints stem from technical limitations of the equipment. A base station cannot be connected to too many other base stations without creating a significant traffic reduction. Therefore, we assume that every base station and every RNC can only have a limited number of allowed connections. The planning problem is then to design the best possible tree taking these limitations as well as the cost of establishing the links into account. As we show in this chapter, this is a computationally difficult task.

In this chapter we rigorously study the problem of designing tree-topology based access networks for future cellular systems, and describe the theoretical as well as the practical aspects of the solutions to this problem.
3.1.1 Definitions and background

Consider a set \( I = \{1, 2, \ldots, n\} \) of base stations and a set \( J = \{1, 2, \ldots, m\} \) of RNCs. A symmetric connection cost \( w(i_1, i_2) \) is associated with every pair of base stations \( i_1 \) and \( i_2 \); another given cost is \( w(i, j) \) representing the cost of connecting base station \( i \) to RNC \( j \), for every \( i \in I, j \in J \). A tree-topology based access network is designed as a forest, such that each of its trees is rooted at an RNC node and contains the base stations that are under the responsibility of this RNC. In addition, every node \( u \in I \cup J \) is allowed to connect to no more than \( b(u) \) neighbors in the forest, for some \( b(u) \geq 1 \).

Given a spanning tree \( T \) of a subset \( I_T \subseteq I \) of base stations, rooted at RNC \( r \), we define the routing cost, \( d_T(r, i) \), between the RNC \( r \) and a base station \( i \in I_T \), as the sum of the costs along the unique path between them in the tree \( T \). The routing cost of the tree itself is defined by \( \sum_{i \in I_T} d_T(r, i) \). Our goal is to design a tree-topology based access network with a minimum total cost. Note that the closer a connection is to the root of the tree, the higher is its contribution to the routing cost of the tree.

An important observation is that the problem of designing access networks that comprise of multiple trees (i.e., multiple RNS radio access network) is reducible to the problem of designing a network comprising of a single tree (i.e., a single RNS). This means that the problem of dividing base stations among the possible RNCs is directly-solvable via our model. We represent the input to the problem as a complete graph \( G = (V, E) \), such that \( V = I \cup J \cup \{\hat{r}\} \), where \( \hat{r} \) is a “special” vertex to be defined later. There is an edge between every pair of vertices in \( I \) weighted by their corresponding connection cost. In addition, each RNC vertex in \( J \) is connected to all the vertices in \( I \) by an edge of weight equal to the connection cost between the corresponding base station and the RNC. Finally, the vertex \( \hat{r} \) is connected to all the vertices of \( J \) by an edge of zero weight. All other edges of \( G \) are assumed to have an infinite weight. The degree constraints of vertices of \( I \cup J \) are equal to the corresponding degree constraints given for base stations and RNCs. The degree constraint of \( \hat{r} \), \( b(\hat{r}) \), is defined to be \( |J| = m \). Hence, our access network design problem consists not only of the association of base stations to RNCs, but also of selecting a subset of RNCs to be deployed in the network. A consequence of the above reduction is that the problem of finding a forest with multiple RNCs is equivalent to the problem of finding a tree for a single RNC. We will therefore limit our attention to the case where \( |J| = 1 \).

Notice that when setting, in the above reduction, the value of \( b(\hat{r}) \) to be \( k \), \( k \leq m \), the model is extended to select only \( k \) out of the \( m \) RNCs to be installed on the network.

We define the bounded-degree minimum routing cost spanning tree problem (BDRT) as the problem of finding a minimum routing cost spanning tree, rooted at a given root \( r \), that meets the degree constraints \( b(v) \), for all \( v \in V \).

3.1.2 Our contribution

This chapter investigates algorithmic aspects of access network design in future cellular systems. We study the BDRT problem as a general model that captures several aspects of radio access networks design.
We first show that it is NP-hard to approximate the general problem (Section 3.3), and then present an approximation algorithm for the case where edge weights satisfy triangle inequality. To the best of our knowledge this is the first approximation algorithm for this problem. We show, in Section 3.3.2, that the case of $b(v) \geq 3$, for every $v \in V$, is approximable within a factor of $O(\log n)$. Four more heuristic algorithms for BDRT are presented in Section 3.3.3 and compared to the approximation algorithm. A generalization of BDRT is described in Section 3.4. In this generalization the traffic requirement of the base stations is also taken into account in the cost model. In this way, the model can deal with large variance in the traffic loads among the base stations, as expected in future networks.

In Section 3.5 we describe our simulations. We study the performance and the quality of the solutions of each of the five studied algorithms on instances of both BDRT and its generalization. Finally, we conclude that the combination of the approximation algorithm and one of the heuristics achieves a proven performance guarantee of $O(\log n)$ in the worst-case, together with a close to optimum solutions in practice, both for BDRT and its generalization.

### 3.2 Related work

Several non-star topologies for radio access networks have been proposed in the last few years [36, 37, 43, 45, 58, 67, 77].

Ring topologies have been proposed in [45, 58, 77]. The advantage of such a topology is, of course, its reliability; on the other hand, the delay on the path from a base station to the RNC may be significant. The authors of [45] present an $O(n^3)$-time algorithm for solving the corresponding design problem. However, such an approach is unlikely to be optimal since this problem, as modeled in [45], is the well-known traveling salesman problem (TSP) which is NP-hard to approximate in general. The algorithm presented in [45] is a “nearest neighbor” algorithm that guarantees a solution that is within a factor of $O(\log n)$ of the optimal solution of the problem [72, Chapter 3.2] only when the cost on the links satisfy triangle inequality.

Tree-topology radio access network design in UMTS cellular networks has been studied in [36, 37, 43, 67]. The multiple-RNS design problem is considered in [36], [37] and [67] deal with the single-RNS version of the problem, and [43] proposes an approach to solve both problems. Note that (as we indicated earlier) both problems are algorithmically equivalent. Since all these papers use the simulated annealing technique, the quality of the solutions depends on the duration of the execution.

Fault-tolerance in access networks is considered in [37] and [67]. In both papers the cost model is very similar to the one adopted here, and the cost function can handle both wired (e.g., leased-line, fiber, coax) and wireless (microwave) interconnections. However, the constraints of the models used in these papers are different from ours. It is assumed that each base station specifies its level in the tree and a uniform out-degree bound is used for all nodes.

Tree-topology-based design for access networks has also been discussed in [43]. In this work the authors proposed a simulated annealing based algorithms compared with a lower bound for the single-tree version of the problem. Using this Lagrangian relaxation-based lower
bound, a branch-and-bound method is proposed to compute the theoretical optimal solution to this problem for networks of small sizes. The cost model described in [43] has two important characterizations. The first one is that the cost function also depends on the level in which the base station is located in the tree. Secondly, its objective function is defined as the sum of the connection costs of the tree (rather than the sum of the cost of the paths between the RNC and each of the base stations, as studied in this chapter). Since connections closer to the RNC aggregate more traffic, the cost model should capture the flow of the traffic throughout the tree and therefore the cost model of [43] does not reflect this behavior of traffic. Moreover, optimal solutions for these two different objectives can be far from each other by a factor of $\Theta(n)$ (as in the case of unit-weight complete graph on $n$ vertices, all have a degree bound of two). From an algorithmically point of view, if the objective function is a minimization the total sum of connection costs, the problem can be studied under the framework of the well-known bounded-degree minimum spanning tree problem (e.g., [34], [28], [60]).

Several problems can be viewed as a generalization of the BDRT problem, where there are no limitations on the degree bounds. The minimum routing cost spanning tree problem, asks for a spanning tree where the sum of the routing cost is taken over all pairs of nodes and not only from the root to all other nodes.

Finding a spanning tree of minimum routing cost in general weighted undirected graphs is NP-hard [42]. (Notice that the “single-source” version of the problem without the degree constraints is polynomial-time solvable and can be seen as the single-source shortest path problem.) Wu et al. [76] showed that finding a minimum routing cost tree in a general weighted graph $G$ is equivalent to solving the same problem on a complete graph in which edge weights satisfy triangle inequality. This result implies that the minimum routing cost spanning tree problem in a metric space is also NP-hard. The best result known today for this problem is a polynomial-time approximation scheme (PTAS) due to Wu et al. [76]. In this paper, the authors show that this problem has an $(1 + \epsilon)$-approximate solution which runs in time $O(n^{2\lceil\frac{2}{\epsilon}\rceil - 2})$.

Hu [40] introduced a generalization of the minimum routing cost spanning tree problem that he called optimum communication spanning trees. In this problem, in addition to the weight on edges, a requirement value $r(v_i, v_j)$ is specified for every pair of vertices $v_i, v_j$. The communication cost between a pair of vertices in a given spanning tree is the cost of the path between them in the tree multiplied by their requirement $r(v_i, v_j)$. The communication cost of the tree is the sum of all pairwise communication costs. Thus, the routing cost is a special case of the communication cost when all the requirements are one.

Several $O(\log^2 n)$-approximation algorithms for the metric case of the minimum communication cost spanning tree problem are presented in [76] and [59, 61]. This problem is shown to be MAX SNP-hard [61], implying that a PTAS can not be achieved unless P = NP.

In Section 3.4 we present an extension to our model which uses similar flow requirements. This extension can be viewed as a single source, bounded-degree version of the minimum communication cost spanning tree problem.
3.3 The bounded-degree minimum routing cost spanning tree problem

The important goal of efficient planning of access networks is beyond our reach since this problem is NP-hard, as we mentioned before. In this chapter we use two approaches for coping with hard optimization problem: approximation algorithms and heuristics. Instead of finding an optimal solution, an approximation algorithm settles for a near, yet provable, optimal solution for every instance. Heuristic algorithms, on the other hand, work well on many instances, though not necessarily on all instances.

Unfortunately, it is not even possible to design polynomial-time approximation algorithm for BDRT, unless P = NP. Such $\gamma(n)$-approximation algorithm will solve the Hamiltonian path problem, for any computable function $\gamma(n)$. Given a graph $G$ on $n$ vertices we can transform it to an instance $G'$ of BDRT such that $G$ has a Hamiltonian path connecting $r$ and a vertex $v$ if and only if $G'$ has an optimal solution for BDRT of value $\xi(n) = 1 + 2 + \ldots + (n - 1) = \frac{1}{2}n(n - 1)$. This transformation can be done by assigning a unit weight to the edges of $G$, and a weight of $\gamma(n) \cdot \xi(n)$ to non-edges so as to obtain a complete graph $G'$. Degree bounds for $r$ and $v$ are 1 and for all other vertices in $G'$ degree bounds are taken to be two, and $r$ is fixed to be the root. We can now state the following theorem.

**Theorem 3.3.1.** Unless $P = NP$, there is no polynomial-time approximation algorithm for BDRT.

*Proof.* Assume, for contradiction, that there is a polynomial-time $\gamma(n)$-approximation algorithm, $A$, for BDRT. We show that $A$ can be used for deciding the Hamiltonian path problem in polynomial time, thus implying $P = NP$.

The main idea is a reduction from the Hamiltonian path problem to BDRT, that transforms a graph $G$ on $n$ vertices to an instance $G'$ of BDRT such that if $G$ has a Hamiltonian path, then the cost of an optimal solution to BDRT in $G'$ is $\xi(n)$, and if $G$ does not have a Hamiltonian path, then an optimal BDRT solution in $G'$ is of cost greater than $\gamma(n) \cdot \xi(n)$, where $\xi(n) = \frac{1}{2} \left( \left\lfloor \frac{n-1}{2} \right\rfloor^2 + \left\lceil \frac{n-1}{2} \right\rceil^2 + n - 1 \right)$.

The reduction is as follows. Assign a unit weight to the edges of $G$, and a weight of $\gamma(n) \cdot \xi(n)$ to non-edges, so as to obtain a complete graph $G'$. Degree bounds for each of the vertices in $G'$ are taken to be two, and one of the vertices, $r$, of $G'$ is fixed to be the root. Now, if $G$ has a Hamiltonian path, then the corresponding solution to BDRT in $G'$ is a tree, rooted at $r$, having two branches of sizes $\left\lfloor \frac{n-1}{2} \right\rfloor$ and $\left\lceil \frac{n-1}{2} \right\rceil$. This solution has a cost $\xi(n)$. On the other hand, if $G$ has no Hamiltonian path, any solution to BDRT in $G'$ must use an edge of weight $\gamma(n) \cdot \xi(n)$, and therefore has cost greater than $\gamma(n) \cdot \xi(n)$.

When considering planning of access networks in real-life applications, the weights on the edges in BDRT typically satisfy triangle inequality. Notice that in order to obtain the above inapproximability result we had to use very large edge weights that indeed violate triangle inequality. If we restrict ourselves to metric instances of BDRT (abbreviated *metric BDRT*) in which edge weights satisfy triangle inequality, the problem remains NP-hard (even for...
\[ b(v) \geq 3, \text{ for all } v \in V, \] but as we shall see in the following, it is no longer impossible to approximate the optimal solution.

**Theorem 3.3.2.** The metric BDRT with \( b = b(v) \geq 3 \), for every \( v \in V \), is NP-hard.

**Proof.** The proof is via a reduction from the Hamiltonian path problem. Given an instance \( G = (V, E) \) of the Hamiltonian path problem, we can construct an instance \( G' = (V', E') \) of metric BDRT with \( b = b(v) \geq 3 \), for every \( v \in V \), such that an optimal solution to BDRT on \( G' \) gives an optimal solution to the Hamiltonian path problem on \( G \).

Any instance \( G \) to the Hamiltonian path problem can be transformed to an instance to BDRT as follows: we construct \( G' \) as the complete graph on the vertices \( V \) of \( G \). We assign each edge \( e \) of this complete graph a weight of one if \( e \in E \), and a weight of two otherwise. In addition, we connect \( b - 2 \) more leaves to every vertex of the complete graph and define their weights to be 1. That is, \( G' \) has \( |V|((b - 2) + 1) \) vertices and \( |V|^2 + |V|(b - 2) \) edges with weights given in a metric space. Let \( r \) be the median of \( G' \), i.e., the vertex with minimum total distances to all vertices and we set the degree constraints for all vertices in \( V' \) to be \( b \) (\( \geq 3 \)). Now if \( G \) has a Hamiltonian path, then the corresponding solution to the BDRT in \( G' \) is a tree, rooted at \( r \), having a cost of

\[ \chi = b + \frac{b - 1}{2} \left( \left\lfloor \frac{n - 1}{2} \right\rfloor^2 + \left\lceil \frac{n - 1}{2} \right\rceil^2 + n - 5 \right), \]

where \( n = |V| \). On the other hand, if \( G \) has no Hamiltonian path, any solution to BDRT on \( G' \) must use an edge of weight 2, and therefore has cost greater than \( \chi \).

Notice that \( G' \) is not defined as a complete graph. Let graph \( G'' \) be the metric completion of \( G' \), i.e., the weights \( \{w(i, j)\}_{i, j \in V'} \) are defined to be equal to the length of the shortest-path connecting vertices \( i \) and \( j \) in \( G' \). Graph \( G'' \) gives the same hardness result, completing the proof. \( \Box \)

### 3.3.1 When the degree constraints are the lowest possible

Consider the case in which the root vertex has a degree constraint of \( k \) and all other vertices have degree bounds of 2, for every integer \( k \geq 1 \). This case of metric BDRT is also known as the \textit{k-Traveling Repairman Problem}. In this problem we are given \( k \) repairmen residing in a common depot \( r \) and \( n \) customers sitting in some metric space at prescribed distances from each other and the depot. The goal is to find tours on which to send the repairmen that minimize the average time a customer has to wait for a repairman to arrive, while making sure that all customers are served.

The case where \( k = 1 \) has been studied in a number of previous papers, and is often called simply the \textit{minimum latency problem} or the \textit{traveling repairman problem}. It is NP-hard for a general metric [62]. The best result known today for the \( k \)-repairman problem is due to Fakcharoenphol et al. [26]. For the general case they gave an \( 8.497\alpha \)-approximation algorithm, where \( \alpha \) denotes the best achievable approximation factor for the problem of finding the minimum-cost rooted tree spanning \textit{any} \( i \) vertices (\( i \)-MST problem). For the case \( k = 1 \)
Algorithm 3.1 BDRT-approximation

1: Construct a shortest-path tree, with root \( r \), on the input complete (metric) graph; renumber the vertices so that \( v_i, i = 1, 2, \ldots, n \), is the \( i \)th closest vertex to the root.
2: Set \( v_1 \) to be the root of the tree \( T \); \( i \leftarrow 1 \).
3: while \( T \) is not a spanning tree do
4: Pick the \( \xi(v_i) \) vertices of least indices and assign them, from the left most child to the rightmost, as the children of vertex \( v_i \), where \( \xi(v_i) = b(v_i) \), for \( i = 1 \), and \( \xi(v_i) = b(v_i) - 1 \) otherwise.
5: \( i \leftarrow i + 1 \)
6: end while

they present a 3.59\( \alpha \)-approximation algorithm. Notice that the best approximation factor known today for the \( i \)-MST problem (i.e., the current lowest \( \alpha \)) is 2 (due to Garg [32]).

Finally, since the case of \( k = 1 \) is known to be MAX-SNP-hard [4] we have the following observation on the hardness of approximation of BDRT.

**Theorem 3.3.3.** The metric BDRT is MAX-SNP-hard hence cannot have a PTAS unless \( P = NP \).

For the rest of the chapter we assume that \( b(v) \geq 3 \) for all \( v \in V \).

### 3.3.2 An \( O(\log n) \)-approximation algorithm

Next we present an \( O(\log n) \)-approximation algorithm for the metric version of BDRT, where the degree constraints for all vertices are assumed to be greater than or equal to 3. The lower bound we use is the cost of the shortest-path tree, rooted at \( r \), of the input graph \( G \). Nevertheless, the total cost of a shortest-path tree is not unique, in general. Graphs can have several shortest-path trees (rooted at the same vertex) of different costs yet each preserves the shortest-distance between the root and each of the vertices. However, when edge weights satisfy triangle inequality, the shortest-path tree is unique, drawn as a star centered at the root vertex. Obviously, this solution is a lower bound for any instance of BDRT, since every edge of this star is counted only once in the total cost.

Our approximation algorithm (Algorithm 3.1), has two phases. First, the shortest-path tree rooted at \( r \) is constructed, and the vertices are renumbered by their distance from the root on the shortest-path tree (\( v_1 \) is the root itself, \( v_2 \) is the closest vertex to the root on the input graph, \( v_3 \) is the second closest vertex, and so on).

In the second phase, the output tree is constructed meeting the degree constraints of all vertices. The algorithm starts at the root vertex \( r \), picks the \( b(r) \) vertices of least index and assigns them as its children, from the left most child to the rightmost one. Moving to the next level in the constructed tree, for every vertex \( v \), the algorithm picks the \( b(v) - 1 \) unpicked vertices of least index and assigns them as its children, from the left most child to the rightmost one. This process terminates when the tree contains all the vertices of \( G \).

Before bounding the cost of the tree constructed by Algorithm 3.1, let us consider a concrete example. Let \( b(v) = 3 \) for every vertex \( v \) of \( G \), meaning that the output tree is of
the largest height, \( h \). Since there is one vertex in the 0th level of this tree, three vertices in the first level, \( 3 \cdot 2 \) vertices in the second level, and \( 3 \cdot 2^{\ell-1} \) vertices in the \( \ell \)-th level, the height \( h \) is \( \lceil \log \frac{n+2}{3} \rceil \leq \log n \). Now, consider for example, vertex \( v_{12} \) in Figure 3.3, and let us bound its contribution to the total cost of the solution.

\[
d(v_{12}) = w(v_1, v_2) + w(v_2, v_5) + w(v_5, v_{12}) \\
\leq w(v_1, v_2) + (w(v_1, v_2) + w(v_1, v_5)) \\
+ (w(v_1, v_5) + w(v_1, v_{12})) \\
\leq w(v_1, v_{12}) + 2(w(v_1, v_2) + w(v_1, v_5)) \\
\leq w(v_1, v_{12}) + 2 \log n \cdot w(v_1, v_6) 
\]

(3.1)

Using the triangle inequality we can bound this contribution by (3.1). Finally, since \( v_6 \) is the last internal vertex in the tree (colored in grey in Figure 3.3), \( w(v_1, v_6) \geq w(v_1, v_j) \) for every \( j \leq 6 \), hence the heaviest path connecting the root to any vertex has cost at most \( 2 \log n \cdot w(v_1, v_6) \) (by (3.2)).

**Theorem 3.3.4.** Algorithm 3.1 is an \( O(\log n) \)-approximation algorithm for the metric BDRT with \( b(v) \geq 3 \), for every \( v \in V \).

**Proof.** Let \( T \) be the tree constructed by the algorithm, and let \( v_k \) be the last internal vertex of \( T \), that is, vertices \( v_{k+1}, \ldots, v_n \) are all leaves. Recall that the routing cost, \( d(v_i) \), of any vertex \( v_i \) is the sum of the weights along the unique path to the root \( v_1 \). In general, the routing cost can be computed as follows:

\[
d(v_i) = \sum_{j=0}^{\lfloor \log i \rfloor - 1} w(v_{\lfloor \frac{i}{2^j+1} \rfloor}, v_{\lfloor \frac{i}{2^j} \rfloor}) \\
\leq w(v_1, v_i) + 2 \sum_{j=0}^{\lfloor \log i \rfloor - 1} w(v_1, v_{2^j}) \\
\leq w(v_1, v_i) + 2 \log n \cdot w(v_1, v_k), 
\]

(3.3)

(3.4)

(3.5)

where (3.4) is the result of triangle-inequality, (3.5) follows since \( w(v_1, v_k) \geq w(v_1, v_j) \) for every \( j \leq k \), and \( \log n \) is an upper bound on the largest height \( T \) can have.
Since the cost of the shortest-path tree of $G$ (with $v_1$ as its root), as computed in the first step of the algorithm, is a lower bound on the cost of the optimal solution, $OPT$, we have,

$$\sum_{i=1}^{n} w(v_1, v_i) \leq OPT.$$  \hfill (3.6)

However, since $b(v) = 3$, $T$ has $\lceil n^2 \rceil$ leaves and the shortest path from each leaf is no less than $w(v_1, v_k)$, we have

$$\left\lceil \frac{n}{2} \right\rceil w(v_1, v_k) \leq OPT.$$  \hfill (3.7)

Finally, from the above we get,

$$\sum_{i=1}^{n} d(v_i) \leq \sum_{i=1}^{n} \left( w(v_1, v_i) + 2 \log n \cdot w(v_1, v_k) \right)$$

$$\leq OPT + 4 \log n \cdot OPT$$  \hfill (3.9)

$$= O(\log n) \cdot OPT.$$  \hfill (3.10)

When designing approximation algorithms one might be interested in improving the performance guarantee of the suggested algorithm. We next show that the $O(\log n)$-factor of Algorithm 3.1 is tight, as follows by the next example.

Consider a complete graph corresponding to a set of points on the real line (Figure 3.4). The points are divided into groups $\{G_i\}_{i \geq 0}$ as follows: $G_0$ contains only the origin, $G_1$ contains 2 points at distance 1 and $1 + \epsilon$ from the origin, on the right-side. $G_2$ contains 4 points at distances $1 + 2\epsilon$, $1 + 3\epsilon$, $1 + 4\epsilon$, and $1 + 5\epsilon$ from the origin, on the left-side. In general, $G_i$ contains $2^i$ points at distances $\{1 + (2^i - 2)\epsilon, \ldots, 1 + (2^{i+1} - 3)\epsilon\}$ from the origin, to from the left (right) side of the origin if $i$ is odd (even). Since distances are computed on the real line, the weights clearly satisfy triangle inequality, and the costs of a shortest-path on the constructed tree are preserved and not affected by the number of edges on the path.

![Figure 3.4: A tight example for the $O(\log n)$-approximation algorithm](image)

Technion - Computer Science Department - Ph.D. Thesis PHD-2008-02 - 2008
Taking $\epsilon = 1/n$ ensures that the solution obtained by the algorithm is of cost $\Theta(n \log n)$ (Figure 3.4(b)). However, the optimal solution of BDRT for this instance is of cost $\Theta(2n)$, as shown in Figure 3.4(a). This shows that the performance guarantee of Algorithm $A'$ is tight.

3.3.3 Heuristic algorithms

In this section we concentrate on a family of greedy heuristics for solving BDRT. This is perhaps the most common approach taken in practice. Since heuristics work well on many instances, though not on all of them, they cannot be rigourously analyzed. However, the heuristics presented here are based on insights obtained from our $O(\log n)$-approximation algorithm studied in the previous section.

The following algorithms work in a similar way: They start at the root vertex $r$, pick the best $b(r)$ vertices as its children, from the left most child to the rightmost one, and then move to the next level in the constructed tree. For every vertex $v$ ($\neq r$), the algorithm picks the best $b(v) - 1$ unpicked vertices and assigns them as its children, until spanning all the vertices of $G$. By “best” we mean the most preferred vertices according to a given criterion. We note that ties are broken arbitrarily.

The first greedy algorithm, called GA$_1$, picks the vertices $v$ in an increasing order of their ratio $d_T(r, v)/b(v)$, where $d_T(r, v)$ is the cost of the shortest-path connecting $v$ to $r$ in the constructed tree $T$. Although this algorithm might seem as the most natural one, its performance guarantee can be bad as $\Omega(n)$ as described in the following example.

**Observation 3.3.5.** There exist instances of the metric version of BDRT for which the cost of the solution, produced by GA$_1$, is within a factor of $\Omega(n)$ of the optimum, where $b(v) \geq 3$ for all vertices $v$.

**Proof.** Consider a set of $n + 4$ nodes given on the real line with the origin as the root (Figure 3.5). In this example, a set of nodes $V = \{v_1, \ldots, v_n\}$ is located at distance of 3 from the origin and a set $U = \{u_1, u_2, u_3\}$ is located at distance $\frac{n}{2}$. Each node $v \in V \cup \{0\}$ has a degree constraint of $b(v) = 3$ while nodes in $U$ have degree constraints of $n$.

It is not hard to verify that the optimal solution for this instance has a cost of $O(n)$ while GA$_1$ builds a tree of cost $O(n^2)$.

The second algorithm, called GA$_2$, is very similar to GA$_1$ but here vertices are picked in an increasing order of their ratio $d_T(r, v)/(b(v))^2$. This algorithm emphasizes the importance of the degree bound, in particular for those vertices with the same $d_T(r, v)/b(v)$-ratio.

The third algorithm, GA$_3$, picks the vertices $v$ in an increasing order of their ratio $w(u, v)/b(v)$. Given a partially constructed tree, we denote $u$ as the last vertex that is

![Figure 3.5: A bad example for GA$_1$](image-url)
already picked and joined to the tree. So, we are now ready to assign \( u \) up to \( b(u) - 1 \) children. This algorithm, as opposed to \( \text{GA}_1 \), selects the vertices \( v \) that are most closely to \( u \) in \( G \) having the largest degree constraints \( b(v) \). Notice that both \( \text{GA}_2 \) and \( \text{GA}_3 \) also have the same worse performance guarantee; this can be shown in a similar way as done for \( \text{GA}_1 \) (Observation 3.3.5).

We call our approximation algorithm (Algorithm 3.1) \( \text{GA}_4 \) and denote the following similar version of it as \( \text{GA}_5 \). This algorithm picks the vertices \( v_i \) in an increasing order of their ratio \( v_i/b(v_i) \), where \( v_i, i = 1, \ldots, n \), is the \( i \)th vertex closest to the root in the input graph \( G \). In other words, \( \{v_i\}_{i=1}^n \) are the vertices ordered as in the first phase of Algorithm \( A' \).

### A sample execution

A sample execution of the above five algorithms for BDRT is described in Figure 3.6. The input is a complete graph \( G \) on 7 vertices, given by the adjacency matrix below. The root is chosen to be vertex \( a \) and the degree bounds are defined as \( b(b) = b(g) = 2 \), while other vertices have a degree bound of 3.

In this example, the solutions produced by algorithms \( \text{GA}_1 \), \( \text{GA}_2 \), and \( \text{GA}_5 \) have a cost of 31 (Parts (a), (b), and (e), respectively), algorithm \( \text{GA}_3 \) achieves a cost of 30 (Part (c)), and algorithm \( \text{GA}_4 \), which is the approximation algorithm, produces a solution of cost 29 (Part (d)). Notice that the theoretical lower bound in this case, namely the cost of the shortest-path tree of \( G \), rooted at vertex \( a \), is 25 (Part (f)).

### 3.4 Extensions

In this section we extend our model to be more sensitive to traffic requirements of the different base stations. Notice that 4G networks, for example, will use a variety of technologies and it is very likely that some cells would support a large area and a large density of traffic, while other cells may be designed for very small traffic densities. Given an instance of BDRT, we assume now that each base station \( i \in I \) has traffic requirement \( t_i \), representing its expected traffic load. Since a base station is connected to an RNC via a path of base stations, its traffic is aggregated along that path. The routing cost of the path connecting base station \( i \) to RNC

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>6</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>c</td>
<td>3</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>d</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>e</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>f</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>4</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>g</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.1: Adjacency matrix for the sample execution
Figure 3.6: Sample execution of the five algorithms.

$r$ in the tree $T$ is defined to be $t_i \cdot d_T(r, i)$. The generalized bounded-degree minimum routing cost spanning tree problem (GBDRT) is to find a spanning tree, rooted at a given root $r$, that meets the degree constraints $b(v)$, for all $v \in V$ and the cost

$$\sum_{i \in I} t_i \cdot d_T(r, i)$$

(3.11)

is minimized.

Obviously, BDRT is a special case of GBDRT, by taking $t_i = 1$, for all $i \in I$. Moreover, our approximation algorithm (Algorithm 3.1) when applied to an instance of GBDRT, gives the same performance guarantee. Notice that the lower bound in this case is the star, centered at the root vertex, computed via (3.11). Hence we have the following:

**Theorem 3.4.1.** The metric generalized bounded-degree minimum routing cost spanning tree problem (GBDRT) with $b(v) \geq 3$, for all $v \in V$, is approximable within a factor of $O(\log n)$ of the optimum.

A similar generalization can also be applied to the heuristic algorithms described in Section 3.3.3. In this case the greedy algorithms decrease their criteria for a selection of vertex $v$ by a factor of $t_v$, where $v$ is a vertex corresponds to a base station $v$ and $t_v$ is its traffic requirement.

### 3.5 Simulation results

In Section 3.3, we defined five algorithmic solutions for both BDRT and its generalization GBDRT. The first method was our approximation algorithm (Section 3.3.2), and the other four are heuristics (Section 3.3.3). In order to determine a good approach for solving BDRT and GBDRT in practice, we conducted two separate sets of simulations to test BDRT and
GBDRT. In addition, a set of simulation testing the distance from the star lower bound using uniform weights.

Each set of simulations was run on a random complete graph on \( n \) vertices (with values \( n = 10, 20, \ldots , 200 \)) where the (integer) degree bounds were selected uniformly at random between 3 and 8.

When simulating BDRT and GBDRT, edge weights are sampled from an \( n \times n \)-square in the Euclidean plain. Each of the five algorithms was executed 1500 times, and the average cost as well as the standard deviation values were recorded. In the case of GBDRT, traffic loads are uniformly taken from the set \( \{1, 2, 4, 8, \ldots , 128\} \).

Figure 3.7(a) describes the results for the uniform case of BDRT. The results indicate that the optimal solutions, produced by algorithms \( GA_1, GA_2, GA_3, \) and \( GA_5 \), are of cost within a factor of 3.4 from the lower bound. However, \( GA_4 \) was far from the optimal solution by at most 23% (in the case of \( n = 80 \)).

It is not hard to verify that in the uniform case (Figure 3.7(a)), algorithms \( GA_1, GA_2, GA_3, \) and \( GA_5 \) perform the same. Consider the greedy algorithm that picks the vertices of \( G \) in a decreasing order of their degree bounds. Clearly, changing the position of a vertex will not decrease the height of the vertex and hence the total cost of the tree will not be smaller. Assume, for example, that a vertex of level \( h \) in the constructed tree is considered to be selected by these algorithms. Since all candidates examined both by \( GA_1 \) and \( GA_2 \) are of length \( h \) from the root, the chosen vertex will be the one with the highest degree constraint. Now, since edge-weights are uniform, Algorithm \( GA_3 \) will pick the highest degree-constrained vertex as well. Finally, all the vertices in the shortest-path tree of the input graph are of the same distance from the root and therefore, the vertex of highest degree-constraint will be selected also by Algorithm \( GA_5 \). However, Algorithm \( GA_4 \) does not involve degree-constraints in its selection criteria hence it yields a different solution.

Figure 3.7(b) summarizes the results for BDRT. In this case algorithms \( GA_1 \) and \( GA_3 \) performed better than their counterparts. These algorithms achieve solutions that are within
a factor of 1.28 from the lower bound. However, all five algorithms reached average costs of up to a factor of 1.49 of the lower bound, significantly far from the worst-case $O(\log n)$-factor. Standard deviation of the runs of these algorithms were between 0.14 to 0.28 for small sized graphs (Table 3.2 on Page 59).

Notice that the performance of BDRT simulations (Figure 3.7(b)) outperform the results of the uniform case (Figure 3.7(a)) by a factor of 2.7 as in the case of GA3. The main reason for this interesting behavior is that when a vertex is placed further down in the tree its distance to the root increases. In the non-uniform cases vertices in these positions would have lower weights and thus less effects on the overall cost.

Figure 3.8 summarizes the results for GBDRT. In this case, algorithms GA1, GA2, GA3, and GA5 achieve solutions that are within a factor of 1.67 from the lower bound (GA3 has reached a factor of only 1.33). Algorithm GA4 was far approximately 2.7 times the lower bound. Standard deviation of the runs of GA3 were between 0.56 to 0.22 (Table 3.3 on Page 59).

In addition, the worst-case running time of the algorithms, for all cases, was approximately two seconds for $n = 200$, on a Pentium M machine, 1.4 GHz, and 256 Mb of RAM.

3.6 Conclusions and open problems

Planning radio access networks for future cellular systems requires a replacement of the commonly used star based architecture (in which an RNC is connected to a set of base stations via direct links) with a new tree-topology structure. In the new architecture a base station can be connected to an RNC via other base stations, resulting in a complex network design problem.

We studied five algorithms for solving the metric version of BDRT. These methods involve both an approximation algorithm and a family of greedy heuristics. Our results indicate that
a combination\(^1\) of a greedy heuristic (GA\(_3\)), that picks the vertices \(v\) in an increasing order of their ratio \(w(u,v)/b(v)\) and our \(O(\log n)\)-approximation algorithm generates a solution, which produces a close-to-optimal result in practical scenarios with a guaranteed worst case bound. Moreover, this solution can be efficiently computed for networks of large size.

Finally, two open problems are of special interest. Is there a better lower bound than the shortest-path tree of the input graph? This might be a crucial challenging question towards improving the approximation factor for the metric version of BDRT. Notice that such an approximation algorithm cannot be achieved if the lower bound is indeed the cost of the shortest-path tree of the input graph. To see this consider the uniform edge-weights instance when all vertices have degree bound of 3. The optimal solution is a tree of cost \(O(n \log n)\) while the shortest-path tree has cost of \(n - 1\).

As we described in this chapter, when designing tree-topology radio access networks, the communication between base stations and RNCs causes a certain amount of delay in communication. In order to reduce this delay, the longest path between a base station to an RNC on every component should be of bounded length.

We can define a metric version of BDRT together with a \textit{depth constraint} as the generalized version of BDRT. The second interesting problem is how well can this generalization be approximated?

\(^1\)In the sense of executing both algorithms and choosing the lower cost solution.
Table 3.2: Algorithms for solving BDRT

<table>
<thead>
<tr>
<th>n</th>
<th>GA_1</th>
<th>GA_2</th>
<th>GA_3</th>
<th>GA_4</th>
<th>GA_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.182 (0.26)</td>
<td>1.215 (0.28)</td>
<td>1.182 (0.26)</td>
<td>1.168 (0.25)</td>
<td>1.184 (0.26)</td>
</tr>
<tr>
<td>20</td>
<td>1.231 (0.21)</td>
<td>1.227 (0.21)</td>
<td>1.239 (0.20)</td>
<td>1.255 (0.21)</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>1.254 (0.19)</td>
<td>1.304 (0.20)</td>
<td>1.305 (0.19)</td>
<td>1.310 (0.19)</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>1.281 (0.18)</td>
<td>1.342 (0.20)</td>
<td>1.344 (0.18)</td>
<td>1.356 (0.19)</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1.290 (0.18)</td>
<td>1.360 (0.20)</td>
<td>1.277 (0.18)</td>
<td>1.370 (0.18)</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>1.294 (0.17)</td>
<td>1.368 (0.19)</td>
<td>1.279 (0.17)</td>
<td>1.387 (0.17)</td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>1.302 (0.16)</td>
<td>1.378 (0.18)</td>
<td>1.287 (0.16)</td>
<td>1.411 (0.16)</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>1.289 (0.16)</td>
<td>1.359 (0.17)</td>
<td>1.273 (0.16)</td>
<td>1.404 (0.15)</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>1.286 (0.16)</td>
<td>1.356 (0.17)</td>
<td>1.272 (0.16)</td>
<td>1.425 (0.15)</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1.287 (0.15)</td>
<td>1.354 (0.16)</td>
<td>1.273 (0.16)</td>
<td>1.460 (0.14)</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>1.287 (0.15)</td>
<td>1.357 (0.16)</td>
<td>1.272 (0.16)</td>
<td>1.445 (0.15)</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>1.282 (0.15)</td>
<td>1.350 (0.16)</td>
<td>1.269 (0.16)</td>
<td>1.450 (0.15)</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>1.274 (0.15)</td>
<td>1.340 (0.15)</td>
<td>1.261 (0.15)</td>
<td>1.444 (0.14)</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>1.279 (0.15)</td>
<td>1.344 (0.15)</td>
<td>1.267 (0.16)</td>
<td>1.460 (0.14)</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>1.280 (0.15)</td>
<td>1.341 (0.15)</td>
<td>1.267 (0.16)</td>
<td>1.467 (0.14)</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>1.283 (0.15)</td>
<td>1.342 (0.15)</td>
<td>1.269 (0.16)</td>
<td>1.479 (0.14)</td>
<td></td>
</tr>
<tr>
<td>170</td>
<td>1.276 (0.15)</td>
<td>1.337 (0.15)</td>
<td>1.264 (0.15)</td>
<td>1.475 (0.14)</td>
<td></td>
</tr>
<tr>
<td>180</td>
<td>1.283 (0.15)</td>
<td>1.388 (0.15)</td>
<td>1.271 (0.15)</td>
<td>1.484 (0.13)</td>
<td></td>
</tr>
<tr>
<td>190</td>
<td>1.281 (0.15)</td>
<td>1.339 (0.15)</td>
<td>1.270 (0.16)</td>
<td>1.491 (0.14)</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>1.284 (0.15)</td>
<td>1.341 (0.15)</td>
<td>1.273 (0.16)</td>
<td>1.489 (0.14)</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Algorithms for solving GBDRT

<table>
<thead>
<tr>
<th>n</th>
<th>GA_1</th>
<th>GA_2</th>
<th>GA_3</th>
<th>GA_4</th>
<th>GA_5</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>1.082 (0.56)</td>
<td>1.089 (0.57)</td>
<td>1.081 (0.56)</td>
<td>1.652 (0.87)</td>
<td>1.083 (0.56)</td>
</tr>
<tr>
<td>20</td>
<td>1.198 (0.45)</td>
<td>1.216 (0.47)</td>
<td>1.189 (0.45)</td>
<td>1.931 (0.73)</td>
<td>1.206 (0.46)</td>
</tr>
<tr>
<td>30</td>
<td>1.272 (0.41)</td>
<td>1.297 (0.42)</td>
<td>1.252 (0.40)</td>
<td>2.213 (0.71)</td>
<td>1.294 (0.41)</td>
</tr>
<tr>
<td>40</td>
<td>1.312 (0.36)</td>
<td>1.341 (0.37)</td>
<td>1.279 (0.35)</td>
<td>2.265 (0.64)</td>
<td>1.351 (0.37)</td>
</tr>
<tr>
<td>50</td>
<td>1.354 (0.33)</td>
<td>1.379 (0.34)</td>
<td>1.304 (0.32)</td>
<td>2.324 (0.58)</td>
<td>1.408 (0.35)</td>
</tr>
<tr>
<td>60</td>
<td>1.376 (0.31)</td>
<td>1.412 (0.33)</td>
<td>1.313 (0.30)</td>
<td>2.386 (0.57)</td>
<td>1.443 (0.33)</td>
</tr>
<tr>
<td>70</td>
<td>1.395 (0.30)</td>
<td>1.424 (0.31)</td>
<td>1.320 (0.29)</td>
<td>2.412 (0.52)</td>
<td>1.476 (0.31)</td>
</tr>
<tr>
<td>80</td>
<td>1.416 (0.29)</td>
<td>1.455 (0.31)</td>
<td>1.330 (0.27)</td>
<td>2.497 (0.53)</td>
<td>1.512 (0.31)</td>
</tr>
<tr>
<td>90</td>
<td>1.416 (0.27)</td>
<td>1.453 (0.29)</td>
<td>1.324 (0.26)</td>
<td>2.519 (0.50)</td>
<td>1.518 (0.29)</td>
</tr>
<tr>
<td>100</td>
<td>1.433 (0.27)</td>
<td>1.474 (0.29)</td>
<td>1.332 (0.25)</td>
<td>2.588 (0.48)</td>
<td>1.549 (0.28)</td>
</tr>
<tr>
<td>110</td>
<td>1.432 (0.25)</td>
<td>1.473 (0.26)</td>
<td>1.328 (0.23)</td>
<td>2.601 (0.46)</td>
<td>1.558 (0.26)</td>
</tr>
<tr>
<td>120</td>
<td>1.441 (0.26)</td>
<td>1.484 (0.27)</td>
<td>1.330 (0.23)</td>
<td>2.594 (0.44)</td>
<td>1.584 (0.27)</td>
</tr>
<tr>
<td>130</td>
<td>1.450 (0.25)</td>
<td>1.498 (0.26)</td>
<td>1.333 (0.23)</td>
<td>2.639 (0.44)</td>
<td>1.607 (0.27)</td>
</tr>
<tr>
<td>140</td>
<td>1.444 (0.24)</td>
<td>1.484 (0.26)</td>
<td>1.325 (0.22)</td>
<td>2.633 (0.42)</td>
<td>1.601 (0.26)</td>
</tr>
<tr>
<td>150</td>
<td>1.449 (0.24)</td>
<td>1.494 (0.25)</td>
<td>1.326 (0.22)</td>
<td>2.659 (0.42)</td>
<td>1.619 (0.26)</td>
</tr>
<tr>
<td>160</td>
<td>1.443 (0.23)</td>
<td>1.486 (0.24)</td>
<td>1.319 (0.21)</td>
<td>2.653 (0.42)</td>
<td>1.615 (0.25)</td>
</tr>
<tr>
<td>170</td>
<td>1.450 (0.23)</td>
<td>1.496 (0.24)</td>
<td>1.321 (0.21)</td>
<td>2.663 (0.39)</td>
<td>1.634 (0.24)</td>
</tr>
<tr>
<td>180</td>
<td>1.468 (0.23)</td>
<td>1.506 (0.24)</td>
<td>1.332 (0.21)</td>
<td>2.696 (0.41)</td>
<td>1.665 (0.25)</td>
</tr>
<tr>
<td>190</td>
<td>1.459 (0.23)</td>
<td>1.505 (0.24)</td>
<td>1.326 (0.21)</td>
<td>2.709 (0.39)</td>
<td>1.663 (0.24)</td>
</tr>
<tr>
<td>200</td>
<td>1.464 (0.22)</td>
<td>1.507 (0.22)</td>
<td>1.328 (0.20)</td>
<td>2.686 (0.38)</td>
<td>1.669 (0.23)</td>
</tr>
</tbody>
</table>
Chapter 4

Cell Selection in Future Cellular Networks

4.1 Introduction

The ability to provide services in a cost effective manner is one of the most important building blocks of competitive modern cellular systems. Usually, an operator would like to have a maximal utilization of the installed equipment, that is, to maximize the number of satisfied customers at any given point in time. This chapter addresses one of the basic problems in this domain, the cell selection mechanism that determines the base station (or base stations) that provides the service to a mobile station - a process that is performed when a mobile station joins the network (called cell selection), or when a mobile station is on the move in idle mode (called cell reselection, or cell change, in HSPA).

In most current cellular systems the cell selection process is done by a local procedure initialized by a mobile device according to the best detected SNR. In this process the mobile device measures the SNR to several base stations that are within radio range, maintains a “priority queue” of those that are best detected (called an active set), and sends an official service request to subscribe to base stations by their order in that queue. The mobile station is connected to the first base station that positively confirmed its request. Reasons for rejecting service requests may be handovers or drop-calls areas, where the capacity of the base station is nearly exhausted.

Consider for example the settings depicted in Figure 4.1. Assume that the best SNR for Mobile Station 1 (MS1) is detected from microcell A, and thus MS1 is being served by this cell. When Mobile Station 2 (MS2) arrives, its best SNR is also from microcell A, who is the only cell able to cover MS2. However, after serving MS1, microcell A does not have enough capacity to satisfy the demand of MS2 who is a heavy data client. However, if MS1 could be served by picocell B then both MS1 and MS2 could be served. Note that MS1 and MS2 could represent a cluster of clients. The example shows that the best-detected-SNR algorithm can be a factor of $\max\{\tilde{d}\}/\min\{\tilde{d}\}$ from an optimal cell assignment, where $\tilde{d}$ is the demand of any mobile station in the coverage area. Theoretically speaking, this ratio can be arbitrarily large.
This simple example illustrates the need for a global, rather then a local, cell selection solution that tries to maximize the global utilization of the network, and not just the SNR of a single user. In voice only networks, where base station capacities are considered to be high, sessions have limited duration, and user demands are uniform, this may not be a big barrier. That is, the current base station selection process results, in most cases, in a reasonable utilization of the network. However, in the forthcoming future cellular networks this may not be the case.

The increased number of base stations, and the variable bandwidth demand of mobile clients, will force operators to optimize the way the capacity of a base station is utilized. Unlike in previous generations, the ability of a base station to successfully satisfy the service demand of all its mobile clients would be highly limited and will mostly depend on its infrastructure restrictions, as well as on the service distribution among its mobile clients.

Another interesting aspect is the support for different QoS classes for the mobile stations, (e.g., gold, silver, or bronze). In such a case, the operator would like to have as many satisfied "gold" customers as possible, even if this means several unsatisfied "bronze" customers.

In this chapter we study the potential benefit of a new global cell selection mechanism, which should be contrasted with the current local mobile SNR-based decision protocol. In particular, we rigorously study the problem of maximizing the number of mobile stations that can be serviced by a given set of base stations in such a way that each of the serviced mobile stations has its minimal demand fully satisfied. We differentiate between two coverage paradigms: The first is cover-by-one where a mobile station can receive service from at most one base station. The second is cover-by-many, where we allow a mobile station to be simultaneously satisfied by more than one base station. This means that when a mobile stations has a relatively high demand (e.g., video-on-demand) in a sparse area (e.g., sea-shore), several base stations from its active set can participate in its demand satisfaction. This option is not available in third-generation networks (and not even in HSPA networks) since these networks have universal frequency reuse and the quality of a service a mobile station receives will be severely damaged by the derived co-channel interference. However, OFDMA-based technology systems and their derivatives are considered to be among the prime candidates for future cellular communication networks. The ability to satisfy the demand of a mobile station by more than one member of its active set is possible in these systems, as defined by the IEEE 802.16e standard. An important question in this context is whether cover-by-many
is indeed more powerful than cover-by-one, in the sense that it improves the ability of the network to satisfy more clients.

4.1.1 Our Contribution

This chapter presents a new approach for cell selection that is derived from the anticipated planned future technologies. To the best of our knowledge, despite recent extensive research done on future cellular networks planning and coverage optimization (e.g., [12, 57]), there is no explicit study in the literature discussing cell selection, nor are we aware of any research discussing the new IEEE 802.16e possibility of simultaneous coverage of mobile clients by more than one base station.

We model, in Section 4.2, the cell selection problem as an optimization problem called all-or-nothing demand maximization (AoNDM). We show that the general version of AoNDM cannot be approximated within a factor better than $|J|^{1-\epsilon}$, unless NP = ZPP, for any $\epsilon > 0$. Motivated by this result, we address a special case of the problem. Following practical scenarios, we define a restrictive version of AoNDM, the $r$-AoNDM problem, where the network satisfies the condition that $d(j) \leq r \cdot c(i)$, for every client $j \in J$ and every base station $i \in I$ that can potentially cover $j$. We show that even this special case of the problem is NP-hard. These results appear in Section 4.4.

We further present, in Section 4.4, two different algorithms for this problem. The first is a $\frac{1-r}{2-r}$-approximation algorithm, which uses the cover-by-one paradigm, i.e., every mobile station is covered by at most one base station. Note that this approximation guarantee is with regard to the optimal cover-by-many assignment. The second algorithm uses the cover-by-many paradigm, where a mobile station can be covered simultaneously by more than one base station. It is a careful refinement of the first algorithm, and has a performance guarantee of satisfying at least a $1-r$ fraction of an optimal assignment, at a price of increased running time.

In order to evaluate the practical differences between a global and a local mechanism for cell selection in future networks we conducted an extensive simulation study (Section 4.5). We compare between global mechanisms that are based on our approximation algorithms and the current best-SNR greedy cell selection protocol. We study the relative performance of these three algorithms under different conditions. In particular, we show that in a high-load capacity-constrained 4G-like network, where clients’ demands may be large with respect to cell capacity, global cell selection can achieve up to 20% better coverage than the current best-SNR greedy cell selection method.

4.2 Model and definitions

Consider a bipartite graph $G = (I, J, E)$ where $I = \{1, 2, \ldots, m\}$ is the set of base stations and $J = \{1, 2, \ldots, n\}$ is the set of mobile stations (or clients). Every client $j \in J$ has a non-negative demand $d(j)$, and a non-negative profit $p(j)$, and every base station $i \in I$ has a non-negative capacity $c(i)$. In addition, for every base station $i \in I$, the coverage area of $i$ is modeled by a subset $S_i \subseteq J$ of clients which can be serviced by $i$. The set of base stations $N(j) \subseteq I$
connected by edges to a client \( j \in J \), represents the active set of this client. We further extend the above definitions to sets of nodes, such that for every \( A \subseteq J \), \( d(A) = \sum_{j \in A} d(j) \) and \( p(A) = \sum_{j \in A} p(j) \), and for every \( B \subseteq I \), \( c(B) = \sum_{i \in B} c(i) \). Furthermore, given any \( A \subseteq J \), we let \( N(A) = \bigcup_{j \in A} N(j) \). Given a subset of clients \( S \subseteq J \), a cover plan for \( S \) is a weight function \( x : E \to \mathbb{R}^+ \), such that for every \( j \in S \), \( \sum_{i : (i,j) \in E} x(i,j) \geq d(j) \), and for every \( i \in I \), \( \sum_{j : (i,j) \in E} x(i,j) \leq c(i) \). Notice that such a restriction of \( \sum_{i : (i,j) \in E} x(i,j) \geq d(j) \), for every \( j \in S \), is also known as all-or-nothing-type of coverage. This means that clients that are partially satisfied are not considered to be covered (such a model appears, for example, in OFDMA-based networks where mobile stations have their slot requirements over a frame and these are not useful if not fulfilled).

The all-or-nothing demand maximization problem (AoNDM) is to find a subset of clients \( S \subseteq J \), and a cover plan \( x \) for \( S \), such that \( p(S) \) is maximized.

For \( i \in I \), we use \( x(i) = \sum_{j : (i,j) \in E} x(i,j) \), and for \( j \in J \), we use \( x(j) = \sum_{i : (i,j) \in E} x(i,j) \). As before, we extend these notations to sets of nodes, such that for every \( A \subseteq I \), \( x(A) = \sum_{i \in A} x(i) \), and for every \( B \subseteq J \), \( x(B) = \sum_{j \in B} x(j) \). We further extend this notation to subgraphs of \( G \), such that given any \( A \subseteq I \) and \( B \subseteq J \), \( x(A,B) = \sum_{(i,j) \in E \cap (A \times B)} x(i,j) \).

In addition, for every \( v \in I \cup J \) we denote by \( E(v) \) the set of edges with endpoint \( v \), and for every \( W \subseteq I \cup J \), let \( E(W) = \bigcup_{v \in W} E(v) \). We further denote for every \( A \subseteq I \) and \( B \subseteq J \), \( E(A,B) = \{(i,j) \in E \cap (A \times B)\} \).

Given any constant \( r < 1 \), we say an instance is \( r \)-restricted if for every \( (i,j) \in E \), \( d(j) \leq r \cdot c(i) \). We further define the problem of \( r \)-AoNDM as the AoNDM problem limited to \( r \)-restricted instances.

### 4.3 Related work

Cell selection has received much attention in recent years (e.g., [35, 54, 63, 64]) where research focused mainly on multiple-access techniques, as well as on power control schemes and handoff protocols [35, 63, 64].

In [35] a cell selection algorithm is presented where the goal is to determine the power allocations to the various users, as well as a cover-by-one allocation, so as to satisfy per-user SINR constraints. An HSPA-based handoff/cell-site selection technique is presented in [63, 64], where the objective is to maximize the number of connected mobile stations (very similar to our objective), and reaching the optimality of this objective is done via a new scheduling algorithm for this cellular system. All the above results did not take into account variable base station capacities nor mobile station bandwidth demands. In the case of [63, 64], this enables the authors to reduce their corresponding optimization problem to a polynomial-time solvable matching problem. As shown in our paper, when base station capacities and/or mobile stations’ demands are incorporated, this approach is no longer feasible.

An integrated model for optimal cell-site selection and frequency allocation is shown in [54], where the goal is to maximize the number of connected mobile stations, while maintaining quasi-independence of the radio based technology. The optimization problem in this model is shown to be NP-hard.
AoNDM is very closely related to the problem of planning future cellular networks under budget limitation as described in [11, 13]. In this problem, in addition to the input of AoNDM, we are given a set \( I \) of possible configuration of base stations (i.e., antenna type, geographic position, height, tilt, azimuth and any other relevant parameters of a base station antenna that together with the technology determine the coverage area of the antenna and the interference pattern) each has an opening cost \( w(i) \), a capacity \( c(i) \), and a coverage area represented by a set \( S_i \subseteq J \) of clients admissible to be covered (or satisfied) by it. When a client belongs to the coverage area of more than one base station, interference between the servicing stations may occur. These interferences are modeled by a penalty-based mechanism and may reduce the contribution of a base station to a client. The \textit{budgeted cell planning problem} asks for a subset of base stations \( I' \subseteq I \) whose cost does not exceed a given budget \( B \), and the total number of fully satisfied clients is maximized (See Chapter 2.3 on Page 20 of this thesis). Notice that in these settings, by taking the set \( I \) of base stations with zero opening costs, without interferences, we get a special case of AoNDM where all clients have the same profit. It was shown there that this problem cannot be approximated, unless \( P=NP \), and that a \( \frac{e\sqrt{3}}{3\sqrt{2}} \)-approximation algorithm exists for a special case of the problem where every set of \( k \) open base stations can fully satisfy at least \( k \) clients, for every integral value of \( k \).

The case where we restrict the clients in AoNDM to be satisfied by a \textit{single} base station belongs to the family of generalized assignment problems. Among this class of problems the most related problem to AoNDM is the \textit{separable assignment problem} (SAP) [29].

In this problem we are given a set \( U \) of \( m \) bins, a set \( H \) of \( n \) items, and a profit, \( f_{ij} \), for assigning item \( j \) to bin \( i \). The assignment constraints are such that every \( i \in U \) has a family \( \mathcal{T}_i \) of feasible subsets that can be packed in bin \( i \), such that \( \mathcal{T}_i \) is closed under taking subsets, i.e., if \( A \in \mathcal{T}_i \), then so is every subset of \( A \). The goal is to find an assignment of items to bins with the maximum aggregate profit.

The suggested SAP solution presented in [29], depends on an algorithm which solves the single-bin subproblem in SAP. Given a \( \beta \)-approximation algorithm for finding the highest profit packing of a single bin, they present a polynomial-time LP-rounding based \( (1-\frac{1}{e})\beta \)-approximation algorithm and a polynomial-time local search \( \frac{\beta}{\beta+1} - \epsilon \)- approximation algorithm, for any \( \epsilon > 0 \). Specifically, for all special cases of SAP that admit an approximation scheme for the single-bin problem, there exists an LP-based algorithm with a \( (1-\frac{1}{e} - \epsilon) \)-approximation guarantee, and a local search algorithm with a \( (\frac{1}{2} - \epsilon) \)-approximation guarantee.

This problem is a generalization of several well known problems. Among these problems are the \textit{maximum generalized assignment problem} (GAP), and the \textit{multiple knapsack problem} (MKP). In GAP we are given a set of bins with capacity constraints and a set of items that have a possibly different size and profit for each bin. We wish to pack a maximum-profit subset of items into the bins. MKP is the special case of GAP where the size and the profit of each item are the same for all the bins.

Shmoys and Tardos [65] give an LP-rounding based 2-approximation algorithm for the minimization version of GAP. However, Chekuri and Khanna [18] observed that a \( 1/2 \)-approx. algorithm for standard GAP is implicit in [65]. In addition, Chekuri and Khanna [18] develop
a PTAS for MKP and also classify the APX-hardness of GAP.

Another closely related problem is the all-or-nothing multicommodity flow problem discussed in [19] and [20]. In this problem we are given a capacitated undirected graph $G = (V, E, u)$ (where $u$ is the edge-capacity function) and set of $k$ pairs $(s_1, t_1), \ldots, (s_k, t_k)$. Each pair has a unit demand. The objective is to find a largest subset $S$ of $\{1, \ldots, k\}$ such that one can simultaneously route for every $i \in S$ one unit of flow between $s_i$ and $t_i$. It is straightforward to verify that the unit profit version of AoNDM is a special case of this problem.

It was shown that the all-or-nothing multicommodity flow problem can be approximated within an $O(\log^2 k)$ factor of the optimum [20]. On the other hand, for any $\epsilon > 0$, the problem cannot be approximated to within a factor of $O(\log^{\frac{1}{3}-\epsilon}|E|)$ of the optimum, unless $NP \subseteq ZPTIME(|V|^{\text{polylog}|V|})$ [15]. However, no special attention is given to specific network topologies (e.g., bipartite graphs, as in our case), and other special instances.

### 4.4 Approximating AoNDM and $r$-AoNDM problems

The important goal of efficient solution to AoNDM is beyond our reach since this problem is NP-hard, as we mentioned before. Moreover, it is not even possible to design a close-to-optimal polynomial-time approximation algorithm for AoNDM, unless $NP = ZPP$. To be precise, since a solution for the general version of AoNDM can be used to solve the Maximum Independent Set Problem in graphs (Problem GJ20 in [31]), and since the latter cannot be approximated within a factor better than $|J|^{1-\epsilon}$, unless $NP = ZPP$, for any $\epsilon > 0$, this hardness of approximation can be used as a lower bound for AoNDM, as stated by the following theorem.

**Theorem 4.4.1.** For any $\epsilon > 0$, AoNDM cannot be approximated to within a factor better than $|J|^{1-\epsilon}$, unless $NP = ZPP$.

**Proof.** We present a reduction from the Maximum-Size Independent Set (MIS) problem to AoNDM. Let $G = (V, E)$ be any input to MIS. Consider the bipartite graph $\tilde{G} = (I, J, \tilde{E})$, where $I = E$, $J = V$, and $\tilde{E} = \{(e, v) \in E \times V \mid v$ is an endpoint of $e\}$. For every $e \in V$, let $\delta(v)$ denote the degree of $v$ in $G$, and let $M = \max_v \delta(v)$. For every $j \in J$ we set $d(j) = \delta(j)$ and set $p(j) = 1$. Finally, we define for every $i \in I$, $c(i) = 1$.

Since all clients have unit profit, our goal is to maximize the number of clients which can be covered. Let $S$ be any subset of $J$, and let $x$ be any cover plan for $S$. For any $j \in S$, the overall capacity of the base stations connected to $j$ is

$$\sum_{i : (i, j) \in \tilde{E}} c(i) = \delta(j) = d(j).$$

It follows, that any client covered in $S$ uses all the capacity of the base stations in its range. Hence, a base station may contribute to the covering of at most one client, and in particular, any $e \in E$ can contribute to covering at most one of its endpoints. It follows that for any $S \subseteq V$, $S$ has a cover plan if and only if it is an independent set in $G$. Since for any $\epsilon > 0$, MIS cannot be approximated to within a factor better than $|V|^{1-\epsilon}$, unless $NP = ZPP$ [38], the same holds for AoNDM.

$\square$
Following practical scenarios, a restrictive version of AoNDM, the $r$-AoNDM problem defined before, where the network satisfies the condition that $d(j) \leq r \cdot c(i)$, for every client $j \in J$ and every base station $i \in I$ that can potentially cover $j$. Fortunately, this restricted version of AoNDM is still NP-hard but no longer “hard-to-approximate”, as stated next.

**Theorem 4.4.2.** For any fixed $r < 1$, the $r$-AoNDM problem is NP-hard, even if there is only one base station.

**Proof.** We show a reduction from the Knapsack problem, which is known to be NP-hard. Let $K$ be any instance to the Knapsack problem, which comprises of a set of elements $A$, and a knapsack of size $B$, such that every element $a \in A$ has a size $s(a)$, and a value $v(a)$. Let $S = \sum_{a \in A} s(a)$, and $P = \sum_{a \in A} v(a)$. Given any $r \in (0, 1)$, we let $M \in \mathbb{N}$ such that $\frac{1}{M} \leq r < \frac{1}{M-1}$.

We construct a bipartite graph $G = (I, J, E)$, such that $I$ consists of a single node $i$, and $J = J_A \cup J_B$ where $J_A = \{j_a | a \in A\}$, and $J_B = \{b_1, \ldots, b_M\}$. We let $E = I \times J$. For every $j_a \in J_A$ we let $d(j_a) = s(a)$, and $p(j_a) = v(a)$. For every $b_i \in J_B$ we let $d(b_i) = 2S$, and $p(b_i) = 2P$. We set $c(i) = 2MS + B$.

First note that the above instance to AoNDM is $r$-restricted. To see this note that for every $j \in J_A$, $d(j) \leq S < 2S + \frac{B}{M}$, and for every $j \in J_B$, $d(j) = 2S < 2S + \frac{B}{M}$. Since $2S + \frac{B}{M} = \frac{c(i)}{M} \leq r \cdot c(i)$ we have for every $(i, j) \in E$, $d(j) \leq r \cdot c(i)$.

In addition, note that for every optimal solution $X$ to the above $r$-AoNDM instance, $J_B \subseteq X$. This follows from the fact that $J_B$ is a feasible solution, and for every $j \in J_B$, the profit obtained by covering $j$ is strictly greater than the profit obtained from covering all of $J_A$. It therefore follows that the cover plan for $X$ uses exactly $2MS$ units of $i$’s capacity, leaving a capacity of $B$ to cover clients in $J_A$. Hence, the subset $X \cap J_A$ induces an optimal solution to the original knapsack problem. This completes the proof that $r$-AoNDM is NP-hard for any $r < 1$, even if there is only one base station. \qed

In what follows we present two approximation algorithms for the $r$-AoNDM problem. The algorithms are local-ratio algorithms that are based on a decomposition of the profit obtainable from every client into two non-negative terms; One part is proportional to the demand of the client, while the other part is the remaining profit. We define a family of feasible solutions, which we dub ”maximal” (see below for the formal definition), and prove that any such solution is an approximate solution when considering a profit function which is proportional to the demand. The algorithms we present generate such maximal solutions recursively. We then apply an inductive argument which proves that the solution generated by the algorithm is also an approximate solution w.r.t. the original profit function.

We first present an approximation algorithm that guarantees a solution whose value is within a factor of $\frac{1}{1-r}$ from the value of an optimal solution. This algorithm follows the cover-by-one paradigm, and thus every mobile station is covered by at most one base station.

Our second algorithm is obtained by a careful refinement of this algorithm, and an appropriate change to the notion of maximality. This algorithm uses the cover-by-many paradigm, and is guaranteed to produce a solution whose value is within a factor of $(1 - r)$ from the value of
an optimal solution, while the complexity increases by a polynomial factor. Next we specify several definitions needed for the analysis of the proposed algorithms.

Given any instance of \( r\)-AoNDM over a graph \( G = (I, J, E) \), and any two subsets \( A \subseteq I \) and \( B \subseteq J \), we define the \( A-B \) flow-graph of \( G \), \( G_f(A, B) = (V, F) \), such that \( V = \{s\} \cup A \cup B \cup \{t\} \) for new vertices \( s, t \notin I \cup J \), and \( F = (\{s\} \times A) \cup E(A, B) \cup (B \times \{t\}) \). We define a capacity function \( \gamma : F \rightarrow \mathbb{R}^+ \) as follows:

\[
\gamma(u, v) = \begin{cases} 
  c(v) & \text{if } u = s, v \in A \\
  \infty & \text{if } u \in A, v \in B \\
  d(u) & \text{if } u \in B, v = t.
\end{cases}
\]

For brevity of notation, we let \( G_f = G_f(I, J) \). Given any two subsets \( C, D \subseteq V \), we let \( \gamma(C, D) = \sum_{u,v \in F \cap (C \times D)} \gamma(u,v) \).

A cover plan \( x \) for \( S \subseteq J \) is said to be a cover-by-one plan if for every \( j \in S \), there is exactly one \( i \in I \) such that \( x(i, j) > 0 \). Given a cover-by-one plan \( x \) for \( S \subseteq J \), a cover-by-one plan \( x' \) for \( T \subseteq J \) is said to be a \( T \)-extension of \( x \), if for any \( j \in S \) and every \( i \in I \), \( x'(i, j) = x(i, j) \). Note that in such a case one is guaranteed to have \( S \subseteq T \). Given a cover plan \( x \) for \( S \subseteq J \), a cover plan \( x' \) for \( T \subseteq J \) is said to be a \( T \)-rearrangement of \( x \), if \( S \subseteq T \).

Given any cover-by-one plan \( x \) for \( S \subseteq J \), we say that \( x \) is cover-by-one-maximal (CBO-maximal) if for any \( j \in J \setminus S \), no \( S \cup \{j\} \)-extension of \( x \) exists. We further say \( S \subseteq J \) is CBO-maximal when it has a CBO-maximal cover plan which is clear from the context. For any \( A \subseteq I \) and \( B \subseteq J \), and any flow \( y \) in \( G_f(A, B) \), we can denote the value of the flow by \( y(s) \). Given any cover plan \( x \) for \( S \subseteq J \), we say that \( x \) is rearrangement-maximal if for any \( j \in J \setminus S \), no \( S \cup \{j\} \)-rearrangement of \( x \) exists. Given any set \( S \subseteq J \), let \( \overline{S} = J \setminus S \) and \( Y_S = I \setminus N(S) \). We say a cover plan \( x \) for \( S \subseteq J \) is cover-by-many-maximal (CBM-maximal) if \( x \) is rearrangement-maximal, and \( x(Y_S, S) \) is a maximum flow in the flow graph \( G_f(Y_S, S) \). As before, we further say \( S \subseteq J \) is CBM-maximal when it has a CBM-maximal cover plan which is clear from the context.

The following lemma, appearing in [16], serves as a basic tool with which we analyze the approximation guarantee of the algorithms proposed in this section.

**Lemma 4.4.3 (Local Ratio).** Let \( I \) be an instance to \( r\)-AoNDM, over a graph \( G = (I, J, E) \), with profit function \( p \). Then, if \( p = p_1 + p_2 \), and \( x \) is a cover plan for some set \( S \subseteq J \) which is \( c \)-approximate w.r.t. \( p_1 \), and also \( c \)-approximate w.r.t. \( p_2 \), then \( x \) is \( c \)-approximate w.r.t. \( p \).

### 4.4.1 A cover-by-one \( \frac{1-\epsilon}{2-\epsilon} \)-approximation algorithm

We start with Algorithm CBO-MC; roughly speaking, under CBO-MC, given a specific ordering of the clients, and given an existing cover plan \( x \), a client is added greedily by finding a CBO-extension of \( x \), if such an extension exists. Otherwise, the client is discarded. See Algorithm 4.1 for the pseudocode of the algorithm.

**Lemma 4.4.4.** Consider any instance of the \( r\)-AoNDM problem such that for every client \( j \), \( p(j) = \epsilon \cdot d(j) \), for some constant \( \epsilon \). Any cover-by-one plan \( x \) for \( S \subseteq J \) which is CBO-maximal is a \( \frac{1-\epsilon}{2-\epsilon} \)-approximate solution w.r.t. profit function \( p \).
Algorithm 4.1 CBO-MC \((G = (I, J, E), \text{demands } d, \text{profits } p, \text{capacities } c)\)

1: if \(J = \emptyset\) then
2: \hspace{1em} return \(x \equiv 0\)
3: \hspace{1em} end if
4: if there exists a \(j \in J\) such that \(p(j) = 0\) then
5: \hspace{1em} \(x \leftarrow \text{CBO-MC } (G' = (I, J \setminus \{j\}, E \setminus E(j)), d, p, c)\)
6: \hspace{1em} return \(x\)
7: else
8: \hspace{2em} for every \(j \in J\), set \(\varepsilon_j = \frac{p(j)}{d(j)}\)
9: \hspace{2em} set \(\varepsilon = \min_j \varepsilon_j\)
10: \hspace{2em} for every \(j \in J\), set \(p_1(j) = \varepsilon \cdot d(j)\)
11: \hspace{2em} set \(p_2 = p - p_1\)
12: \hspace{2em} \(x \leftarrow \text{CBO-MC } (G, d, p_2, c)\)
13: \hspace{2em} for every \(j\) such that \(p_2(j) = 0\) do
14: \hspace{3em} if \(\exists i \in N(j)\) such that \(c(i) - x(i) \geq d(j)\) then
15: \hspace{4em} set \(x(i, j) = d(j)\)
16: \hspace{3em} else
17: \hspace{4em} discard \(j\)
18: \hspace{3em} end if
19: \hspace{2em} end for
20: \hspace{1em} return \(x\)
21: \hspace{1em} end if

Proof. Let \(\overline{S} = J \setminus S\). Without loss of generality, we can assume that no uncovered client receives any service, i.e., for every \(j \in \overline{S}\), \(x(j) = 0\).

If \(S = J\), then \(x\) is an optimal cover plan, and therefore clearly a \(\frac{1-r}{1-r^2}\) approximate solution. Assume therefore that \(S \subsetneq J\). First note that for every \(i \in N(S)\), one of the following holds:

- Either there are no edges between \(i\) and \(\overline{S}\), or
- \(x(i) = x(i, S) > (1-r)c(i)\).

To see this, assume by contradiction that there exists an \(i \in N(S)\) such that there are edges between \(i\) and \(\overline{S}\), and \(x(i) \leq (1-r)c(i)\). By the assumption, there exists at least one client \(j \in \overline{S}\) such that \((i, j) \in E\). Consider the function \(x' : E \rightarrow \mathbb{R}^+\) defined by

\[
x'(i', j') = \begin{cases} 
  d(j') & \text{if } i' = i, j' = j \\
  x(i', j') & \text{otherwise.}
\end{cases}
\]

Clearly, for every \(i' \neq i\), \(x'\) does not violate the capacity constraint imposed by \(c(i')\), since by the feasibility of \(x\), for every such \(i'\), \(x'(i') = x(i) \leq c(i)\). Furthermore, since \(x\) was a cover-by-one plan, then so is \(x'\). Consider base station \(i\). Since by the assumption \(x(i) \leq (1-r)c(i)\), using the fact that the instance is \(r\)-restricted, we have \(x'(i) = x(i) + d(j') \leq c(i)\), hence the capacity constraint is satisfied for \(i\) as well. Finally, note that all clients \(j' \in S \cup \{j\}\) are
Algorithm

We prove by induction on the recursion that the cover plan returned from every call and profit function \( p \) function \( p \) and therefore clearly a \( \text{optimal profit w.r.t the graph } \mathbf{B} \) hypothesis, \( \text{recursive calls decreases by at least 1, thus the recursion will terminate.} \)

It follows that \( \text{for every client } i \in N(\overline{\mathbf{S}}), \ x(i) > (1 - r) c(i). \)

Let \( \text{OPT} \subseteq J \) denote any optimal solution to the problem. Note that

\[
p(\text{OPT}) = p(\text{OPT} \cap \mathbf{S}) + p(\text{OPT} \cap \overline{\mathbf{S}}) \leq p(\mathbf{S}) + \varepsilon \cdot \sum_{j \in \text{OPT} \cap \overline{\mathbf{S}}} d(j) \leq p(\mathbf{S}) + \varepsilon \cdot c(N(\overline{\mathbf{S}}))
\]

where the last inequality follows from the feasibility of OPT.

On the other hand, by the maximality of \( \mathbf{S} \), we are guaranteed to have

\[
d(S) = \sum_{j \in \mathbf{S}} d(j) = \sum_{i \in I} x(i) \geq \sum_{i \in N(\mathbf{S})} x(i) > \sum_{i \in N(\overline{\mathbf{S}})} (1 - r) \cdot c(i) = (1 - r) \cdot c(N(\overline{\mathbf{S}})),
\]

which in turn implies

\[
p(S) = \varepsilon \cdot d(S) > \varepsilon (1 - r) \cdot c(N(\overline{\mathbf{S}})).
\]

It follows that

\[
p(\text{OPT}) \leq p(S) + \frac{p(S)}{1 - r} = p(S) \left(1 + \frac{1}{1 - r}\right) = \frac{2 - r}{1 - r} \cdot p(S),
\]

hence \( \mathbf{S} \) is a \( \frac{1}{2 - r} \) approximate solution w.r.t the profit function \( p \).

\( \square \)

**Theorem 4.4.5.** Algorithm CBO-MC produces a \( \frac{1}{2 - r} \) approximate solution.

**Proof.** We prove by induction on the recursion that the cover plan returned from every call is a \( \frac{1}{2 - r} \)-approximate solution. Note that the number of clients in every two consecutive recursive calls decreases by at least 1, thus the recursion will terminate.

For the base case, since \( J = \emptyset \), there are no clients to cover, hence \( x \equiv 0 \) is an optimal cover, and therefore clearly a \( \frac{1}{2 - r} \)-approximate solution. For the inductive step, we have two cases to consider. First, consider the cover plan \( x' \) for \( B \subseteq J \setminus \{ j \} \) returned in line 6. By the induction hypothesis, \( B \) is a \( \frac{1}{2 - r} \) approximate solution w.r.t. the graph \( G' = (I, j, E) \) and profit function \( p \). Since \( p(j) = 0 \), the optimal profit w.r.t the graph \( G = (I, J, E) \) and profit function \( p \) cannot be greater than the optimal profit w.r.t the graph \( G' \) and profit function \( p \). Hence, \( B \) is also a \( \frac{1}{2 - r} \) approximate solution w.r.t. the graph \( G = (I, J, E) \) and profit function \( p \). The second case to consider is the cover plan \( x' \) for \( B \) returned in line 20. By the induction hypothesis, \( B \) is a \( \frac{1}{2 - r} \) approximate solution w.r.t. the graph \( G = (I, J, E) \) and profit function \( p_2 \). Since for every client \( j \) considered in lines 13–19, \( p_2(j) = 0 \), the optimal profit w.r.t the graph \( G = (I, J, E) \) and profit function \( p_2 \) cannot be greater than the optimal profit attainable from the instance returned from the recursive call. Hence, the solution returned in line 20 is a \( \frac{1}{2 - r} \) approximate solution w.r.t. the graph \( G = (I, J, E) \) and profit function \( p_2 \), and so is any extension of it using clients \( j \) such that \( p_2(j) = 0 \). Note that for every client \( j \) such that \( p_2(j) = 0 \), who has a neighbor with sufficient residual capacity, \( j \)
is added to the cover, where exactly one base station is used to satisfy its demand. It follows that the solution returned in line 20 is a CBO-maximal solution. By Lemma 4.4.4 it follows that this solution is a $\frac{1-r}{2-r}$ approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p_1$. Using Lemma 4.4.3 we conclude that the solution returned is a $\frac{1-r}{2-r}$ approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p = p_1 + p_2$, which completes the proof.

Note that the solution $x$ produced by algorithm CBO-MC is a cover-by-one plan. It therefore follows that the ratio between the optimal cover-by-one solution and the optimal cover-by-many solution is at most $\frac{1-r}{2-r}$ as well.

### 4.4.2 A cover-by-many $(1 - r)$-approximation algorithm

We now turn to describe our second algorithm, called CBM-MC, which achieves an approximation ratio of $(1-r)$ using the cover-by-many paradigm. Under CBM-MC, a client is added by first trying to exhaust the capacities of base stations which cannot contribute to uncovered clients, and then using the capacity of the remaining base stations in order to complete the cover. If such a cover cannot be produced, then the client is discarded. The pseudocode of the algorithm is given in Algorithm 4.2, where we use the subroutine $\text{EK-MAXFLOW}(G_f(A, B))$ to denote the computation of the maximum s-t flow in the flow graph $G_f(A, B)$ using the Edmonds-Karp algorithm [25]. Our choice of the Edmonds-Karp algorithm is motivated by two of its properties, namely, the fact that it converges from any feasible flow, and the fact that it uses augmentation paths. This choice can be substituted by any algorithm for computing maximum flow, which satisfies these properties. Note that by duality, given any s-t flow in a flow graph $G_f(A, B)$, it is easy to verify if a cut is a minimum cut by checking that all the edges are saturated.

Given a cover plan $x$ for $S \subseteq J$, let $\overline{S} = J \setminus S$, and consider $I$ as partitioned into two sets: $N_{\overline{S}} = N(S)$, and $Y_S = I \setminus N_{\overline{S}}$. Note that by definition, for every $j \in \overline{S}$ and $i \in Y_S$, $(i, j) \notin E$. The following lemma, whose proof is omitted due to space constraints, provides a necessary and sufficient condition for covering a set of clients.

**Lemma 4.4.6.** For any instance of r-AoNDM over a graph $G = (I, J, E)$, and any $A \subseteq I$ and $B \subseteq J$, $\{t\}$ is a minimum s-t cut in the flow-graph $G_f(A, B)$ if and only if $A$ can cover all clients in $B$.

**Proof.** Let $x$ be a maximum s-t flow in $G_f(A, B)$. By duality, the value of the maximum s-t flow is the same as the capacity of the minimum s-t cut, and the edges of any such cut are all saturated by any maximum flow. Assume $\{t\}$ is a minimum s-t cut in $G_f(A, B)$, and assume by contradiction that there is client $j \in B$ which is not fully covered. It follows that $\sum_{(i,j) \in E} x(i,j) < d(j)$. By flow conservation, we have $x(j, t) = \sum_{(i,j) \in E} x(i,j)$, which implies $x(j, t) < d(j)$. Since $(j, t)$ is an edge in the minimum s-t cut $\{t\}$, and its capacity is $d(j)$, this contradicts the fact that all such edges are saturated by the maximum flow $x$. Assume now $\{t\}$ is not a minimum s-t cut, and assume by contradiction that all clients can be covered. By flow conservation it follows that for every $j \in J$, $\sum_{(i,j) \in E} x(i,j) = d(j) =$
Algorithm 4.2 CBM-MC ($G = (I, J, E)$, demands $d$, profits $p$, capacities $c$)

1: $x \leftarrow$ EK-MaxFlow ($G_f$)
2: if $\{t\}$ is a MinCut in $G_f$ then
3: return $x$
4: end if
5: if there exists a $j \in J$ such that $p(j) = 0$ then
6: $x \leftarrow$ CBM-MC ($G' = (I, J \setminus \{j\}, E \setminus E(j)), d, p, c$)
7: return $x$
8: else
9: for every $j \in J$, set $\varepsilon_j = \frac{p(j)}{d(j)}$
10: set $\varepsilon = \min_j \varepsilon_j$
11: for every $j \in J$, set $p_1(j) = \varepsilon \cdot d(j)$
12: set $p_2 = p - p_1$
13: $x \leftarrow$ CBM-MC ($G, d, p_2, c$)
14: for every $j$ such that $p_2(j) = 0$ do
15: $S \leftarrow \{j' \in J \mid x(j') = d(j')\}$
16: set $N_{S \setminus \{j\}} = N(J \setminus (S \cup \{j\}))$
17: set $Y_{S \cup \{j\}} = I \setminus N_{S \setminus \{j\}}$
18: $y \leftarrow$ EK-MaxFlow($G_f(Y_{S \cup \{j\}}, S \cup \{j\})$)
19: $z \leftarrow$ EK-MaxFlow($G_f(I, S \cup \{j\})$), starting from the initial feasible flow $y$.
20: if $\{t\}$ is a MinCut in $G_f(I, S \cup \{j\})$ then
21: $x \leftarrow z$
22: end if
23: end for
24: return $x$
25: end if

$x(j, t)$, which by summing over all $j \in J$ implies that the value of the flow equals the capacity of the cut $\{t\}$. By duality, this implies that $\{t\}$ is a minimum $s$-$t$ cut, contradicting our assumption.

Lemma 4.4.6 admits a method for finding a rearrangement-maximal cover plan, as shown in the following lemma:

**Lemma 4.4.7.** Given any instance of $r$-AoNDM over a graph $G = (I, J, E)$, any cover plan $x$ for $S \subseteq J$, and a client $j \in J \setminus S$, the task of finding an $S \cup \{j\}$-rearrangement of $x$, if one exists, can be done in polynomial time.

**Proof.** In order to find a rearrangement of $x$, consider the flow graph $G' = G_f(I, S \cup \{j\})$, and let $y$ be a maximum flow in this graph. If $\{t\}$ is a minimum cut in $G'$, then by Lemma 4.4.6, $y$ is a cover plan for $S \cup \{j\}$, and hence it is an $S \cup \{j\}$-rearrangement. If $\{t\}$ is not a minimum cut in $G'$, then its capacity is strictly greater than the maximum flow, and therefore not all clients in $S \cup \{j\}$ can be satisfied. Verifying whether or not $\gamma(t) = y(t)$ can clearly be done in polynomial time. 

Technion - Computer Science Department - Ph.D. Thesis PHD-2008-02 - 2008
Corollary 4.4.8. Given any instance to r-AoNDM over a graph $G = (I, J, E)$, any cover plan $x$ for $S \subseteq J$, and a client $j \in J \setminus S$, the task of finding a rearrangement of $x$ which is rearrangement-maximal can be done in polynomial time.

Proof. By iteratively applying Lemma 4.4.7, we are guaranteed to obtain a rearrangement-maximal cover plan.

The following lemmas describe the correlation between the maximum flow in $G_f$, and the maximum flow in flow graphs of the form $G_f(Y_S, S)$, for sets $S$ which have a cover plan.

Lemma 4.4.9. Assume $S \subseteq J$ has some cover plan. Then, there exists a maximum flow $x$ in $G_f$ such that $x(Y_S, S) = \text{MAXFLOW}(G_f(Y_S, S))$. Furthermore, such a flow can be found in polynomial time.

Proof. Let $y = \text{MAXFLOW}(G_f(Y_S, S))$. Clearly $y$ is a feasible flow in $G_f$ as well. Consider the Edmonds-Karp Algorithm (EK-MAXFLOW, see [25] for details) for finding a maximum flow, executed on graph $G_f$, starting from the initial feasible flow $y$. We show that for every augmentation path found by EK-MAXFLOW, after increasing the flow along this path and obtaining some flow $y'$, $y'(s, Y_S) \geq y(s, Y_S)$.

First note that we can assume that all the augmentation paths used by the EK-MAXFLOW algorithm are simple paths. Furthermore, note that by the fact that any augmentation path is simple, we obtain that for every flow $y'$ obtained during executing the EK-MAXFLOW algorithm, and for every $i \in Y_S$, $y(s, i) \leq y'(s, i)$, since such flow can only decrease if the algorithm uses a path $p$ such that $(i, s) \in p$, which implies that $p$ is not a simple path.

Since for every feasible flow $z$ we have $z(Y_S, S) = z(s, Y_S)$ (by flow conservation, and using the fact that there are no edges between $Y_S$ and $S$), we can conclude that during the entire execution of the EK-MAXFLOW algorithm, the flow $y'$ resulting in augmenting any path $p$ satisfies $y'(s, Y_S) \geq y(s, Y_S)$. On the other hand, note that given any maximum flow in $G_f$, if we consider its flow path-decomposition, then the set of paths using edges between $Y_S$ and $S$ also constitute a flow in $H_S$ (due to the unidirectionality of edges between $N_S$ and $S$ in $G_f$). Hence these paths cannot support a flow whose value is greater than $\text{MAXFLOW}(G_f(Y_S, S))$.

Finally note that EK-MAXFLOW produces a maximum flow in $G_f$ in polynomial time, which completes the proof of the lemma.

The above lemma gives rise to the following corollary:

Corollary 4.4.10. If there exists a rearrangement-maximal cover plan $y$ for $S \subseteq J$, then there exists a CBM-maximal cover plan $x$ for $S$. Furthermore, such a cover plan can be found in polynomial time.

Proof. Using a similar argument as the one used in Lemma 4.4.9, by running EK-MAXFLOW with an initial feasible flow $z = \text{MAXFLOW}(G_f(Y_S, S))$, we are guaranteed to produce a cover plan for $S$ (by the existence of $y$, $S$ can be covered by $I$). Furthermore, this cover must also be CBM-maximal. Note that such a cover plan can be found in polynomial time by the same arguments as the ones used in Lemma 4.4.9.
The following lemma shows a bound on the value of any maximum flow in $G_f$.

**Lemma 4.4.11.** Given any $S \subseteq J$, if $S$ has a CBM-maximal cover plan, then $\text{MaxFlow}(G_f) \leq \text{MaxFlow}(G_f(Y_S, S)) + c(N_\overline{S})$.

**Proof.** Let $y$ be a CBM-maximal cover plan for $S$, and consider a partition of $y$ into two types of flow paths, each consisting of 3 edges:

- $T_1 = \{p = (s, i, j, t) \mid i \in Y_S\}$.
- $T_2 = \{p = (s, i, j, t) \mid i \in N_\overline{S}\}$.

Note that such a packing exists, by the directionality of the edges in $G_f$.

If we denote the flow along a flow path $p$ by $x(p)$, then clearly $\sum_{p \in T_1} x(p) \leq \text{MaxFlow}(G_f(Y_S, S))$ since all paths in $T_1$ are paths in $G_f(Y_S, S)$, and therefore cannot support a flow greater than $\text{MaxFlow}(G_f(Y_S, S))$. On the other hand, $\sum_{p \in T_2} x(p) \leq c(s, N_\overline{S}) = c(N_\overline{S})$ since all these paths use edges in the cut $(s, N_\overline{S})$. It therefore follows that $\text{MaxFlow}(G_f) \leq \text{MaxFlow}(G_f(Y_S, S)) + c(N_\overline{S})$.

We can now continue in the same way as we did with the simpler algorithm, where CBM-maximality replaces CBO-maximality.

**Lemma 4.4.12.** Consider any instance of the r-AoNDM problem such that for every client $j$, $p(j) = \varepsilon \cdot d(j)$, for some constant $\varepsilon$. Any cover plan $x$ for $S \subseteq J$ which is CBM-maximal is a $(1 - r)$-approximate solution w.r.t. profit function $p$.

**Proof.** Let $x$ be any cover plan for $S \subseteq J$ which is CBM-maximal. If $S = J$, then $x$ is an optimal cover plan, and therefore clearly a $(1 - r)$ approximate solution. Assume $S \subsetneq J$. Note that by maximality of $x$, $x(Y_S, S) = \text{MaxFlow}(G_f(Y_S, S))$, and since $S \subsetneq J$, $x(N_\overline{S}, S) > (1 - r)c(N_\overline{S})$, i.e., $c(N_\overline{S}) < \frac{x(N_\overline{S}, S)}{1 - r}$. By the fact that $x$ is a cover plan for $S$, we have $p(S) = \varepsilon d(S) = \varepsilon (x(N_\overline{S}, S) + x(Y_S, S))$, since $N_\overline{S}, Y_S$ are a partition of $I$.

Let OPT $\subseteq J$ denote any optimal solution to the problem. We wish to bound the value of $p($OPT$)$. Clearly, for any maximum $s$-$t$ flow $y$ in $G_f$, $d($OPT$) \leq y(s)$, since any cover plan

---

1Note that these are not augmentation paths used in computing the maximum flow by EK-MaxFlow. These paths are part of an actual path decomposition of the maximum flow.
for OPT is a feasible flow in $G_f$. Combining the above with Lemma 4.4.11 we obtain that for any maximum s-t flow $y$ in $G_f$,

$$d(\text{OPT}) \leq y(s) \leq \text{MaxFlow}(G_f(Y_S, S)) + c(N_S)$$

$$< x(Y_S, S) + x(N_S, S)$$

$$= \frac{1}{1-r} \left( (1-r) \cdot x(Y_S, S) + x(N_S, S) \right)$$

$$\leq \frac{1}{1-r} \left( x(Y_S, S) + x(N_S, S) \right)$$

$$= \frac{1}{1-r} d(S).$$

By the definition of $p$ we can conclude that $p(S) > (1-r) \cdot p(\text{OPT})$, which completes the proof. \hfill \Box

**Theorem 4.4.13.** Algorithm CBM-MC produces a $(1-r)$-approximate solution.

**Proof.** We prove by induction on the recursion that the cover plan returned from every call is a $(1-r)$-approximate solution.

For the base case, if $\{t\}$ is minimum cut, then by Lemma 4.4.6 all the clients can be covered, hence $x$ is an optimal cover plan, and therefore clearly a $(1-r)$-approximate solution. For the inductive step, we have two cases to consider. First, consider the cover plan $x'$ for $B \subseteq J \setminus \{j\}$ returned in line 7. By the induction hypothesis, $B$ is a $(1-r)$-approximate solution w.r.t. the graph $G' = (I, J \setminus \{j\}, E \setminus E(j))$ and profit function $p$. Since $p(j) = 0$, the optimal profit w.r.t. the graph $G = (I, J, E)$ and profit function $p$ cannot be greater than the optimal profit w.r.t. the graph $G'$ and profit function $p$. Hence, $B$ is also a $(1-r)$-approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p$. The second case to consider is the cover plan $x'$ for $B$ returned in line 24. By the induction hypothesis, $B$ is a $(1-r)$-approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p_2$. Since for every client $j$ considered in lines 13–19, $p_2(j) = 0$, the optimal profit w.r.t. the graph $G = (I, J, E)$ and profit function $p_2$ cannot be greater than the optimal profit attainable from the instance returned from the recursive call. Hence, the solution returned in line 20 is a $(1-r)$-approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p_2$, and so is any superset of this solution produced by adding any of the clients for which $p_2(j) = 0$.

Note that by Lemma 4.4.7, for every client $j \notin S$ such that $p_2(j) = 0$, lines 19 and 20 compute an $S \cup \{j\}$-rearrangement of the current cover plan, if such a rearrangement exists. Hence the resulting solution returned in line 24 is a rearrangement-maximal solution. In addition, by lines 18–19, the cover plan $x'$ computed in every iteration also satisfies that $x(Y_S, S)$ is a maximum flow in the flow graph $G_f(Y_S, S)$. It therefore follows that the cover plan returned in line 24 is also CBM-maximal. By Lemma 4.4.12 it follows that this solution is a $(1-r)$-approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p_1$. Using Lemma 4.4.3 we conclude that the solution returned is a $(1-r)$ approximate solution w.r.t. the graph $G = (I, J, E)$ and profit function $p = p_1 + p_2$, which completes the proof. \hfill \Box
4.5 Simulation results

In the previous sections we proposed two different algorithms for a new global mechanism for cell selection in future cellular networks. The main difference between these two algorithms is the way the demand of a mobile client is satisfied. In the CBO-MC Algorithm (Section 4.4.1) at most one base station satisfies the demand of any given mobile station while the CBM-MC Algorithm (Section 4.4.2) allows satisfaction of the demand simultaneously by more than one base station.

In order to study the expected performance of the proposed global cell selection algorithms with respect to the current local mobile SNR-based protocol we conducted several simulations over high-loaded, capacity constrained, 4G-like networks. A secondary goal of these simulations was to study the “benefit” of using the new ability, defined by the IEEE 802.16e, of a mobile station to be satisfied simultaneously by more than one base station.

4.5.1 Methodology

We built a network consisting of an \( n \times n \)-grid of clients’ locations (demand points that we considered as a single client, or bins). Each client has a service request for either voice or data service. The demand of a voice and data client is defined as 1 and 25, respectively\(^2\). Under this ratio between the demand of data and voice clients, the number of the data clients was taken in such a form so that the overall voice volume is 20\% of the network’s traffic\(^3\). The locations for each type of client was uniformly and randomly selected over the grid. The profit for satisfying the demand of a voice client was defined as 1, while satisfaction of a data client is credited with a profit that is proportional to its demand (i.e., 25 units of profit).

We maintain microcells and picocells in our network. Since we implemented the restricted version of AoNDM, the demand of every client must be less than or equal to an \( r \)-fraction of the capacity of any base station service this client. Therefore, the capacity of a picocell was taken to be about \( 25/r \), for any given value of \( 0 < r < 1 \). To simulate high-loaded networks we assumed that the total sum of (client) demands equals the sum of (base station) capacities in the network. The ratio between the number of picocells and microcells was defined to be \( \lambda \) while this factor was also selected as the ratio between the corresponding radiuses and capacities of microcells and picocells. By taking \( \lambda = 5 \), we can now derive the appropriate number of microcells and picocells. The locations for each type of base station was uniformly and randomly selected over the grid and clients were associated with (omnidirectional) base stations according their distance from each of the centers.

In each of the following three sets of simulations we measured the ratio between the total profit achieved by each of the three algorithms and the total profit of all connected clients. As AoNDM is NP-hard, the maximum possible profit is hard to calculate, and we consider the total profit of all connected clients as an upper bound on the optimal solution.

\(^2\)The bit rate for voice applications is 64Kbps and the downlink rate for data application is approximately 2Mbps in HSDPA. This gives a ratio of 25-30 between the demand of voice and data clients.

\(^3\)To be precise, if \( n_v \) and \( n_d \) are the number of voice and data clients, respectively, and \( d_v \) and \( d_d \) are the corresponding demands, then the following are satisfies for an overall voice volume of \( \gamma \) of the network’s traffic:

\[
\frac{d_v \cdot n_v}{d_v \cdot n_v + d_d \cdot n_d} = \gamma, \quad n_d = n^2 - n_v, \quad n_v = \left\lfloor \frac{\gamma \cdot d_d \cdot n_v^2}{(d_d - d_v) \gamma + 1} \right\rfloor.
\]

In our case \( \gamma = 0.2 \).
4.5.2 Results

In the first set of simulations we study the performance of the three algorithms over a variable size of network (10K to 40K) and different values of $r$ (0.05 to 0.3). Typical results are shown in figures 4.2(a)-4.2(c), where the upper, middle and the lower curves corresponds to the cover-by-many algorithm, cover-by-one algorithm, and the greedy-best detected-SNR algorithm, respectively. In each of the three scenarios, our results show that the cover-by-many algorithm is better than the cover-by-one algorithm by 5% (for $r = 0.05$) to 11% ($r = 0.3$). An improvement of at least 10% (and up to 20%) was achieved by the cover-by-many algorithm in comparison with the greedy-best detected-SNR algorithm. The results show that the performances of all three algorithm are nearly independent of the size of the network. Moreover, due to the existence of the simultaneous coverage in the third algorithm, when $r$ increases the “distance” between the performance of the cover-by-many algorithm and the other two algorithms also increases in a significant fashion. This shows that when there exist mobile clients with demands that are relatively close to the capacity of the servicing cell.
The value of \( r \)
Percentage of the total earned profit
Greedy SNR
Cover-by-One
Cover-by-Many

(a) Expected profit as a function of \( r \) (\( n = 15129 \))

(b) Expected profit as a function of available capacity (\( r = 0.25, n = 15129 \))

Figure 4.3: Results for the second and the third set of simulations

(e.g., in case of picocells) allowing satisfaction of a client by more than one base station is crucial in order to maintain high utilization of the network capacities.

The second set of simulations investigates the level of profit achieved by the three algorithms when the value of \( r \) varies (from \( r = 0.01 \) to \( r = 0.5 \)). We fixed a network of 15129 clients (i.e., a grid of \( 123 \times 123 \)) with a number of picocells and microcells as explained above. Focusing on the relative fraction of the demand of a client with respect to the capacity of any serviced base station, the results show (Figure 4.3(a)) that when this fraction increases the ability to reach a higher percentage of the total possible profit decreases. As shown in Figure 4.3(a), all three algorithms exhibit the same behavior. The performance of the cover-by-many algorithm (upper curve) decreases from 100% to 89% when \( r \) increases from 0.01 to 0.5. The cover-by-one algorithm decreases by 21% (from 100% in \( r = 0.01 \) to 79.5% in \( r = 0.5 \)), and the greedy-best detected-SNR algorithm (lower curve) exhibited a decrease of 30% (from 89% to 59%).

The third set of simulations examines the level of profit obtained by the three algorithms when the available capacity increases. We fixed a network of 15129 clients, where each client has a demand (of any service) that is of at most a fraction of \( 1/4 \) (\( r = 0.25 \)) the capacity of each of the serviced base stations. In this study, the number of picocells as well as microcells was increased by \( j \) times their basic number, \( j = 1, 1.5, 2, \ldots, 5 \), where the basic numbers are the same as the ones computed in the first set of simulations (65 microcells and 327 picocells). Note that for \( j > 1 \), the total capacity is higher than the total demand of clients. As one might expect (see Figure 4.3(b)), when there is a larger number of base stations the performance of the three algorithms can only improve. The greedy-best detected-SNR algorithm (lower curve) achieve an improvement of up to 8% (from 79% to 87%) when the number of base station grows from 392 to 1960. The cover-by-one algorithm (in the middle) achieves an improvement of up to 8% (from 89% to 97%), and the cover-by-many algorithm (upper curve) is nearly constant (around 99%) in its ability to satisfy clients.
Finally, the worst-case running time of each of the algorithms, for all cases, was approximately 4 minutes for the case of \( n = 40000, r = 0.25 \), on a Pentium M machine, 1.4 GHz, and 256 Mb of RAM.

4.6 Conclusions and open problems

In this chapter we present a rigorous study of a new approach for cell selection in fourth generation cellular networks. Unlike the current cell selection protocol, our proposed mechanism is global, has a performance guarantee, and addresses many of the anticipated future technologies. We show that even though AoNDM is hard to approximate to within a reasonable factor, we can still cover all practical scenarios by adopting the assumption that every mobile station has a traffic demand that is relatively smaller than the capacity of any base station that is able to participate in its coverage. We give two approximation algorithms for this problem. The first is a \( \frac{1-r}{2} \)-approximation algorithm for the case where each mobile station can be covered by exactly one base station (cover-by-one). The second is a slower, delicate refinement of the first algorithm, guaranteeing a \((1-r)\)-approximate solution, that adopt the new IEEE 802.16e possibility of simultaneous coverage of mobile clients by more than one base station (cover-by-many). We compare between global mechanisms that are based on our approximation algorithms and a local procedure performed by the current best-SNR greedy cell selection protocol. We show that when clients of very high bandwidth demand, relatively to the base station’s capacity, exist, the use of multiple base station to satisfy the demand of a mobile station can maintain a level of at least 97% of the possible coverage - 20% better coverage than the current best-SNR greedy cell selection method. In addition to future networks, such relevant scenarios may be found in spread areas where there are several very small populated areas and ‘standard’ infrastructure is not cost-effective. In these areas, coverage can be achieved using several WiMAX-cells and situations where such cells are over-loaded may be common. Our scheme for cell selection can be used in order to allow a better utilization of these coverage solutions.

There are several interesting problems that arise from this work. The first is whether or not one can devise a constant-factor approximation algorithm to the \( r \)-AoNDM that is independent of \( r \). In this respect it seems that a primary starting point for answering such a question is determining the complexity of \( r \)-AoNDM in the case where \( r = 1 \). A more general question is whether or not there exists a PTAS for the problem, and under which conditions. We note that by our reduction, the general case is proven to be hard to approximate for instances in which the demand of every client is strictly greater than the capacity of the base stations which can contribute to its coverage. Abusing our notation, it is unclear whether such a phase transition occurs in \( r = 1 \), or is there some \( r > 1 \) for which the \( r \)-AoNDM problem still adheres to good approximation algorithms.
This page was intentionally left blank
Chapter 5

Discussion

In this thesis, we developed new techniques for solving some of the important and interesting problems related to the optimization of future cellular networks.

We studied, in Chapter 2, two important cell planning problems: the budgeted cell planning (BCPP) and the minimum-cost cell planning problems (CPP). As far as we know, previous work do not provide performance guarantee for these problems. Our formulation is very flexible and it allows the incorporation of several future cellular networks characterizations (such as smart antennas, capacities for base stations and for services, interference, and different-size-cells). Due to the NP-hardness of the problems, we resorted to polynomial-time approximation algorithms. We show that although BCPP is NP-hard to approximate, we can still cover all useful scenarios by adopting a very practical assumption, called the $k4k$-property, satisfied in practice by cellular networks, and we develop a fully combinatorial $\frac{1}{2} + \frac{1}{e}$-approximation algorithm for this problem. We obtained a combinatorial $O(\log W)$-approximation algorithm addressed a non-interference version of CPP, where $W$ is the largest capacity over all base stations selected for opening. Then we simulated our algorithm to the CPP and compared its performance to the results produced by previous works. Our results indicate that practical implementation derived from the theoretical scheme can provide solutions that are close to the optimal solution and much better than our proved theoretical bound.

Chapter 3 deals with algorithmic aspects for planning radio access networks for future cellular systems. Such a design requires a replacement of the commonly used star based architecture (in which an RNC is connected to a set of base stations via direct links) with a new tree-topology structure. In the new architecture a base station can be connected to an RNC via other base stations, resulting in a complex network design problem.

We studied five possible solutions for the metric version of the bounded-degree minimum routing cost spanning tree problem (BDRT). These methods involve both an approximation algorithm and a family of greedy heuristics. Our results indicate that a combination of a certain greedy heuristic and our $O(\log n)$-approximation algorithm generates a solution, which produces a close-to-optimal result in practical scenarios with a guaranteed worst case bound. Moreover, this solution can be efficiently computed for networks of large size.

We present, in Chapter 4, a rigorous study of a new approach for cell selection in future
cellular networks. Unlike the current cell selection protocol, our proposed mechanism is global, has a performance guarantee, and addresses many of the anticipated technologies. We show that even though AoNDM is hard to approximate to within a reasonable factor, we can still cover all practical scenarios by adopting the assumption that every mobile station has a traffic demand that is relatively smaller than the capacity of any base station that is able to participate in its coverage.

We give two approximation algorithms for this problem. The first is a $\frac{1-r}{2r}$-approximation algorithm for the case where each mobile station can be covered by exactly one base station (cover-by-one). The second is a slower, delicate refinement of the first algorithm, guaranteeing a $(1 - r)$-approximate solution, that adopt the new IEEE 802.16e–2005 possibility of simultaneous coverage of mobile clients by more than one base station (cover-by-many). We compare between global mechanisms that are based on our approximation algorithms and a local procedure performed by the current best-SNR greedy cell selection protocol, and show that when clients of very high bandwidth demand, relatively to the base station’s capacity, exist, the use of multiple base station to satisfy the demand of a mobile station can maintain a significantly higher level of utilization of the possible coverage.

The results presented in this thesis indicate that a theoretical approach based on approximation algorithms is useful for many of the future planned technologies (in particular better utilization of the base station’s capacity), and provides both theoretically bounded and practical good algorithms for various problems appears in the optimization of cellular networks. This is a significant step towards focused planning and optimization of future networks, and make approximation algorithms to be a major player in many planning and covering problems in cellular networks.
References


This page was intentionally left blank