Linköping Studies in Science and Technology
Licentiate Thesis No. 1473

Tracking Area Planning in Cellular Networks

- Optimization and Performance Evaluation

Sara Modarres Razavi

Department of Science and Technology
Linköping University, SE-601 74, Norrköping, Sweden

Norrköping 2011

Tracking Area Planning in Cellular Networks
- Optimization and Performance Evaluation

sarmo@itn.liu.se

ISBN 978-91-7393-214-1
ISSN 0280-7971
LiU-TEK-LIC-2011:12

Linköping University
Department of Science and Technology
SE-601 74 Norrköping
Tel: +46 11 36 30 48
Fax: +46 11 36 32 70

Printed by LiU-Tryck, Linköping, Sweden, 2011.

Abstract

The enormous competition in the telecommunications market results in the necessity of optimized and cost-efficient networks for the operators and service providers. Tracing users cost-efficiently is one of the major challenges in the study of location management of wireless cellular networks. Tracking Area (TA) is a logical grouping of cells in Long Term Evolution (LTE) networks. TA manages and represents the location of User Equipments (UEs). One of the well-known performance consideration is the signaling overhead of tracking area update versus that for paging. This thesis deals with planning and optimization of tracking area configuration in LTE networks.

TA design must be revised over time in order to adapt to changes and trends in UE location and mobility patterns. Re-optimization of the initial planning subject to different cost budgets is one of the problems considered in the thesis. By re-optimization, the design is successively improved by re-assigning some cells to TAs other than their original ones. To solve the resulting problem, an algorithm based on repeated local search is developed.

By extending the line of research, the trade-off between the performance in terms of overall signaling overhead of the network and the reconfiguration cost is considered. This trade-off is modeled as a bi-objective optimization problem to which the solutions are characterized by pareto-optimality. Solving the problem delivers a host of potential trade-offs among which the selection can be based on the preferences of a decision-maker. An integer programming model and a heuristic based on genetic algorithm are developed for solving the problem in large-scale networks.

In comparison to earlier generations of cellular networks, LTE systems allow for a more flexible configuration of TA design by means of Tracking Area List (TAL). How to utilize this flexibility in applying TAL to large-scale networks remains unexplored. In this thesis, three approaches for allocating and assigning TA lists have been presented, and their performance is compared with each other, as well as with the standard location management scheme.

Automatic reconfiguration is an important element in LTE. The network continuously collects UE statistics, and the management system

adapts the network configuration to changes in UE distribution and demand. In this thesis an evaluation of dynamic configuration of TA design, including the use of TAL, has been performed and compared to the static configuration by using a case study.

Acknowledgement

It would not have been possible to write this thesis without the help and support of the kind people around me. Although I am not able to name them all, I would like to dedicate the thesis to them.

First and foremost, I want to express my sincerest thanks to my supervisor, Prof. Di Yuan. His active academic personality and his outstanding energy in work and life are motivational to me. Without his inspiring ideas, continuous guidance and valuable feedbacks, I would have never been able to accomplish this work.

I also want to show my appreciation to the Ericsson Research group in Linköping, in particular to Dr. Fredrik Gunnarsson and Dr. Johan Moe for the valuable discussions and the informative meetings, which resulted in several publications.

I am grateful for the financial support I received from CENIIT, Linköping Institute of Technology, and Swedish Research Council (Vetenskapsrådet).

I would like to thank all my wonderful colleagues and friends at the Division of Communication and Transport Systems (KTS), for creating such an ideal place to work in, and for helping me to enjoy everyday of my PhD studies. I like to especially thank my roommate, Lei, for his friendship and guidance from the very first day.

My deepest gratitude goes to my family and friends: To my adorable fabulous parents, Reza and Farzaneh, and to my lovely sister, Sonia, for believing in me and supporting me in every way possible throughout my life, and to my so many incredible friends for their care and friendships.

Finally, my heartfelt thanks go to my carrying and loving partner in life, Mahziar, without his love, support, and encouragements all along, I am sure none of this was possible. ♡

Norrköping, March 2011
Sara Modarres Razavi

Abbreviations

3GPP	3rd Generation Partnership Project
CS	Circuit Switched
GA	Genetic Algorithm
GPRS	Global Packet Radio System
GSM	Global System for Mobile Communication
HO	HandOver
LA	Location Area
LAM	Location Area Management
LAU	Location Area Update
LP	Linear Programming
LS	Local Search
LTE	Long Term Evolution
MM	Mobility Management
MME	Mobility Management Entity
MS	Mobile Station
MSC	Mobile Switching Center
MT	Mobile Terminal
NP	Non-deterministic Polynomial time
PS	Packet Switched
PV	Preference Value
QoS	Quality of Service
RA	Routing Area
SMS	Short Message Service
SON	Self Organizing/Optimizing Network
SGSN	Serving GPRS Support Node
S-GW	Serving Gateway
STA	Standard Tracking Area
TA	Tracking Area
TAL	Tracking Area List
TAP	Tracking Area Planning
TAR	Tracking Area Re-optimization
TAU	Tracking Area Update
UE	User Equipment

UMTS	Universal Mobile Telecommunications System
URA	UTRAN Registration Area
UTRAN	Universal Terrestrial Radio Access Network
VLR	Visitor Location Register

Contents

List of Tables

List of Figures

Chapter 1

Introduction

There has been an extreme growth in the area of wireless and mobile communications in the past decades. Having an optimized and efficient network is one of the most important factors in the fierce competition among service providers. Long Term Evolution (LTE) is a recent standard in the mobile network technology. It is initiated to bring mobile broadband via new technology, new applications and new services to the wireless cellular network. This results in new architectures and configurations. Self-optimizing and self-organizing are the capabilities which the 3^{rd} Generation Partnership Project (3GPP) has standardized for LTE [7]. By automating the configuration and optimization of cellular networks, it is possible to lower the cost and the time consumed for the manual operation. It will also improve network performance and flexibility [4, 5].

Mobility management (MM) is one of the main functions in mobile networks. It aims to track the user equipments (UEs) and to allow calls, SMS and other mobile phone services to be delivered to UEs. For any mobility protocol there are two separate problems to be solved. One is location management (or sometimes called reachability), which keeps track of the positions of a UE in the mobile network. The other one is handover management (or sometimes called session continuity), which makes it possible for a UE to continue its sessions while moving to another cell and changing its access point. This thesis focuses on the location management problems.

Tracing UEs in a mobile network is the key task in location management. Tracking Area (TA) in LTE is a logical grouping of cells in a network. TA is almost the same concept as the Location Area (LA) in the circuit-switched (CS) domain and the Routing Area (RA) in the packet-switched (PS) domain in GSM and UMTS [1]. The main function of the TA is to manage and represent the locations of UEs.

1.1 Scope of the Thesis

The thesis aims to address some TA planning and optimization problems and concepts in LTE networks. In configuring TAs, a key consideration is to minimize the total amount of signaling overhead. The overall signaling overhead of a network consists of two terms: update overhead and paging overhead. In the standard scheme of TA update (TAU) and paging for tracking a UE, the Mobility Management Entity (MME) records the TA in which the UE is registered. When a UE moves to a new TA, there will be a TAU signaling overhead. The paging signaling overhead happens when the UE is being called. In order to place the call to the UE, MME broadcasts a paging message in all cells of the UE's registered TA.

Consider a TA design that is optimized for a network in the planning phase. As UE distribution and mobility patterns change over time, the optimized TA configuration will no longer perform satisfactorily. Therefore a TA reconfiguration may be required for reducing the signaling overhead. The present thesis suggests a re-optimization approach for revising a given TA design. The approach is justified by the fact that once a TA design is in use, it is not feasible to deploy a green-field design that significantly differs from the current one.

Reconfiguring TA, such as moving a cell from its original TA to another, usually requires restarting the cell and consequently results in service interruption. Thus, there is a trade-off between approaching minimum signaling overhead and the cost resulted from reconfiguration. In this study, a bi-objective optimization framework is proposed to solve the TA reconfiguration problem.

Tracking Area List (TAL) is a scheme introduced in 3GPP Release 8 [2]. In this scheme, instead of assigning one TA to each UE, one UE can have a list of TAs. The UE receives a TA list from a cell, and keeps the list, until it moves to a cell that is not included in its list. In LTE standards, a cell is also able to give different lists to different UEs. The UE location is known in the MME to at least the accuracy of the TAL allocated to that UE.

If the information of each individual UE's movement and calls were available to the network, then designing an optimum TAL would become trivial and it could essentially result in the elimination of signaling overhead. However, this information is virtually impossible to obtain. The thesis presents solution approaches and novel analysis to shed light on TAL allocation and assignment.

In LTE, there is a possibility to change the TAL assigned to each cell in short time intervals *without any cost of service interruption.* This is the main reason to explore the dynamic framework of standard TA and TAL configurations in LTE systems.

1.2 Contributions

The main contributions of the thesis can be summarized as follows.

1. Formulating the TA re-optimization problem as an integer programming model. The formulation aims to optimize the trade-off between TAU and paging overheads in a network with a budget constraint on the amount of reconfiguration.

2. Developing a heuristic approach for solving the above trade-off problem close to optimality, by using a repeated local search algorithm.

3. Developing two solution approaches to deliver the pareto-optimal solutions of the bi-objective optimization problem. The computational results of both solution approaches are given for several real-life large-scale networks of various sizes.

4. Exploiting the concept of TAL in order to improve the performance of LTE networks and presenting three algorithms to design TAL for a large-scale network.

5. Exploring the challenges in TAL scheme and suggesting a formulation to calculate the signaling overhead in TAL.

6. A performance comparison of three suggested approaches for assigning and allocating TALs for large-scale networks.

7. A comprehensive study of applying a dynamic TA scheme and comparing its performances with a static scheme.

1.3 Publications

Most parts of the material presented in this thesis have been previously appeared in the following publications.

S. Modarres Razavi and D. Yuan, Performance Improvement of LTE Tracking Area Design: A Re-optimization Approach, in *Proc. of the 6th ACM International Workshop on Mobility Management and Wireless Access (MobiWac '08)*, pages 77-84, 2008.

S. Modarres Razavi, D. Yuan, F. Gunnarsson and J. Moe, Optimizing the Tradeoff between Signaling and Reconfiguration: A Novel Bi-criteria Solution Approach for Revising Tracking Area Design, in *Proc. of IEEE Vehicular Technology Conference (VTC '09-Spring)*, 2009.

S. Modarres Razavi, D. Yuan, F. Gunnarsson and J. Moe, Exploiting Tracking Area List for Improving Signaling Overhead in LTE, in *Proc. of Vehicular Technology Conference (VTC '10-Spring)*, 2010.

S. Modarres Razavi, D. Yuan, F. Gunnarsson and J. Moe, Dynamic Tracking Area List Configuration and Performance Evaluation in LTE, in *Proc. of Global Communications Conference (GLOBECOM Workshop '10)*, 2010.

The bi-objective optimization study has resulted the following journal submission.

S. Modarres Razavi, D. Yuan, F. Gunnarsson and J. Moe, Performance and Cost Trade-off in Tracking Area Reconfiguration: A Pareto-optimization Approach, submitted for journal publication, 2010.

1.4 Thesis Outline

The rest of the thesis is organized as follows.

In Chapter 2, first some previous works on investigating location management schemes are reviewed. Second, the standard TA scheme is explained. Third, the signaling overhead formulation used throughout this work is presented.

Chapter 3 presents the re-optimization approach for revising the TA design. The service interruption caused by TA reconfiguration is explicitly taken into account. The complexity and solution characterization of the resulting optimization problem are investigated. In this chapter, an algorithm which is able to deliver high-quality solutions in short computing time is developed.

Chapter 4 proposes the bi-objective optimization framework to solve the trade-off between the signaling overhead and the cost of TA reconfiguration. To obtain the pareto-optimal solutions, two different approaches have been suggested and compared. For performance evaluation, the approaches have been applied to several real-life large-scale networks.

In Chapter 5, the reader is introduced to the concept of Tracking Area List in LTE systems. This chapter illustrates the potential of TAL by clarifying the limitations of the standard TA scheme. The challenge in applying TAL to a large-scale network is explained.

A formula for calculating the signaling overhead in TAL is proposed in Chapter 6. The chapter presents three algorithms to design TAL with the available data at hand, and discusses the pros and cons of each scheme.

In Chapter 7, the reader is given an approach for generating UE-traces scenarios. Two methods are presented for calculating the overall signaling overhead of the UE-traces scenario, which is used for comparing the standard TA scheme and the three TAL design algorithms suggested in Chapter 6. A thorough study of the numerical results is presented in this chapter to compare the suggested algorithms.

After an introduction to the concept of self-organizing networks, Chapter 8 brings a static and dynamic framework to the STA and TAL configurations. The performance of both STA and TAL schemes are studied according to the frameworks.

In Chapter 9, the author draws some conclusions and gives an overview of possible extensions of the thesis work.

Chapter 2

Tracking Area

In this chapter, some background and basic materials for tracking area planning (TAP) are explained. Moreover the signaling overhead formulation under the standard scheme, which is considered throughout the thesis, is presented.

2.1 Basic Technical Terms

The technical definitions explained in this section are produced by 3GPP in Release 9 [1]. The following terms are used throughout the thesis and the author brings them here as a background to the whole study.

- **Cell** is an area of the radio coverage identified by a base station identification. A hotspot cell is a cell where many users are densely located.

- **MME** is the control plane entity which supports many functions including tracking area list management.

- **Location register** is a function for storing the location information of the users in order for the network to enable the communication.

- **Location Area (LA)** is defined as an area in which a user may move freely without updating the Visitor Location Register (VLR). The LA is related to the CS domain and is the term used in GSM.

The CS domain refers to the set of all the core networks and the related signaling entities offering circuit switched type of connection for user traffic.

- **Routing Area (RA)** is defined as an area in which a user, in certain operation modes, may move freely without updating the Serving GPRS Support Node (SGSN). The RA is related to the PS domain and belongs to GPRS and UMTS networks. An RA is always contained within an LA. The PS domain refers to the set of all the core networks and the related signaling entities offering packet switched type of connection for user traffic.

- **Tracking Area (TA)** is defined as an area in which a user may move freely without updating the MME. TA is a term used in LTE networks. The network allocates a list with one or more TAs to the user. In certain operation modes, the UE may move freely in all TAs of the list without updating the MME.

2.2 Location Management

There is an extensive amount of literature on location management in cellular networks (see, for example [11] for an overview). All the problems related to the LA and RA planning and optimization can be generalized to the study of TA. Throughout this section, the term LA is mostly used, because it is used in the related references. There are some proposed strategies for location management in the literature. In [11], [19], and [66], most of these strategies have been reviewed and categorized. This section tries to summarize the most studied schemes. They can be categorized in two main sections: location area update schemes and paging schemes.

2.2.1 Location Area Update Schemes

The Location Area Update (LAU) procedure begins with an update message from the user over the uplink control channel followed by some signaling which updates the database. Due to the use of network bandwidth and core network communication, for the purpose of modification of location databases, each LAU is a costly exercise.

There are several different schemes to reduce the number of update messages from the users. Usually, the LAU schemes are partitioned into two categories: *static* and *dynamic*. In the static schemes, the LAUs are triggered based on the topology of the network, while in the dynamic ones the LAUs are based on the user's call and mobility patterns. Static schemes allow efficient implementation and low computational requirements as they are independent of user characteristics. Unlike the static schemes, the dynamic ones usually require the online collection and processing of data, which consume significant computing power. However, the dynamic schemes have a higher level of signaling overhead reduction compared to static schemes. Thus, for dynamic schemes in order for the network to support the computation effectively, a careful design is necessary [11].

Examples of Static Update Schemes

- *Always-update*: In this scheme, the user updates its location whenever it moves into a new cell. The network has a complete knowledge of the user's location and no paging is required. This scheme performs well for users with low mobility rates and high call arrival rates. However, this scheme is practically never used, due to excessive LAUs.

- *Never-update*: In this scheme, the user never updates its location, which means that the location update overhead is zero. However it leads to excessive paging for large-scale networks and also networks with high call arrival rates. This scheme is practically never used either.

- *Reporting cells*: In this scheme, the user updates its location only when visiting one of the predefined *reporting cells*. For paging a user, a search must be conducted around the vicinity of the last reporting cell from which the user has updated its location [13]. Without considering the movements of users, it is not possible to assign an optimum arrangement for the reporting cells.

- *Forming LA*: In this scheme, the user updates its location whenever it changes an LA. The paging of a user will occur inside the LA in which the user is located. This scheme is referred to as the standard update scheme, and it is the assumed scheme in the thesis.

Examples of Dynamic Update Schemes

- *Selective LA update*: In this scheme, the LAU is not performed every time the user crosses an LA border. The LAU process at certain LAs can be skipped, as the user might spend a very short period of time in those LAs [57].
- *Time-based*: In this scheme, the user updates its location at constant time intervals. In order to minimize the number of update messages, the time interval can be optimized per user [48].
- *Profile-based*: In this scheme, the network maintains a profile for each user. The profile has a sequential list of the most likely LAs that the user is located at different time periods. The LAs on the list are being paged sequentially from the most to the least likely LA where a user can be found. The profile of each user should be updated from time to time [53, 60].
- *Movement-based*: In this scheme, the user updates its location after a given number of boundary crossings to other cells in the network. The boundary-crossing threshold can be optimized per-user based on its individual movement and call arrival pattern [10].
- *Distance-based*: In this scheme, the user updates its location when it has moved away a certain distance from the cell where it has last updated its location. The distance threshold can be optimized per user based on its individual movement and call arrival pattern [67].
- *Predictive distance-based*: In this scheme, the network determines the probability density function of the user's location based on location and speed reports. The user performs LAU whenever its distance exceeds the threshold measured from the predicted location [35].

2.2.2 Paging Schemes

By paging, the network determines the exact location at cell level of a specific user. Each step in the attempt of determining the location of a user is referred to as a polling cycle. During each polling cycle, polling signals are sent over the downlink control channel to all cells where a user is likely to be present. All the users listen to the paging message and only the called user sends a response message back over the uplink control channel. During the paging process, radio bandwidth is used. Therefore, the paging overhead is proportional to the number of polling cycles, as well as the number of cells being polled in each cycle. In each polling cycle there is a timeout period, and if the user is not found in that time frame, another group of cells will be chosen in the next polling cycle. The maximum paging delay depends on the maximum number of polling cycles allowed for finding the user. Because the goal is to reduce the paging overhead, all paging schemes are based on a prediction of where the user can be located.

Examples of Paging Schemes

- *Blanket polling* (simultaneous paging): In this scheme, all cells in the user's LA are paged simultaneously. This scheme requires no extra knowledge of user location, and it is the most practical and used scheme in current networks. It is also called the standard paging scheme in the thesis.

- *Shortest-distance-first*: In this scheme, the network pages the user by starting from the last cell where the user has updated its location and moving outward based on the shortest-distance-first order.

- *Sequential paging*: In this scheme, the user is paged sequentially in sub-groups of cells in the LA. The sub-groups are ordered in their estimated probabilities of having the user located in them.

- *Velocity paging*: In this scheme, the users are classified based on their velocities at the moments of location updates. In this case, the paging area is dynamically generated based on the user's last LAU time and velocity class index [63].

Beside the above examples, various sequential paging schemes have been proposed in [10, 37, 39, 53, 55, 64]. Although selective LAU and paging schemes discussed here and in the previous section reduce the signaling overhead, their use requires modification of system implementation and collection of additional user information. Hence, the standard scheme remains widely used.

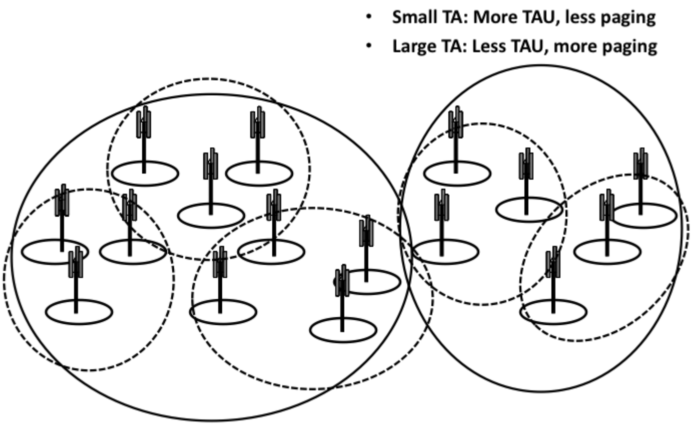


Figure 2.1 An illustration of the TAU and paging trade-off.

2.3 TA Design Optimization

Under the standard scheme of TAU and paging, the main design task is the formation of TAs, with the objective of minimizing the total amount of signaling overhead. Having TAs of very small size (e.g., one cell per TA) virtually eliminates paging, but causes excessive TAU, whereas TAs of too large size give the opposite effect. Thus, the natural objective in TAP is to reach an optimal balance between TAU and paging signaling. Figure 2.1 illustrates the basic trade-off in TAP. Tcha et al. [62] applied mathematical programming to this problem. They presented an integer programming model and a cutting plane algorithm, and reported optimality of a GSM network of 38 cells. Because the problem is NP-hard, solutions to large networks are typically obtained by heuristic algorithms, such as insertion and exchange local search [52], simulated annealing [21], and genetic algorithms [29]. A heuristic based on the notion of matrix decomposition is presented in [12].

In [56], a host of heuristic algorithms for LA design are evaluated in terms of optimality and computational effort. In addition to LA design, the authors of [56] address cell-to-switch assignment for load balancing. Joint LA design and cell-to-switch assignment, under the assumption of hexagon-shaped cells, is solved by a greedy algorithm in [15]. A simulated annealing algorithm for a similar problem is presented in [22].

Multi-layer LA design, where each LA may contain several paging areas, is solved by simulated annealing in [50]. The authors of [34] provide an integer programming model for this problem, and a solution approach based on a graph-partitioning heuristic. In [65], the author makes use of the simulation tools developed by the EU MOMENTUM project [46], originally intended for cell planning, to predict LAU and paging requests. An integer programming model is used for jointly designing LAs, RAs, and UTRAN registration areas (URAs) in [65].

The thesis follows the standard TAU and paging scheme for location management. This means that movement of a UE crossing the TA boundary leads to a TAU message, and paging is performed simultaneously in all cells of the TA to which the UE is currently registered.

2.4 User Equipment States in Mobility Management

Any device used directly by an end-user to communicate through the network is called User Equipment (UE) in LTE. Almost the same concept was previously called Mobile Station (MS) or Mobile Terminal (MT) in previous generations of cellular networks. UE can be a hand-held telephone, a laptop computer or any other device equipped with mobile-broadband adaptor. From a mobility perspective, the UE can be in one of these three states.

- *LTE-Active*: The network knows the cell which the UE belongs to, and UE can transmit and receive data from the network. No TAU/paging is necessary for active UEs.

- *LTE-Idle*: The network knows the location of the UE at the granularity of a few cells (forming a TA). In the idle mode, the UE is in power-conservation mode and does not inform the network of each cell change.

- *LTE-Detached*: In this mode either the UE is powered off or it is in the transitory state in which the UE is in the process of searching and registering to the network.

Frequently, the UE will be in the LTE-Idle state, and the MME knows the TA in which the UE is last registered. Usually, the only available realistic data from a cellular network are the cell load and cell

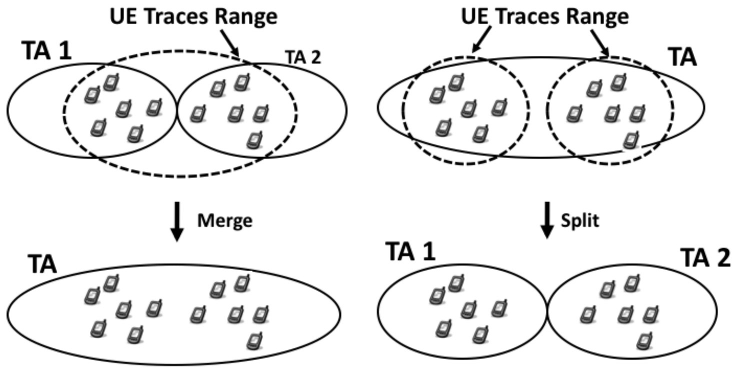


Figure 2.2 Merge and split of TAs.

handovers. Cell load and handover belong to active UEs. Cell load and handover statistics can be a good estimation of UE's location and movement, assuming that idle UEs are having the same mobility behavior as the active ones. Other approaches for estimating the behavior of idle UEs include network simulation [65] and examining traffic density on roads across neighboring cells [16]. Although the technical terms cell load and handover are generally representing the active UEs, in the thesis they are considered to represent the distribution and mobility of idle UEs.

A UE trace is defined as the cell-to-cell movement behavior and the call arrival pattern of a UE in a specific time period. Having information related to the UE traces would significantly help in reducing the signaling overhead and optimizing the TA configuration [69]. From the below example it can be concluded that even a rough estimation of the UE traces can be useful in planning and optimizing TAs.

- Example: In Figure 2.2 the range of UE traces movement is known for the specified area. In the left figure, the UE-traces range suggests that TA1 and TA2 should merge. In the right figure the separation of UE traces indicates that by splitting the TA into two smaller TA, the signaling overhead is reduced.

2.5 Basic Notations and Signaling Overhead Calculation

The set of cells in a network is denoted by $\mathcal{N} = \{1, \ldots, N\}$, and the set of TAs currently in use is denoted by $\mathcal{T} = \{1, \ldots, T\}$. The vector $\mathbf{t} = [t_1, \ldots, t_N]$ is used as a general notation of cell-to-TA assignment, where t_i is the TA of cell i. TA design $\mathbf{t}$ can be alternatively represented by an $N \times N$ symmetric and binary matrix $\mathbf{S}(\mathbf{t})$; in which element $s_{ij}(\mathbf{t})$ represents whether or not two cells are in the same TA, i.e.,

$$s_{ij}(\mathbf{t}) = \begin{cases} 1 & \text{if } t_i = t_j, \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

Let u_i be the total number of UEs in cell i scaled by the time proportion that each UE spends in cell i. For the same time period, h_{ij} is the number of UEs moving from cell i to cell j. The values of u_i and h_{ij} can be assessed by cell load and handover statistics of active UEs. The amount of overhead of one paging and one update are denoted by c^p and c^u, respectively. The exact relationship between c^u and c^p depends on the radio resource consumption. Moreover, parameter α is the call intensity factor/activity factor (i.e., probability that a UE has to be paged). The total update and paging signaling overhead is defined by $c_{SO}(\mathbf{t})$ and is calculated by Equation (2.2):

$$c_{SO}(\mathbf{t}) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}: j \neq i} \left(c^u h_{ij} (1 - s_{ij}(\mathbf{t})) + \alpha c^p u_i s_{ij}(\mathbf{t}) \right) \tag{2.2}$$

Within the outer parentheses of (2.2), the first term accounts for the TAU overhead for UEs moving from i to j (if the two cells are not in the same TA). The second term is the paging overhead introduced in cell j while paging UEs in cell i (if the two cells are in the same TA).

Chapter 3

TA Re-optimization

The optimized TA configuration in the planning phase will not perform satisfyingly after some time period, due to changes in UE distribution and mobility patterns. For re-optimizing the configuration over time, it is not practically feasible to deploy a green-field design, as it might significantly differ from the original configuration. By re-optimization, the design is successively improved by re-assigning some cells to TAs other than their current ones.

There are two reasons for applying a re-optimization approach. First, reconfiguring TAs, such as moving a cell from one TA to another, typically requires temporarily tearing down the cell and thus service interruption – a very costly process from the service standpoint. Second, the benefit of a new, optimized TA design gradually diminishes over time as UE location and mobility patterns change. Thus, one has to weigh the performance improvement of some limited time duration against the cost in terms of service interruption due to reconfiguration. The service interruption aspect is accounted by bounding the amount of UEs that are affected by TA reconfiguration. Here, this bound is referred as the budget.

In this chapter, a re-optimization approach for revising TA design is presented. The service interruption caused by TA reconfiguration is explicitly taken into account. The complexity and solution characterization of the resulting optimization problem are investigated. Finally, an algorithm which is able to deliver high-quality solutions in short computing time is developed. The study in this chapter has been previously published in [41].

3.1 Problem Definition

The most basic and convenient reconfiguration option is used as the building element of re-optimization: to move a cell from its current TA to a new one. That is, the output of the re-optimization process consists of a subset of cells that have changed TAs, and the new TA of each of these cells. Before discussing the details, it is worth remarking that the resulting gain of re-optimization, in terms of reduced total paging and TAU overhead, is a joint effect of the re-assignments, i.e., whether or not a cell should change TA, and to which TA the cell should move, depend on the decisions made for other cells.

For TA re-optimization, the TA design currently deployed in the network is given. This solution is denoted by $\mathbf{t}^0$. If the result of re-optimization is $\mathbf{t}^*$, then reconfiguration means to move all cells i from t_i^0 to t_i^* for which $t_i^0 \neq t_i^*$. The reduction of the number of TAs is allowed, it means that if a TA becomes empty after cell moves, it is simply deleted. To simplify the presentation, increasing the total number of TAs is not considered, although the solution algorithm can be easily extended to include this option.

For every cell, a parameter is defined to represent the cost in service interruption, if the TA of the cell is changed. For convenience and without loss of generality, the UE distribution parameter u_i is used to measure the amount of service interruption of cell i. Let $\mathbf{d}(\mathbf{t}, \mathbf{t}^0)$ be a binary vector representing cells that have been assigned new TAs, that is, $d_i(\mathbf{t}, \mathbf{t}^0) = 1$ if and only if $t_i^0 \neq t_i$, $i \in \mathcal{N}$. Denoting the budget value by B, the following budget constraint is introduced.

$$\sum_{i \in \mathcal{N}} u_i d_i(\mathbf{t}, \mathbf{t}^0) \leq B \tag{3.1}$$

The TA re-optimization (TAR) problem is formalized below.

[**TAR**] Find a TA design $\mathbf{t}$ that satisfies the budget constraint (3.1) and minimizes the total overhead cost $c_{SO}(\mathbf{t})$ as defined in Section 2.2.

Remark 1. A closely related problem, considered in most of the references in Chapter 2, is to make a TA design completely from scratch. Here, this green-field-design problem is referred as tracking area optimization (TAO). The optimum to TAO is a lower bound to the best achievable performance of TAR. This value will be used as a reference in performance evaluation.

3.2 Complexity and Solution Characterization

TAR turns into TAO if the budget constraint is removed. TAO is known to be NP-hard [62]. Bejerano et al. [14] showed that TAO remains *NP*-hard even over a star (i.e., one cell is the only and common neighbor to all other cells).

The above facts do not prove that TAR is *NP*-hard. Its complexity result, assuming (3.1) is *non-redundant*, is formalized in the following proposition.

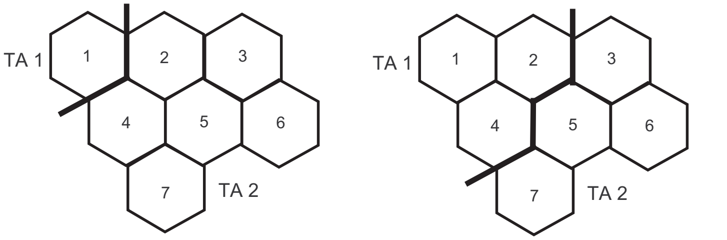
PROPOSITION 1. TAR remains NP-hard when the budget constraint (3.1) is non-redundant.

Sketch of a PROOF. Observing that (3.1) is a knapsack constraint, it can be shown that any instance of the binary knapsack problem can be transformed to an instance of TAR. In the transformation, every item in the knapsack problem corresponds to moving a cell from its current TA to a new one, with the handover values set such that the cell move leads to an improvement in the total overhead cost. The improvement is equal to the objective function coefficient of the knapsack instance. Moreover, no additional improvement is possible other than these moves. Finally, each of these moves is independent from the others, i.e., the improvement of a move is not affected by any of the other moves. Then the two instances become equivalent. $\square$

The following proposition provides a solution characterization.

PROPOSITION 2. If there is no budget limit and any number of TAs is allowed, then a solution is non-optimal if it contains some TA, of which the cells can be partitioned into two (or more) subsets $\mathcal{N}_1$ and $\mathcal{N}_2$, such that there is no handover between the subsets, i.e., $h_{ij} = 0$ for all $i \in \mathcal{N}_1$ and $j \in \mathcal{N}_2$.

PROOF. Suppose the cells in $\mathcal{N}_1$ form a new TA. The TAU overhead does not increase, because any update due to UE mobility from any cell in $\mathcal{N}_1$ to another TA is present before the new TA is formed, and there are no UE movements between cells in $\mathcal{N}_1$ and $\mathcal{N}_2$. The paging overhead



(a) Moving any single cell leads to higher overhead

(b) Improvement by moving two cells

Figure 3.1 An example of the dependency between cell moves.

goes down due to TA split. Hence the conclusion. □

What is stated in Proposition 2 is in fact very intuitive from a network planning point of view: Assuming that the amount of handover $h_{ij} > 0$ *if and only if* cell i and j are geographically adjacent, then in an optimal design of TAO, every TA consists of geographically connected cells. For TAR, the result does not always hold in theory because of the budget constraint and the limit of using at most T TAs. Nevertheless, it tends to be satisfied for practically relevant planning scenarios. This greatly reduces the computational effort in the repeated local search algorithm (see Section 3.3).

Although the complexity result of TAR makes use of the knapsack problem, the former is considerably harder in practice, simply because the changes in the total signaling overhead due to cell moves are dependent on each other.

- Example: Figure 3.1 illustrates the dependency using a simple example of two TAs and seven cells. The boundary between the TAs is shown by the thick lines. All cells have u UEs, and the amount of handover in both directions together is h for all pairs of adjacent cells. For simplicity, let $c^u = c^p = 1$, and $\alpha = 0.1$. The total signaling overhead of the current TA design is $2h + 3u$ (Figure 3.1(a)). Assume h is between $0.4u$ and $0.6u$. It can be verified that moving any single cell from its TA to the other TA (including moving cell 1 and making TA 1 empty) results in higher total overhead. However, there is an improvement if both cells 2 and 4 are moved to TA 1 (Figure 3.1(b)).

The above example illustrates the phenomenon of local optima. Problem TAR is further complicated by the budget constraint, because a collection of cell moves may not be feasible. The solution algorithm presented in Section 3.3 considers these aspects by allowing for some non-improving moves, but limiting the amount of budget they may consume.

3.3 A Solution Approach Based on Repeated Local Search

Solving TAR to optimality may require excessive computational effort in view of its complexity. In this chapter, a simple but effective heuristic algorithm is developed using repeated local search to find high-quality solutions rapidly.

3.3.1 Local Search

The local search algorithm iteratively updates the TA design. In every iteration, the algorithm considers cells that may be moved in respect of the remaining budget, and among these cells selects the cell move that results in the largest improvement. This is repeated until no additional move of any cell is allowed because of the budget limitation or no further improvement can be obtained.

In its first run, the initial solution is $\mathbf{t}^0$, and the local search behaves like a greedy algorithm that successively builds up a solution of TAR. In subsequent runs, solution initialization follows the procedure in Section 3.3.2. The local search algorithm is formalized in Figure 1, in which the solution given to and returned by the algorithm is denoted by $\mathbf{t}^\ell$.

Remark 2. Because $\mathbf{t}^\ell$ is not necessarily equal to $\mathbf{t}^0$ when the algorithm starts, some cells may have been moved from their original TAs in the initial solution $\mathbf{t}^\ell$. Therefore, in Step 4, which constructs the set of cells to be considered for move, the budget constraint (3.1) is checked only if a cell is still in its original TA, as otherwise the corresponding contribution to the left-hand side of (3.1) is already accounted in b^ℓ. For the same reason, in Step 19, b^ℓ decreases (i.e., some of the budget becomes released) if a cell is moved back to its original TA.

Remark 3. In Step 6, the set $\mathcal{T}'$ contains candidate TAs to which cell i may be moved. Motivated by Proposition 2, TAs that at present do not have any cell with positive handover value to cell i are excluded. As a

Algorithm 1 Local_Search

$b^\ell = \sum_{i:t_i^\ell \neq t_i^0} u_i$
repeat
$\delta^* = 0$; $i^* = -$; $t^* = -$;
$\mathcal{N}' = \{i \in \mathcal{N} : t_i^\ell \neq t_i^0 \text{ or } t_i^\ell = t_i^0 \text{ and } b^\ell + u_i \leq B\}$
for all $i \in \mathcal{N}'$ **do**
$\mathcal{T}' = \{m \in \mathcal{T} : \exists j \in \mathcal{N}, t_j^\ell = m \text{ and } h_{ij} > 0\} \setminus \{t_i^\ell\}$
for all $m \in \mathcal{T}'$ **do**
$\mathbf{t}' = \mathbf{t}^\ell$; $t_i' = m$;
if $c_{SO}(\mathbf{t}^\ell) - c_{SO}(\mathbf{t}') > \delta^*$ **then**
$\delta^* = c_{SO}(\mathbf{t}^\ell) - c_{SO}(\mathbf{t}')$; $i^* = i$; $m^* = m$;
end if
end for
end for
if $\delta^* > 0$ **then**
if $t_{i^*}^\ell = t_{i^*}^0$ **then**
$b^\ell = b^\ell + u_{i^*}$;
else
if $m^* = t_{i^*}^0$ **then**
$b^\ell = b^\ell - u_{i^*}$;
end if
end if
$t_{i^*}^\ell = m^*$;
end if
until $\delta^* = 0$
return $\mathbf{t}^\ell$;

result, the size of $\mathcal{T}'$ is much smaller than $T-1$, leading to a significant speed-up of the algorithm. In theory, excluding TAs in this way may overlook some possible improvements, whereas in practice there is no noticeable performance degradation.

3.3.2 Repeated Local Search

Additional improvements can be obtained by applying the local search algorithm repeatedly using different starting solutions. However, to be effective, the initial solutions should satisfy two conditions. First, there must be some slack budget to allow for moving cells from their original TAs. Second, the initial solution should not be a completely randomized one (with a very high total signaling overhead), otherwise no good solution can be found before the entire budget is consumed. Moreover, from the structure of TAR, it is expected that good solutions will have some cell moves in common.

Based on the above observations, an initial solution is constructed as follows. Let $\mathbf{t}^*$ be the best solution so far. Cells are partitioned into two subsets $\mathcal{N}^0$ and $\mathcal{N}^1$, containing cells that remain in the same TA as in the original design $\mathbf{t}^0$, and cells that have been assigned to new TAs by $\mathbf{t}^*$, respectively. A two-step perturbation to $\mathbf{t}^*$ is applied. Two budget parameters, B^1 and B^0, with $B^1 < B^0 < B$, are used. In the first step of perturbation, some randomly chosen cells in $\mathcal{N}^1$ are moved back to their original TAs in $\mathbf{t}^0$, such that the consumed budget becomes less than or equal to B^1, that is, the amount of slack is at least $B - B^1$. Next, some cells in $\mathcal{N}^0$, again chosen randomly, are moved from their TAs to new ones, until the consumed budget reaches B^0. Moving a cell $i \in \mathcal{N}^0$ to a new TA is performed in a greedy manner. That is, the cell is moved to the TA giving the largest improvement, if such TA exists, otherwise the cell is moved to the TA such that the increase in overhead is minimal. This second step of perturbation is aimed at exploring improvements that come from joint effect of multiple cells (see Section 3.2), although none of these moves alone results in improvement.

Figure 2 formalizes the repeated local search algorithm. In the first step, local search is applied to the original TA design $\mathbf{t}^0$. Then perturbation combined with local search are performed K times.

Algorithm 2 Repeated_Local_Search

$\mathbf{t} = \textbf{Local\_Search}(\mathbf{t}^0)$;
$\mathbf{t}^* = \mathbf{t}^0$; $c^*_{SO} = c_{SO}(\mathbf{t}^*)$
for $k = 1 : K$ **do**
  $\mathbf{t}^\ell = \mathbf{t}^*$;
  $b^\ell = \sum_{i:t_i^\ell \neq t_i^0} u_i$;
  $\mathcal{N}^0 = \{i \in \mathcal{N} : t_i^\ell = t_i^0\}$; $\mathcal{N}^1 = \{i \in \mathcal{N} : t_i^\ell \neq t_i^0\}$;
  while $b^\ell > B^1$ and $\mathcal{N}^1 \neq \emptyset$ **do**
    Select randomly a cell $i \in \mathcal{N}^1$;
    $t_i^\ell = t_i^0$;
    $b^\ell = b^\ell - u_i$;
    $\mathcal{N}^1 = \mathcal{N}^1 \setminus \{i\}$;
  end while
  while $b^\ell < B^0$ and $\mathcal{N}^0 \neq \emptyset$ **do**
    Select randomly a cell $i \in \mathcal{N}^0$ with $b^\ell + u_i \leq B$;
    $\mathcal{T}' = \{m \in \mathcal{T} : \exists j \in \mathcal{N}, t_j^\ell = m \text{ and } h_{ij} > 0\} \setminus \{t_i^\ell\}$;
    $m^* = \text{argmin}_{m \in \mathcal{T}'} c([t_1^\ell, \ldots, t_{i-1}^\ell, m, t_{i+1}^\ell, \ldots, t_N^\ell])$;
    $\mathbf{t}_i^\ell = m^*$;
    $b^\ell = b^\ell + u_i$;
    $\mathcal{N}^0 = \mathcal{N}^0 \setminus \{i\}$;
  end while
  $\mathbf{t} = \textbf{Local\_Search}(\mathbf{t}^\ell)$;
  if $c_{SO}(\mathbf{t}) < c^*_{SO}$ **then**
    $c^*_{SO} = c_{SO}(\mathbf{t})$; $\mathbf{t}^* = \mathbf{t}$;
  end if
end for
return $\mathbf{t}^*$;

3.4 Numerical Results

Here the results of performance evaluation using realistic data representing a cellular network for the downtown area of Lisbon, provided by the EU MOMENTUM project [46] is presented. The network consists of 60 sites and 164 cells. A reference scenario of UE distribution and mobility is defined by accumulating the cell load and handover statistics in the data set. Figure 3.2 illustrates the network and the reference scenario. The sites are represented by disks. For every site, its cells are illustrated by squares. The location of a square in relation to its site center shows the direction of cell antenna. The darkness of each cell is set in proportion to accumulated cell load. A link is drawn between two cells if there is any handover between them, and the amount of handover is proportional to the thickness of the link.

Two additional scenarios (I and II) are generated by modifying the cell load and handover statistics. Scenario II has larger deviation from the reference one than scenario I. Provided that the location and mobility patterns have evolved from the reference scenario into each of the two scenarios, the TA re-optimization is conducted. Figure 3.3 illustrates scenario I in the same format as for the reference scenario. In all three scenarios, 5% of the UEs are paged in every cell (i.e., $\alpha = 0.05$). The overhead of a single update c^u is set twice as much as c^p.

The reference scenario in Figure 3.2 represents UE location and mobility patterns to which $\mathbf{t}^0$ is optimal. For this optimization, the model in [62] and software CPLEX [31] are used. Computing the solution is time-consuming. In practicing TAR, $\mathbf{t}^0$ is the design currently in use and hence this computation is not needed. The resulting TA design $\mathbf{t}^0$ is shown in Figure 3.4. There are 44 TAs in the design. In the figure, two cells are connected by an edge if and only if they are in the same TA. Thus, TAs are represented by fully connected subsets of cells. One can observe that, if two cells have a large amount of handover (see Figure 3.2), then they are in the same TA in Figure 3.4.

In addition to $\mathbf{t}^0$, the optimal green-field TA designs for scenarios I and II are also computed and denoted by $\mathbf{t}^*(\text{I})$ and $\mathbf{t}^*(\text{II})$, respectively. The two solutions are attainable only if it is allowed to re-optimize TAs disregarding the budget constraint. Similar to computing $\mathbf{t}^0$, finding these two solutions is hardly feasible for large-scale networks. For the Lisbon network, they can be obtained, although the computing time is long. In order to assess the effectiveness of the algorithm, $\mathbf{t}^*(\text{I})$ and

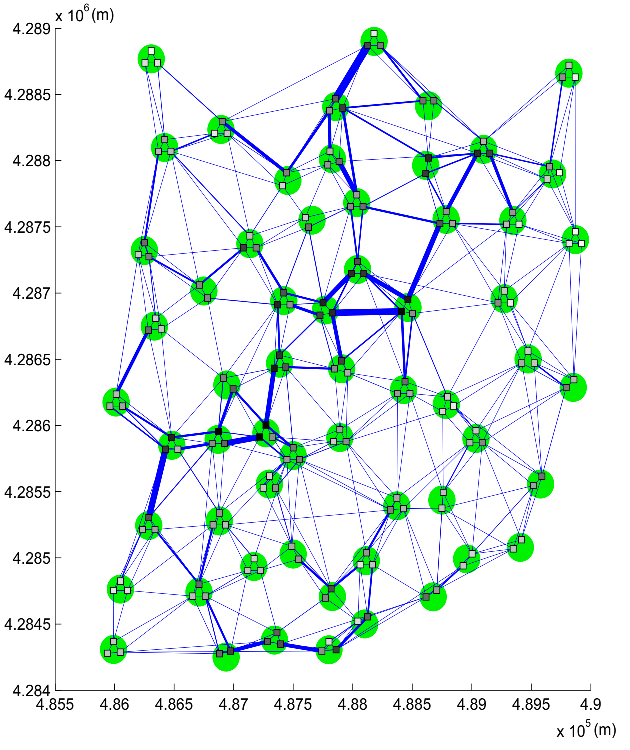


Figure 3.2 An illustration of the reference scenario.

$\mathbf{t}^*(\mathrm{II})$ are used as bounds on the best achievable performance of TAR.

In the repeated local search algorithm, $B^1 = 0.85B$, $B^0 = 0.95B$, and $K = 100$ are set. The computing time is about 30 seconds on a notebook. The processor is of type Intel Core 2 Duo and the clock speed is 2.0 GHz. For each of the scenarios I and II, two budget levels of B, corresponding to 5% and 15% of the total cell load, i.e., $B = B' \cdot \sum_{i \in \mathcal{N}} u_i$ where $B' = 5\%$ and 15%, are used. For performance evaluation, the algorithm without budget limitation ($B' = 100\%$) is also run and compared to $\mathbf{t}^*(\mathrm{I})$ and $\mathbf{t}^*(\mathrm{II})$.

The computational results are summarized in Table 3.1. For the two scenarios, the total overhead values of the initial TA design are shown in row $\mathbf{t}^0$. These values represent the TA performance when the initial

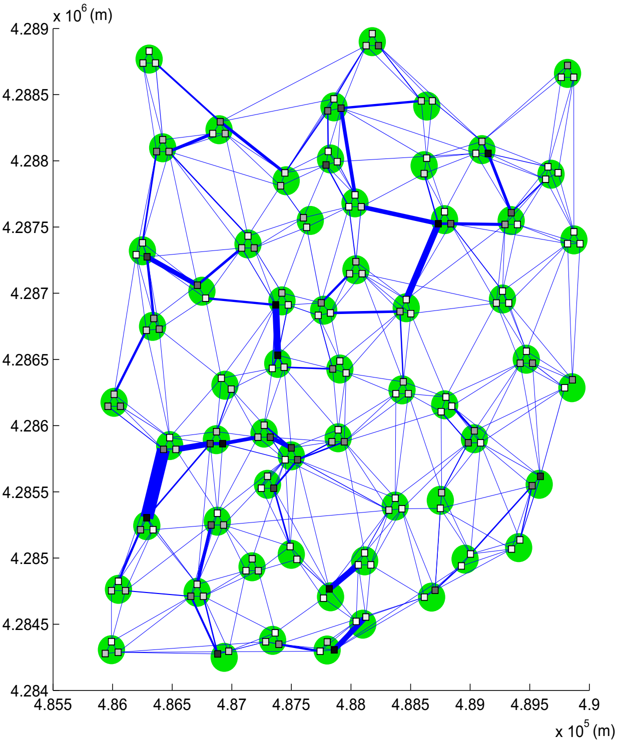


Figure 3.3 An illustration of scenario I.

TA design $\mathbf{t}^0$ is kept for the two scenarios. The results of how much re-optimization improves TA performance for the two budget levels are also reported ($B' = 5\%$ and $B' = 15\%$). The last row displays the optimal solutions with unlimited budget and number of TAs.

From the table, it can be observed that the original TA design $\mathbf{t}^0$, optimized for the reference scenario, is about 20% and 36% away from optimum for scenarios I and II, respectively. By running local search once, it is possible to improve $\mathbf{t}^0$ considerably. An additional amount of improvement is obtained by repeated local search. The improvement grows when B' increases from 5% to 15%; the difference is larger for scenario II because its UE distribution and mobility patterns deviate more from the reference scenario. Moreover, for both scenarios, there

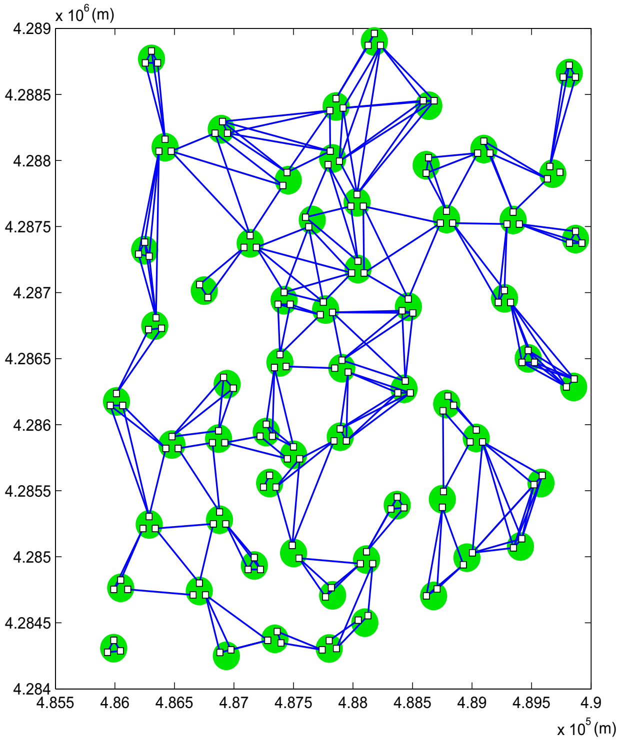


Figure 3.4 TA design $\mathbf{t}^0$ (optimum of the reference scenario).

Table 3.1 Results of TA re-optimization.

(LS = Local search; RLS = Repeated local search.)

	Scenario I		Scenario II	
$\mathbf{t}^0$	292.68		386.62	
	LS	RLS	LS	RLS
$B' = 5\%$	261.52	257.13	386.62	380.03
$B' = 15\%$	257.56	250.25	376.42	354.96
$B' = 100\%$	257.56	245.70	376.42	336.96
	$\mathbf{t}^*$(I)=243.05		$\mathbf{t}^*$(II)=333.73	

is no difference in the solutions of local search for $B' = 15\%$ and $B' = 100\%$. In other words, local search is not able to improve its solution

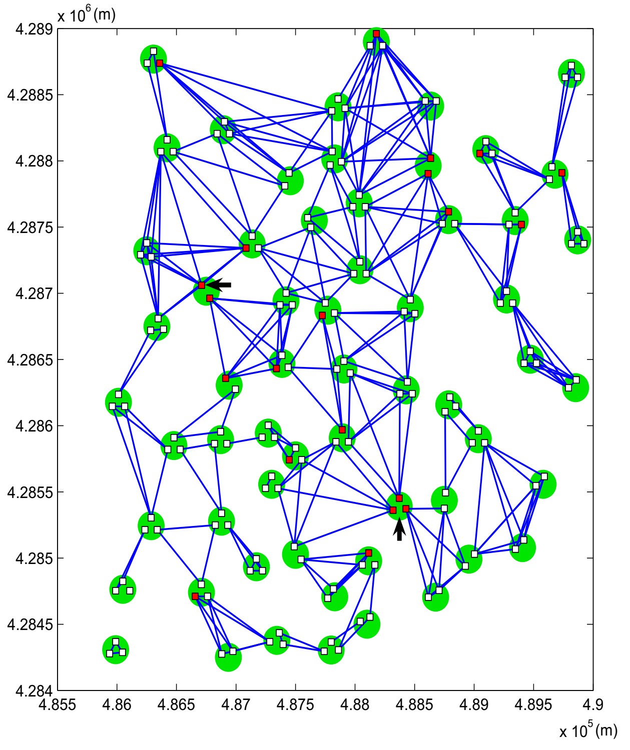


Figure 3.5 Re-optimized TA design for scenario I, $B' = 5\%$.

further even if more budget is made available, because the algorithm already reaches a local optimum for $B' = 15\%$. The results of repeated local search show its strength of overcoming this issue. The effectiveness of repeated local search is further demonstrated by the solutions for $B' = 100\%$. In this case the algorithm's performance is very close to the best achievable – the deviation to optimum is less than 1% for both scenarios.

Figure 3.5 illustrates the re-optimized TA design for scenario I and $B' = 5\%$. In total, 21 cells have changed TAs. These cells are marked in color (red) in the figure. Comparing the solution to $\mathbf{t}^0$, one can see that re-optimization adapts TA design from the reference scenario (Figure 3.2) to scenario I (Figure 3.3). For example, the cell pointed out by the horizontal arrow in Figure 3.5 changed TA, most likely because of the growth in its UE mobility to another cell. At one site, indicated by the vertical arrow, the three cells that were in the same TA have been split into different TAs as a result of fewer numbers of UEs in these cells.

3.5 Conclusions

A re-optimization approach is presented to adapt a given TA design to changes and trends of UE location and mobility patterns. As a novelty of the approach, the cost of reconfiguring TAs is accounted by means of a budget constraint. This is justified by the fact that once a TA design is in use, adopting a new solution of green-field optimization is typically not feasible or does not pay off in real networks. The complexity of the problem is investigated, and a fast algorithm based on repeated local search is developed. The case study on a realistic network shows that the algorithm is able to approach high-quality solutions.

Chapter 4

Performance and Cost Trade-off in TA Reconfiguration

According to the discussion in the previous chapter, reconfiguring TA usually requires to restart the cells which are changing TAs, and consequently results in service interruption. In this chapter, a bi-objective optimization framework is proposed to solve the trade-off between approaching minimum signaling overhead and the cost resulted from the reconfiguration.

Unlike mono-objective optimization problems which have unique optimal values, in bi-objective problems the solution set is formed by pareto-optimal (non-dominated) points. An integer programming model is developed to optimize the overhead by reconfiguration given a specific cost budget constraint. Applying the proposed model to various budget levels leads to a set of pareto-optimal solutions. Depending on the number of pareto-optimal solutions, the integer model may have to be run many times. Solving the integer programming model is very time-consuming and sometimes infeasible for large networks.

To deal with large-scale networks, a genetic algorithm (GA) embedded with local search (LS) is proposed. The algorithm searches for pareto-optimal solutions in one single run. In the GA approach, the concept of dominance in the fitness evaluation is used contrary to the approaches that use a scalarization function or treat the various objectives separately. In the GA algorithm, the amount of dominance explicitly evaluates each solution in terms of pareto-optimality.

The performance of the proposed integer model and GA algorithm is demonstrated via experiments using three large-scale realistic/real-life network scenarios. For the first two scenarios, it was possible to compare the results from the GA algorithm with the ones computed from the integer model. The last network was only solved by the GA algorithm since it was too large and not feasible to be solved with the integer programming model. The results demonstrate the ability of the approaches to deliver various pareto-optimal solutions, and thus giving the operator the opportunity of selecting a proper trade-off between the two objectives. The research presented in this chapter has appeared in [42, 45].

4.1 System Model

Generation of pareto-optimal or non-dominated solutions is the primal goal in solving bi-objective problems. A solution is called *pareto-optimal* if it is not possible to improve a given objective without deteriorating at least another objective [61]. Clearly it does not make sense to choose a solution that is not pareto-optimal. A large amount of references for multi-objective optimization are available in the literature [58, 59, 61].

The system model considered in this chapter is an extension of the definitions described in Sections 2.5 and 3.1, with some modifications described below. The signaling overhead follows (2.2), and for convenience it is re-stated below.

$$c_{SO}(\mathbf{t}) = \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}: j \neq i} (c^u h_{ij}(1 - s_{ij}(\mathbf{t})) + \alpha c^p u_i s_{ij}(\mathbf{t})) \tag{4.1}$$

The cost of reconfiguration is denoted by $c_R(\mathbf{t})$, and it is computed by (4.2), where t^0 is the TA design currently deployed in the network. Equation (4.2) follows the cost definition in the previous chapter.

$$c_R(\mathbf{t}) = \sum_{i \in \mathcal{N}} u_i d_i(\mathbf{t}, \mathbf{t}^0) \tag{4.2}$$

The aim is to observe the trade-off between $c_{SO}(\mathbf{t})$ and $c_R(\mathbf{t})$ of the design $\mathbf{t}$; thus, the problem is modeled with the following bi-objective formulation.

$$\min(c_{SO}(\mathbf{t}), c_R(\mathbf{t})) \tag{4.3}$$

subject to:

$$s_{ij}(\mathbf{t}) = \begin{cases} 1 & \text{if } t_i = t_j, \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

$$d_i(\mathbf{t}, \mathbf{t}^0) = \begin{cases} 1 & \text{if } t_i^0 \neq t_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.5}$$

4.2 An Integer Programming Model

To solve the bi-objective problem formulated in (4.3)-(4.5), one approach is to minimize $c_{SO}(\mathbf{t})$ defined in (4.1) for various reconfiguration cost budgets. In other words, the TA re-optimization problem is solved repeatedly for different limits on $c_R(\mathbf{t})$. By denoting the budget value by B, the budget corresponds to the constraint $c_R(\mathbf{t}) \leq B$ in a binary integer programming model. The model has two sets of binary variables:

- s_{ij} is 1 when i and j are in the same TA and 0 otherwise.
- p_{it} is 1 when cell i belongs to TA t and 0 otherwise.

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}: j \neq i} (c^u h_{ij}(1 - s_{ij}) + \alpha c^p u_i s_{ij}) \tag{4.6}$$

subject to:

$$\sum_{t \in \mathcal{T}} p_{it} = 1, \forall i \in \mathcal{N} \tag{4.7}$$

$$p_{it} + p_{jt} - 1 \leq s_{ij}, \forall i, j \in \mathcal{N}, t \in \mathcal{T} \tag{4.8}$$

$$s_{ij} + p_{it} - 1 \leq p_{jt}, \forall i, j \in \mathcal{N}, t \in \mathcal{T} \tag{4.9}$$

$$s_{ij} + s_{jk} - s_{ik} \leq 1, \forall i, j, k \in \mathcal{N}, i \neq j \neq k \tag{4.10}$$

$$\sum_{i \in \mathcal{N}} u_i(1 - p_{it_i^0}) \leq B \tag{4.11}$$

In the presented model, constraint (4.7) assures that each cell is assigned to only one TA. Constraints (4.8) and (4.9) define the matrix $\mathbf{S}(\mathbf{t})$ and the correlation between s_{ij} and p_{it}. When $p_{it} = p_{jt} = 1$, it means that i and j are in the same TA t, and hence $s_{ij} = 1$ as imposed by constraint (4.8). If $p_{it} = 1$ and $p_{jt} = 0$, then i belongs to TA t while j does not, and therefore $s_{ij} = 0$ (constraint (4.9)). Constraint (4.10) ensures that if two cells i and k belong to the same TA as cell j, they must also be in the same TA. That is, if $s_{ij} = s_{jk} = 1$, constraint (4.10) becomes $s_{ik} \geq 1$, forcing $s_{ik} = 1$. Constraint (4.11) bounds the number of UEs affected by reconfiguration using the budget level. From the definition of the variable p_{it}, it is clear that $p_{it_i^0}$ is one when cell i belongs to the current TA t_i^0 and zero otherwise.

For $B = 0$, the current $\mathbf{t}^0 = [t_1^0, t_2^0 \dots t_i^0 \dots t_N^0]$ is the only feasible solution. The signaling overhead of this configuration is likely not optimum, but on the other hand the corresponding cost is zero. This point is among the pareto-optimal solutions, as one cannot find any solution with better reconfiguration cost. The other pareto-optimal solutions can be calculated by giving other values of B.

4.3 Dominance-based Approach

The solution space of the problem, depending on the scale of the network, can be very large as it is a combinatorial bi-objective problem. To achieve high quality solutions, two aspects should be considered. One is the convergence to the pareto optimal front, and the other aspect is having diversity in the search procedure. In view of this and the complexity results in Section 3.2, it is motivated to apply meta-heuristics to deal with this problem for large-scale networks and to deliver the pareto-optimal solutions in a single run. Multi-objective meta-heuristics can be classified into four main categories based on their solution evaluation strategies.

- *Scalar approaches* transform the problem into a mono-objective problem. A typical example is the *weighted sum* method, which combines the objective functions by non-negative weights and converts them into one objective function [32]. Another example would be the *goal programming* method that uses a target value for each objective function, and the overall goal is to minimize the deviation from the target values [18].

- *Criterion-based approaches* which treat the various incommensurable objectives separately, such as the *parallel* approach [51] and the *lexicographic* approach [25]. In the latter, to evaluate a solution against another, the two objective function vectors are compared lexicographically.

- *Indicator-based approaches* which use performance quality indicators as a search guide [68].

- *Dominance-based approaches* use the concept of dominance in solution evaluation [9].

Among the approaches above, weighted sum is a frequently used method for solving multi-objective optimization [32]. This approach is however not used for the problem here for three reasons: *First* our problem is a combinatorial bi-objective problem, as the configuration solutions are described by discrete variables. Thus, the number of pareto-optimal solutions can be exponential in the problem size [23]. *Second*, there may exist pareto-optimal solutions which cannot be resulted from any weighted sum of the objective functions. *Third*, to obtain a diverse set of pareto-optimal solutions by the weighted sum approach, multiple runs of different weight combinations are required. In general, setting the weights is a difficult task.

A dominance-based approach is used to evaluate the solutions, by defining a parameter called Preference Value (PV) for each solution $\mathbf{t}$. PV($\mathbf{t}$) is the number of other solutions which are dominating solution $\mathbf{t}$. This means that a solution with PV $= 0$ is among the pareto-optimal ones.

- Example: Figure 4.1 shows the PV values for a set of solutions. Consider the signaling overhead cost $c_{SO}(\mathbf{t})$ and the reconfiguration cost $c_R(\mathbf{t})$ as the two objective functions of the problem. In this figure, there is a point that is dominated by two other solutions, therefore for this point PV $= 2$. The points with PV $= 0$ are the pareto-optimal solutions.

4.4 Genetic Algorithm

Genetic Algorithm (GA) [28] embedded with Local Search (LS) [8] is used. The two reasons for choosing a GA approach are:

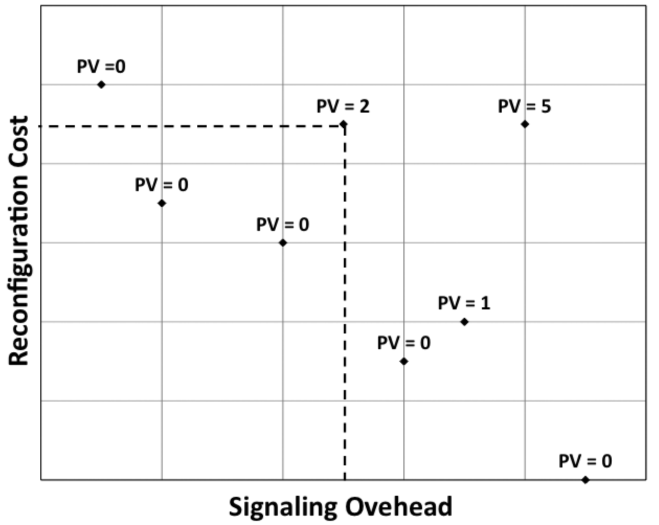


Figure 4.1 An illustration of the PV definition.

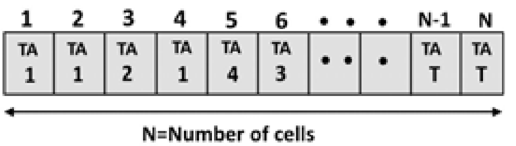


Figure 4.2 Solution vector representation.

1. The encoding of solutions is simple by means of integer-valued vectors.

2. The desired pareto-optimal solutions form a population of solutions. Thus, a population-based meta-heuristic approach is a reasonable algorithm candidate.

For solution encoding, a fixed length vector of size N is used. The elements in the vector represent the TA numbers which the cells belong to. Figure 4.2 illustrates the solution vector representation.

Figure 4.3 summarizes the principle design of the solution algorithm. In this figure, POPSIZE is the population size considered in GA. The sizes of the outputs from the crossover and mutation operators are also equal to POPSIZE. The initial pool is a set of high-quality solutions, which are considered in the initial phase of the algorithm. Iteration

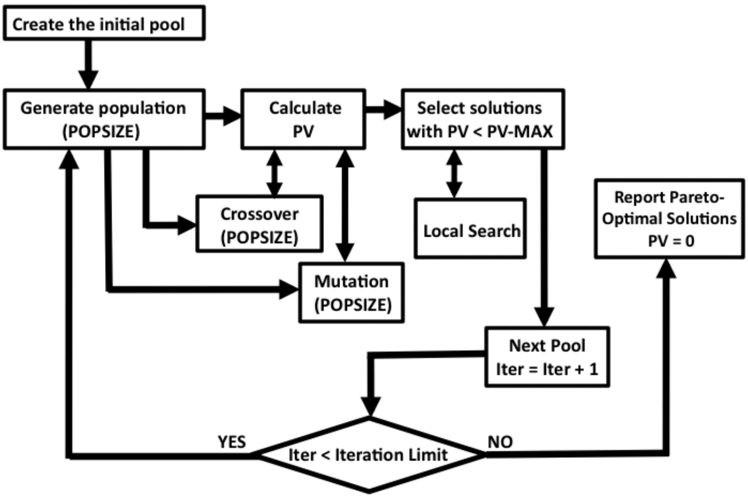


Figure 4.3 Principle design in finding pareto-optimal configurations.

Limit is used in the termination criterion for the algorithm. The PV threshold, which is used to identify the eligible solutions to be selected for the next population, is denoted by PV-MAX. Note that after some iterations, the PV of a pareto-optimal solution might change and not remain zero. In this case, the solution is taken out of the pareto optimal set. The output of the algorithm is the solutions with PV = 0. More explanation of each step will be given in the coming sections.

4.4.1 Population Initialization

Generating the first population of a GA plays an important role in approaching good solutions rapidly. The population must be rich enough to enable high-quality solutions. In order to set the first population, an initial pool is generated.

The current TA configuration $\mathbf{t}^0$, which is among the pareto-optimal solutions, is a natural starting point. To create diversity in the initial pool, the local search algorithm discussed in Section 3.3.1 is applied. Starting from $\mathbf{t}^0$, the local search algorithm iteratively updates the TA design. In every iteration, the algorithm considers cells that may be moved, and among these cells selects the cell move that results in the largest improvement in signaling overhead (cf. the local search algorithm in Chapter 3). This is repeated, without accounting for the reconfigu-

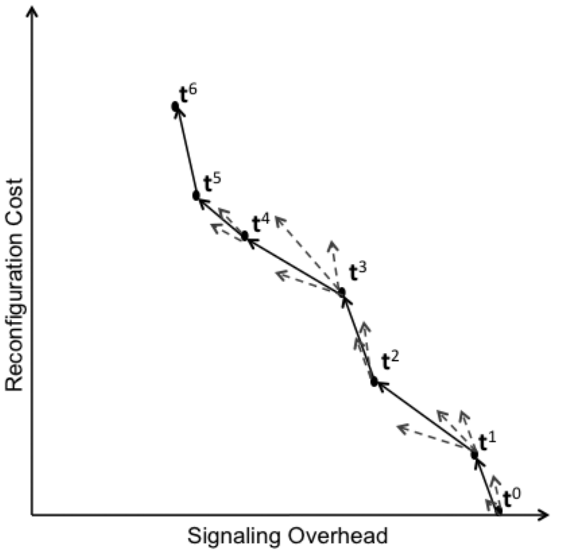


Figure 4.4 Applying local search to create the initial pool.

ration cost, until no further improvement can be obtained. In Section 3.3.1 the goal was to find the optimum reconfiguration with a budget limit, while here the goal is to keep all configurations encountered on the way to the lowest found signaling overhead. The initial pool consists of all the configuration points visited by the local search algorithm. Figure 4.4 illustrates the local search procedure in obtaining the configuration points. The dashed arrows represent the possible moves from $\mathbf{t}^n$ to $\mathbf{t}^{n+1}$, where n is the iteration counter. The solid arrows show the moves with largest improvement in the signaling overhead.

From Figure 4.4, it is observable that, while the local search starts from $\mathbf{t}^0$ and searches for configuration points with lower signaling overhead, the reconfiguration cost of those points are successively higher. The reason is that more cells change configuration compared to the initial design $\mathbf{t}^0$.

All points in the initial pool will be inside the first population. For generating the rest of the population, the GA algorithm randomly picks a configuration from the initial pool and perturbs the TA configuration of 20% of the cells. This is repeated until the population size reaches POPSIZE. To avoid poor configurations, during the perturbation, a cell can change TA, only if it is geographically located on the boundary of

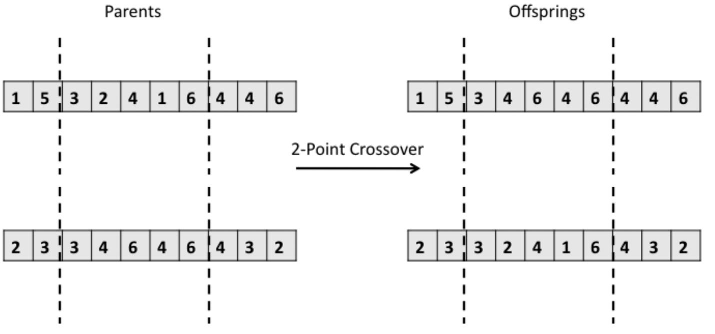


Figure 4.5 The 2-point crossover method in GA.

its TA. This is the case if the cell has at least one neighboring cell with positive handover and the neighboring cell is currently assigned to a different TA. In addition, the new TA is picked among the TAs of the neighboring cells.

4.4.2 Crossover

The role of the crossover operator is to inherit some characteristics of the two parents to generate the offsprings [61]. The PV values of the entire population are calculated. In the crossover operator, two parents are chosen randomly with the preference of having lower PV values. The elements are swapped between the randomly chosen two points to make two offsprings. Figure 4.5 explains the 2-point crossover method applied in this study. It is apparent from the figure that the cells in each offspring follow one of the parents' TA assignments, and therefore the output offsprings from the crossover operator are valid TA design solutions.

In the GA algorithm, the crossover operation is repeated until the number of offsprings is equal to POPSIZE. In order to avoid identical offsprings, first the algorithm makes sure that the chosen parents are different, and second the two crossover points are chosen with the condition that the two parents differ in at least one position between the two points.

4.4.3 Mutation

The mutation operator randomly modifies the elements of TA configuration vectors to promote diversity. A configuration is randomly chosen from the population with the preference of having low PV to enter the mutation operator. In the selected configuration, 5% of the elements are mutated. In the GA algorithm the mutation operation is repeated in POPSIZE times. Similar to the perturbation procedure described in Section 4.4.1, the mutation of a cell may take place only if the cell is on the boundary of its TA, and the TA of that cell can only be changed to a neighboring TA.

4.4.4 PV Local Search Algorithm

Usually by some simple modifications, the solutions obtained from GA can be improved. In this study, during each iteration of GA, a PV Local Search (LS) algorithm is used to further strengthen the GA algorithm. For each solution given to LS, the algorithm considers moving cells to other neighbor TAs one by one. Among these moves, the first move which results in a lower PV value is chosen, as long as the point defined by the signaling overhead and reconfiguration cost has not been visited yet. If the LS gets stuck in the situation where no move results in unvisited point with lower PV, the algorithm moves to an unvisited point with equal PV. The algorithm stops if all possible moves lead to visited points or higher PV values. All points visited by LS are stored and considered as visited in later runs of LS.

The goal of using LS in this stage of GA is to first find new solutions with lower PV to improve GA performance, and second to look for new pareto-optimal solutions. The next pool in the GA algorithm consists of solutions with PV $<$ PV-MAX after the LS.

It is possible to tune the number of points entering the LS by giving a value to PV-MAX. For example by setting PV-MAX = POPSIZE, all the points will be considered as an input to LS. PV-MAX is set to be lower than POPSIZE to save computing effort in case of large-scale networks.

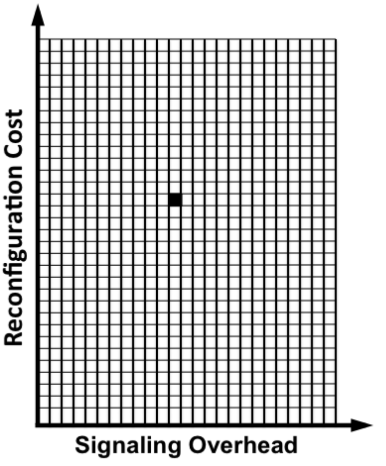


Figure 4.6 Quantization of the overhead and the reconfiguration cost.

4.5 Efficiency Improvement

There are two computational bottlenecks in the suggested GA algorithm: *First*, the PV of a solution is a relative value with respect to other solutions. Therefore, in order to calculate and update the PV of each solution, its signaling overhead and cost should be compared to all other solutions. *Second*, points that are visited should be stored in order to avoid being generated repeatedly. Ideally, one would like to record all the solution points found by the algorithm in all iterations. On the other hand, this becomes computationally unaffordable, since the number of accumulated solutions grows rapidly from one iteration to another. In this section, a method to resolve these bottlenecks is proposed by quantizing the two objective values.

The quantization of the bi-objective value space approximates the signaling overhead and the reconfiguration cost by a fixed and large number of intervals. With this process, the very large set of possible configuration points is approximated by a grid. Figure 4.6 gives an illustration of the quantization. Each pixel of the grid represents all TA configurations which give the signaling overhead and reconfiguration cost within the value intervals defining that pixel. To practically use this grid over the signaling overhead and cost axes, the grid is mapped to a matrix with the same dimensions as the grid size. Two matrices of same dimension are defined. Each of them will help solving one of the two

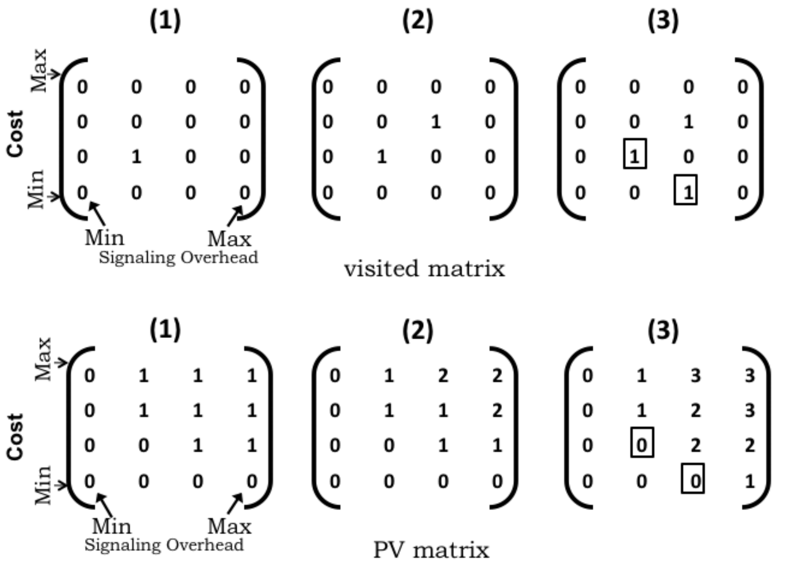


Figure 4.7 An example of the visited and PV matrices.

mentioned bottlenecks.

4.5.1 Visited Matrix

The visited matrix is defined to keep track on value intervals visited by the algorithm. It is a binary matrix to illustrate whether a grid element has been so far visited or not. If an element of this matrix is one, it means that the corresponding point has been already visited by a solution having signaling overhead and cost within that pixel, otherwise the value is zero. The upper part of Figure 4.7 shows a small example of how the visited matrix gets updated while new solutions are found by the algorithm. Moving from the first matrix to the third (left to right), in each step one solution is added to the visited matrix by changing one element from zero to one in each step.

4.5.2 PV Matrix

In order to find out the PV of a solution in constant time, a PV matrix is defined. This matrix has the same dimension as the visited matrix, and it is used to calculate the PV of each solution. Each new solution found by the algorithm is used to update the PV matrix by increasing all the dominated elements to the right and up of the corresponding pixel of the new solution by one. With this method of updating the PV matrix, it can be concluded that at any time, the value of each element in the PV matrix represents the number of found solutions which dominate the solution of the corresponding element. The lower part of Figure 4.7 illustrates a small example of how to update the PV matrix while adding a new solution. Note that the pareto-optimal solutions are the elements which are one in the visited matrix and zero in the PV matrix. Thus in the figure, the elements in the boxes represent pareto-optimal solutions.

4.6 Performance Evaluation

In this section, results of performance evaluation for realistic/real-life data of three large-scale networks are presented. In real-life networks, splitting a site into different TAs is not a common practice. Therefore, although all the discussions before considered cell-level TA assignment, the evaluation of the three networks is done on the site level, unlike in Chapter 3. In all scenarios, it is assumed that 5% of the UEs are paged in every site ($\alpha = 0.05$). The overhead of a single update c^u is set ten times as much as c^p [33].

For each of the first two networks, a reference scenario of UE distribution and mobility is defined. The scenario contains load and handover statistics of the network. The initial TA configuration, $\mathbf{t}^0$, is optimal for the reference scenario. Another UE scenario, called scenario I is generated by modifying the load and the handover statistics. It is considered that the reference scenario has evolved to scenario I over time. The aim is to find the pareto-optimal solutions of TA reconfiguration for scenario I. The third network is a real-life case, and $\mathbf{t}^0$ is the configuration used in the past few years. The suggested algorithm is applied to find the pareto-optimal solutions for reconfiguring $\mathbf{t}^0$ for the up-to-date UE distribution and mobility data.

The integer programming model defined in Section 4.2 has been implemented in the Gurobi optimizer [30]. The solver has been run on a processor with the clock speed of 2.4 GHz and 7 GB available RAM. For the first network, the integer programming model delivers all the exact pareto-optimal solutions. For the second network, some but not all of the exact pareto-optimal solutions can be calculated by the integer programming model. Due to the size of the third network and memory

Table 4.1 Minimum-overhead solutions found by the two approaches.

	$c_{SO}(\mathbf{t})$	$c_R(\mathbf{t})$	$\frac{c_{SO}(\mathbf{t}^0)-c_{SO}(\mathbf{t})}{c_{SO}(\mathbf{t}^0)}$
Integer Prog. Model	1.1140×10^5	1.4499×10^5	31.40%
GA Algorithm	1.1764×10^5	9.5504×10^4	27.84%

limitation, the integer programming model cannot be applied in this case.

The GA algorithm presented in Section 4.4 is implemented in MATLAB. The computations are run on a processor of type Intel Core 2 Duo with the clock speed of 2.1 GHz. For the three networks, the sizes of the visited matrix and PV matrix are chosen to deliver a sufficiently high resolution.

4.6.1 Network 1

The first set of data is from a cellular network of the downtown area of Lisbon, that is provided by the EU MOMENTUM project [46]. This network has been used in Chapter 3. The network consists of 60 sites and 164 cells. The optimum configuration for the reference scenario, $\mathbf{t}^0$, is computed by the model in [62]. There are seven TAs in $\mathbf{t}^0$. Figure 4.8 shows the pareto-optimal solutions found for the Lisbon network by the two approaches. The black dots represent the exact pareto-optimal solutions found by the integer programming model. There are 25 pareto-optimal solutions in Figure 4.8, however the model has been run more than 25 times in order to find these points. The model takes an average time of about 20 minutes to find a solution for a given cost budget (B). Therefore, for finding these 25 points by the integer programming model, about 8 hours and 20 minutes has been spent. The plus signs in Figure 4.8 illustrate the heuristic pareto-optimal solutions obtained by the GA algorithm with the following parameters: POPSIZE = 100, Iteration Limit = 10 and PV-MAX = 20. The PV matrix and the visited matrix are set to a size of 650-by-700, which gives a resolution of 255.18 and 249.83 units in overhead and cost, respectively. The GA computation took about 1 minute. The observations arrived from this figure are as follows.

- By successively allowing higher reconfiguration cost, there is a jump in the improvement of overhead. This shows the importance

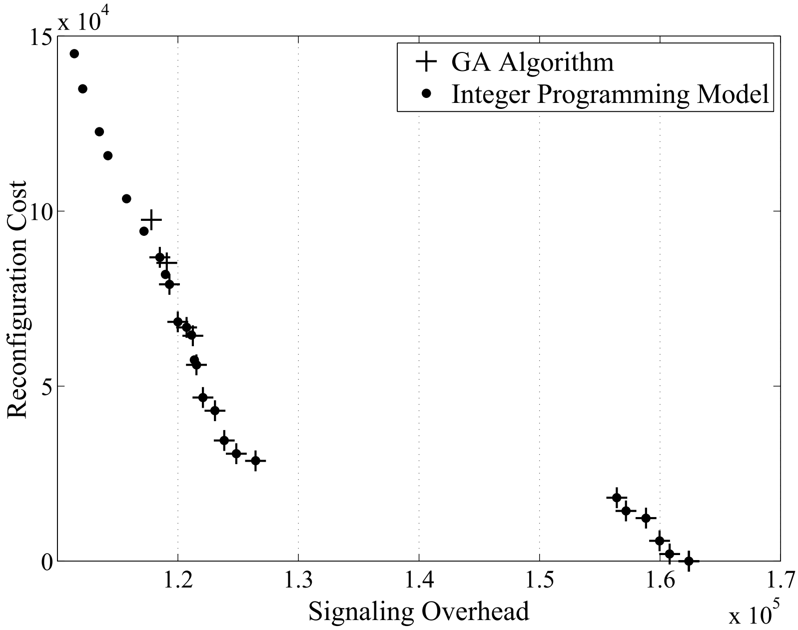


Figure 4.8 Pareto-optimal solutions of Network 1.

of approaching as many pareto-optimal solutions as possible.

- The performance of the GA algorithm is close to optimality. It did not approach the point with the minimum overhead and highest reconfiguration cost. However, the relative performance difference is small. Table 4.1 compares the minimum-overhead solutions found by each approach for the Lisbon network. The overhead improvement of the integer programming model is 5.60% over the GA algorithm. Note that for achieving this extra improvement, the reconfiguration cost will increase by 51.81%.

4.6.2 Network 2

The second data set represents a realistic deployment scenario for a network in one of the capital cities of Europe. The network consists of 75 sites and 225 cells. The optimum design for the reference scenario, $\mathbf{t}^0$, has twenty two TAs. Figure 4.9 shows the pareto-optimal solutions

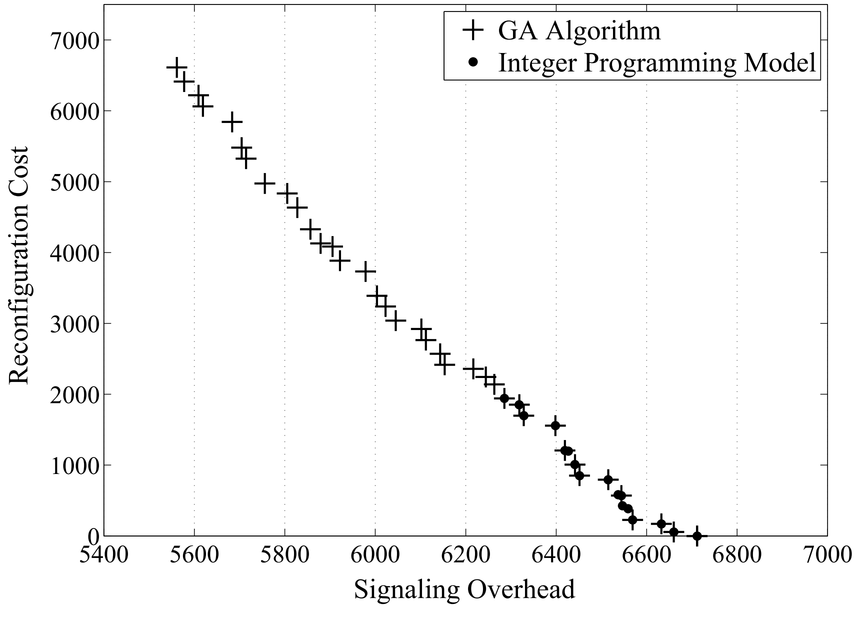


Figure 4.9 Pareto-optimal solutions of Network 2.

of this network found by the two approaches. The integer programming model found some but not all of the exact pareto-optimal solutions. It takes at least 1 hour to find each solution point. When B grows, the time for finding a solution increases rapidly to 8 hours. Therefore, searching for exact pareto-optimal solutions for $B \geq 2000$ is not computationally feasible. To get the heuristic pareto-optimal solutions from the GA algorithm in Figure 4.9, the following parameters are set: POPSIZE = 100, Iteration Limit = 10 and PV-MAX = 20. The PV matrix is set to a size of 700-by-700, which gives a resolution of 10.55 and 9.59 units in overhead and cost, respectively. The GA computation took about 10 minutes. Below are the remarks from the figure.

- The shape of the pareto frontier, which is the set of all pareto-optimal solutions of the signaling overhead and the reconfiguration cost, differs from that of network 1. The curve in Figure 4.9 is close to linear, meaning that for obtaining improvement in overhead, the reconfiguration cost scales up proportionally.

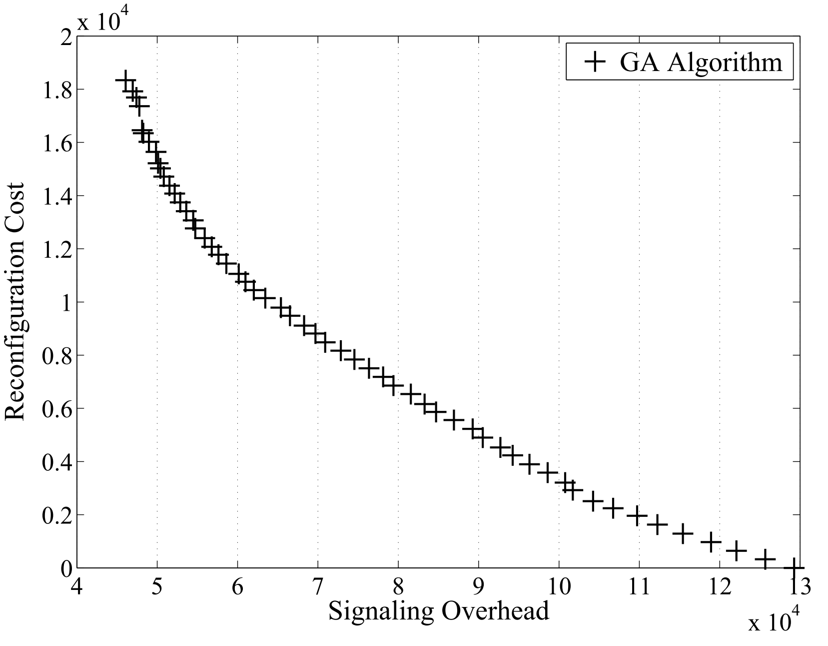


Figure 4.10 Pareto-optimal solutions of Network 3.

- The exact pareto-optimal solutions available from the integer programming model indicate that the pareto-optimal solutions found by the GA algorithm are of very high quality. The GA algorithm performs very well and time-efficiently for this network.

4.6.3 Network 3

The experiments for the third network use real-life data. The network is in use in a capital city of Asia. The network consists of 339 sites and 978 cells. The number of defined TAs in the current configuration is six. For Network 3, the computer memory needed by the solver exceeds what is available and therefore it is not possible to use the integer programming model. Figure 4.10 shows the pareto-optimal solutions of this network obtained by the GA algorithm. The PV and visited matrices are set to a size of 1200-by-1400, which gives a resolution of 102.96 and 109.20 units in overhead and cost, respectively. It took 2 hours and 20 minutes for the GA algorithm to find the pareto-optimal solutions in Figure 4.10

with the following parameters: POPSIZE = 300, Iteration Limit = 3 and PV-MAX = 20. After the third iteration, no new pareto-optimal solution was found. The observations from this figure are as follows.

- The smooth pareto frontier indicates that the decision-maker has a large set of available trade-offs between the signaling overhead and the reconfiguration cost.
- The current TA configuration of the network is far from optimum in terms of signaling overhead. The pareto-optimal solutions show that it is possible to decrease the overhead by 64%. Figure 4.11 shows the initial TA design $\mathbf{t}^0$ of Network 3. Figure 4.12 illustrates the same network, while the pareto-optimal solution with the lowest signaling overhead has been chosen. Each specific symbol in the two figures represents the sites inside one TA. In Figure 4.12, 111 sites have changed TAs in comparison to Figure 4.11. Therefore, about 32% of the sites in the network are reconfigured. There are some parts in both figures giving the impression that the TAs are disjoint. The reason is the existence of highways which make direct handovers possible between those parts of the city.

4.7 Conclusions

A bi-objective optimization model has been presented for pareto-optimal solutions for the trade-off between the signaling overhead and the TA reconfiguration cost. The proposed integer programming model provides the exact pareto-optimal solutions, and the suggested GA algorithm is simple in implementation and efficient in performance for large-scale networks. The experiments on several real-life networks demonstrate that the characteristic of the pareto frontier varies by network, and that the proposed GA algorithm provides close-to-optimal solutions for large-scale networks in feasible time.

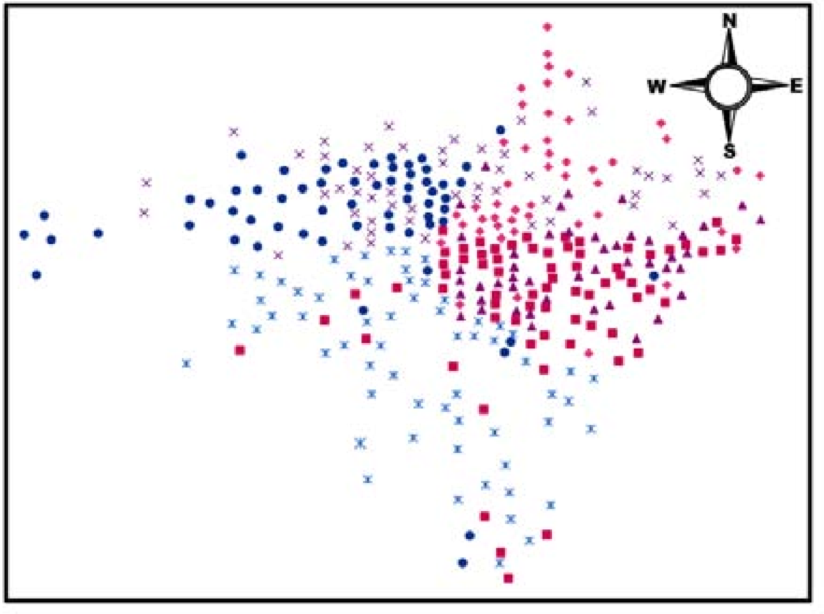


Figure 4.11 The initial TA design $\mathbf{t}^0$ of Network 3.

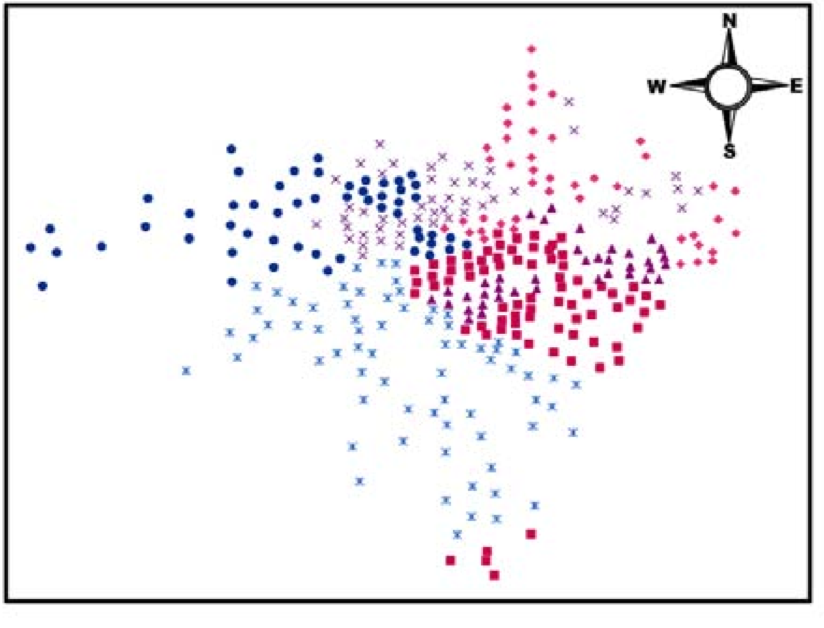


Figure 4.12 A pareto-optimal solution of Network 3.

Chapter 5

Tracking Area List

Tracking Area List (TAL) is a scheme introduced in 3GPP Release 8 [5]. This scheme allows more flexible TA configuration and is expected to overcome some of the limitations of the standard TA. Before investigating the TAL scheme, it is instructive to consider the limitations of the standard TA scheme, which has been used in the previous chapters. It has been already suggested in the literature that the TAL scheme can prevent the frequent updates when a UE keeps hopping between two or more adjacent cells in different TAs (the so called ping-pong effect). Second, TAL can solve the problem of high uplink traffic due to simultaneous updates of a large number of UEs crossing a TA boundary (the train scenario) [3, 40]. This chapter aims to shed light on the idea of TAL, which is explored further in the coming chapters. Some of the discussions presented in this chapter have been published in [43].

5.1 Limitations of Standard TA

In the standard TA scheme, cells/sites are grouped into mutually disjoint sets, each being a TA. A cell/site belongs to exactly one TA, and each UE is registered to only one TA. This scheme, which is used so far in the thesis, has some performance limitations.

5.1.1 Ping-Pong Effect

UEs at the border of neighboring TAs usually move back and forth between the two or three neighboring TAs (Figure 5.1). This phenomenon is referred to as the ping-pong-TAU effect. Apart from the mobility

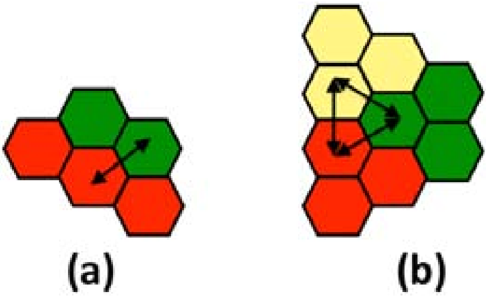


Figure 5.1 (a) ping-pong effect, (b) generalized ping-pong effect.

of the UEs, fading of the radio channel can also cause the ping-pong-TAU effect. The effect causes excessive TAUs and accounts for a non-negligible portion of the total TAU signaling overhead. In the standard TAU and paging scheme, no matter how the TAs are designed, the ping-pong effect exists either between two neighbor TAs, or sometimes between three neighbor TAs of a corner. The authors in [24] referred to such kind of ping-pong effect as the generalized ping-pong effect, see Figure 5.1(b). In general, reducing the ping-pong effect can significantly improve the performance of a network and therefore has received quite some attention in the literature [20, 24, 36]. However, most of the proposed schemes for reducing this effect introduce a large paging overhead.

5.1.2 Massive Mobility Signaling Congestion

If a large number of UEs simultaneously move into a hotspot cell (i.e., UEs in a train arriving quickly to a platform [40]), there is a risk of increased network load caused by excessive TAU from the UEs in a short period of time. Figure 5.2 shows the TAU storm, when a train passes a TA border. This is an undesirable situation from the network standpoint, as it could decrease the quality of service (QoS) in the cell and may create signaling resource congestion [38].

5.1.3 Symmetry Limitation

For the standard TA scheme, the following implication always exists. If cell A considers B in the same TA, then cell B also considers A in the same TA. If cells could have different perspectives towards each other, then this flexibility may lead to a lower signaling overhead.

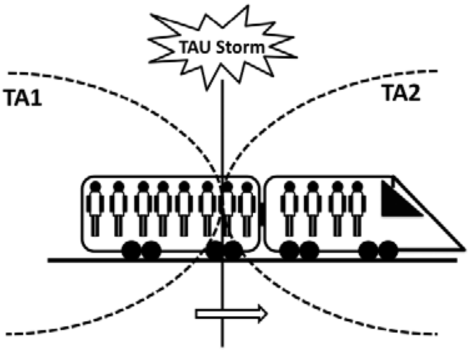


Figure 5.2 Example of TAU storm at the border of two TAs.

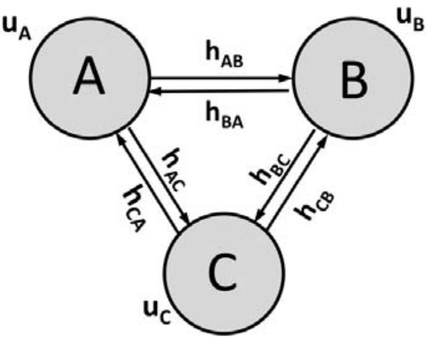


Figure 5.3 A three-cell network.

- Example: Figure 5.3 considers a network of three cells, $\mathcal{N} = \{A, B, C\}$. Ignore cell C for a moment and assume that the network consists of only two cells, A and B. The TA design either considers these two cells in the same TA, or separates them into two TAs. Based on (4.1), in the case that A and B are in the same TA, the signaling overhead is c_1 given in (5.1), and when A and B are in separate TAs the signaling overhead is c_2 given in (5.2).

$$c_1 = \alpha c^p (u_A + u_B) \tag{5.1}$$

$$c_2 = c^u (h_{AB} + h_{BA}) \tag{5.2}$$

Although it is not achievable by the standard TA scheme, let us consider a design in which all UEs in A percept that B is in another TA but all UEs in B assume A is in the same TA. In this situation the signaling overhead is c_3 given in (5.3).

$$c_3 = c^u h_{AB} + \alpha c^p u_B \tag{5.3}$$

In conditions where $c_1 - c_3 > 0$ and $c_2 - c_3 > 0$ hold, corresponding to $\alpha c^p u_A > c^u h_{AB}$ and $c^u h_{BA} > \alpha c^p u_B$, the third perception (5.3) leads to lower signaling overhead. Thus when cell A has high cell load, and there is no or few flow moving from cell A to cell B, and B has low cell load but high number of moves towards A, there is a potential to reduce the signaling overhead if the cells can break the symmetry in their view of TA.

Another implication by the standard TA scheme is transitivity: If cells A and B are in a same TA, and cells B and C are in a same TA, then cells A and C must also be in the same TA.

- Example: Consider Figure 5.3 and two TA configurations for this network. The first configuration puts all three cells in one TA, which results in matrix $\mathbf{S}_1(\mathbf{t})$ given in (5.4), and in the second one A and B are in the same TA, while C forms its own TA. The second configuration results in matrix $\mathbf{S}_2(\mathbf{t})$. The corresponding signaling overheads, c_4 and c_5 are given in (5.6) and (5.7), respectively.

$$\mathbf{S}_1(\mathbf{t}) = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{5.4}$$

$$\mathbf{S}_2(\mathbf{t}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{5.5}$$

$$c_4 = 2\alpha c^p (u_A + u_B + u_C) \tag{5.6}$$

$$c_5 = \alpha c^p (u_A + u_B) + c^u (h_{AC} + h_{CA} + h_{BC} + h_{CB}) \tag{5.7}$$

Let us assume that a design represented by matrix $\mathbf{S}_3(\mathbf{t})$ can be created disregarding the fact that it is not achievable using the standard TA scheme. The signaling overhead of matrix $\mathbf{S}_3(\mathbf{t})$ is equal to c_6 in (5.9).

$$\mathbf{S}_3(\mathbf{t}) = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \tag{5.8}$$

$$c_6 = \alpha c^p(u_A + 2u_B + u_C) + c^u(h_{AC} + h_{CA}) \tag{5.9}$$

In situations where $c_4 - c_6 > 0$ and $c_5 - c_6 > 0$, meaning that $\alpha c^p(u_A+u_C) > c^u(h_{AC}+h_{CA})$ and $\alpha c^p(u_B+u_C) < c^u(h_{BC}+h_{CB})$, designs (5.4) and (5.5) have higher signaling overhead compared to design (5.8). This example illustrates that there is a potential of reducing the signaling overhead if the transitivity condition can be relaxed.

Generally, in the standard TA scheme, there are three properties of any $\mathbf{S}(\mathbf{t})$ matrix:

- $\mathbf{S}(\mathbf{t})$ is a binary matrix, where $s_{ij}(\mathbf{t})$ represents whether or not two cells are in the same TA.
- $\mathbf{S}(\mathbf{t})$ is a symmetric matrix, representing the obvious fact that if cell i and j are in a same TA, then cell j and i are also in the same TA.
- $\mathbf{S}(\mathbf{t})$ has the transitive property, meaning that whenever cells i and j are in the same TA, and cells j and k are in the same TA, then i and k are also in the same TA.

5.2 Tracking Area List

There were extensive discussions in 3GPP on the preferred TA scheme. The standard TA scheme, which consists of static non-overlapping TAs, was used in earlier technologies, such as GSM. However, there are newer schemes that have the potential of avoiding ping-pong effects, distributing the TAU load more evenly across cells and reducing the overall TAU signaling overhead [47]. Some of the candidate schemes that were discussed are as follows:

- *Overlapping TA*: In this scheme one cell holds a list of overlapping TAs and a UE is assigned only to one TA of the cell's list. The UE does not perform a TAU while moving to a cell which has the assigned TA in its list.
- *Multiple TAs*: In this scheme, a cell belongs to only one TA, but a UE can be assigned with more than one TA using a list. If one UE is assigned a list of TAs, the UE does not perform TAUs when it crosses the boundaries between these TAs. The TAs in this scheme are non-overlapping.

Introducing the concept of list gives more flexibility to the operators in their TA management. It should be mentioned that the above schemes are considered to specifically reduce the signaling overhead resulted from TAUs, while the problem of paging overhead is considered as a much less critical issue. The concept of Multiple TAs is currently the more preferred scheme among the two [49].

In the TAL scheme discussed in the thesis, both UEs and cells are assigned to a list of non-overlapping TAs. A UE receives a TAL from a cell, and keeps the list, until it moves to a cell that is not included in any TA of the list. The UE location is known to the MME to at least the accuracy of the TAL allocated to that UE.

- Example: In Figure 5.4, TAL1 consists of TA1, TA2, TA3 and TA4, and TAL2 consists of TA2, TA7 and TA9. By assuming that the network gives TAL1 to UE2 and TAL2 to UE1, UE1 will not have any TAU while moving from TA9 to TA7. UE2 will only make TAU when passing from TA2 to TA7, because UE2 does not have TA7 in its TAL. Note that if any of these UEs were paged, the paging message will be sent to all the cells inside the UE's TAL. Therefore, in TAL design giving a proper TAL to each individual UE can be very beneficial.

Potentially, the TAL scheme is expected to overcome some of the limitations of the standard TA scheme. For example, TAL can avoid the ping-pong TAU by including the last visited TA in the assigned TAL. To overcome the signaling congestion in the train scenario, the

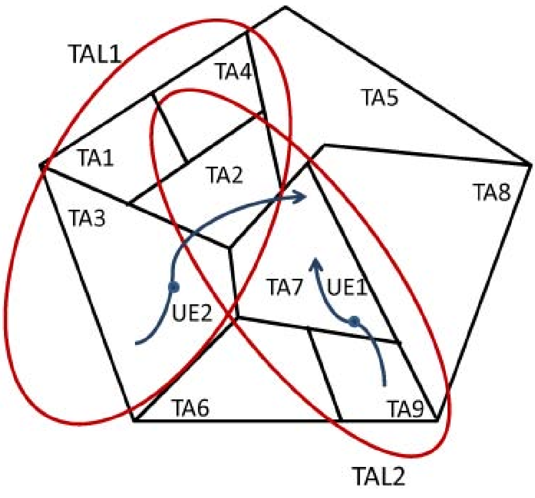


Figure 5.4 An example of TAL.

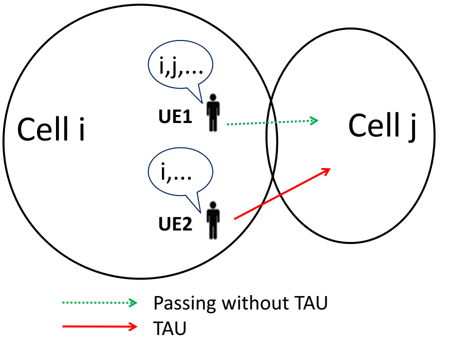


Figure 5.5 UEs holding different TALs in one cell.

cells along the railway path can give different TALs to the UEs inside the train. Because the UEs inside the train are holding different TALs, not all of them will perform TAU at the same time.

5.3 Challenges in Applying TAL

By TAL, the UEs in one cell might have different TALs, depending on the cells from which the TALs are assigned. This perspective difference makes the estimation of signaling overhead difficult. Figure 5.5 illustrates two UEs in cell i moving towards cell j. The UEs are holding different TALs which affect the TAU overhead calculation. If cell j is included in the TAL of cell i, then no TAU overhead is needed for UEs having i's list and moving to cell j. This is the case for UE1 in the figure. UE2, on the other hand, does TAU because it does not have j in its list. Similarly, UEs having TALs of other cells, such that the TALs have i but not j will generate TAU when moving from i to j.

From the above discussion, it can be concluded that in designing TAL for a network, it is highly valuable to have accurate data traces of UEs' movements. Unfortunately, this data is not available or too expensive to obtain. Also, designing TAL according to traces of UE mobility patterns will limit the solutions to those specific movement sequences. If the UEs change their movement behavior, which is quite probable, the TALs would become inefficient.

Chapter 6

Applying TAL in Cellular Networks

Although TAL is expected to overcome some of the limitations of the standard TA scheme, how to apply TAL in large-scale networks remains unexplored. This chapter proposes a signaling overhead calculation formula, taking the discussion in Section 5.3 into account. Three algorithms are suggested to utilize TAL in large-scale networks. The advantages and disadvantages of each of these schemes are explained. This chapter is based on the work presented in [43, 44].

If the information of each individual UE's movement and calls were available to the network, then designing an optimum TAL would become trivial and could essentially result in the elimination of signaling overhead: The network could give a specific, tailored list to each UE including all the cells the UE intends to pass before it will be called. This information, if available at all, is costly to obtain. Moreover, the validity of the information expires fast, because the UE trace is the history of the UE's movement, and the UE's intention of where and when to move in future is unknown.

6.1 Signaling Overhead Calculation for TAL

To explore the flexibility of TAL, one can consider a TA as small as a single cell (i.e., no restriction on a given TA layout). LTE allows a cell to assign UEs different TALs. Using this possibility is however out of the scope of the thesis. Here, the assumption is that a cell will give only one common TAL to all UEs getting updated in that cell. Inevitably, this will imposes restriction on the performance of TAL.

Note that for the sake of reusing Equation (2.2), the notations $s_{ij}(\mathbf{t})$ and $\mathbf{S}(\mathbf{t})$ are kept. However, for TAL the vector $\mathbf{t}$ does not exist and the TAL assignments can be defined in the form of an $N \times N$ matrix. For TAL, $s_{ij}(\mathbf{t})$ is defined in the thesis as the proportion of UEs in cell i, who have j in their TAL. Thus, the $\mathbf{S}(\mathbf{t})$ matrix would contain fractional values which are all between 0 and 1. Although the matrix is not binary any more, Equation (2.2) remains valid. Given a TAL-cell assignment, there can be several ways to estimate $s_{ij}(\mathbf{t})$. In this section, first a one-hop calculation is considered, and then the idea is extended to two hops.

6.1.1 One-hop Calculation

The uncertainty factor in calculating $s_{ij}(\mathbf{t})$ stems from the fact that UEs in cell i may hold TALs of different cells, in particular those other than cell i. To estimate $s_{ij}(\mathbf{t})$ in the one-hop calculation, the impact of neighbor cells of i are considered. The formula is shown in (6.1). The denominator shows an estimation of the overall number of UEs in i. The second term in the denominator calculates the number of UEs moving to i without having updated by cell i, while u_i estimates the number of UEs in cell i having the TAL of i. The numerator estimates the number of UEs which are probable to have j in their TALs. Parameter l_{ij} is defined 1 if j is in the TAL of i and 0 otherwise. The neighbors of i having i in their TAL form the Q_i set. Factor γ represents the probability of UEs entering cell i and having been updated in Q_i.

- Example: In Figure 6.1(a) only the impact of neighbor cells of i are considered, and the orange cells represent the neighbors containing i in their TAL. If $l_{n_1 i} = l_{n_2 i} = 1$, then $Q_i = \{n_1, n_2\}$.

$$s_{ij}(\mathbf{t}) = \frac{u_i l_{ij} + \gamma \sum_{n \in Q_i} h_{ni} l_{nj}}{u_i + \gamma \sum_{n \in Q_i} h_{ni}} \tag{6.1}$$

The standard TA scheme can be used as a baseline for validating (6.1). For all $n \in Q_i$, $l_{ni} = 1$, because Q_i is the set of neighbors of i having i in their TAL. Also, for the standard TA scheme all the cells

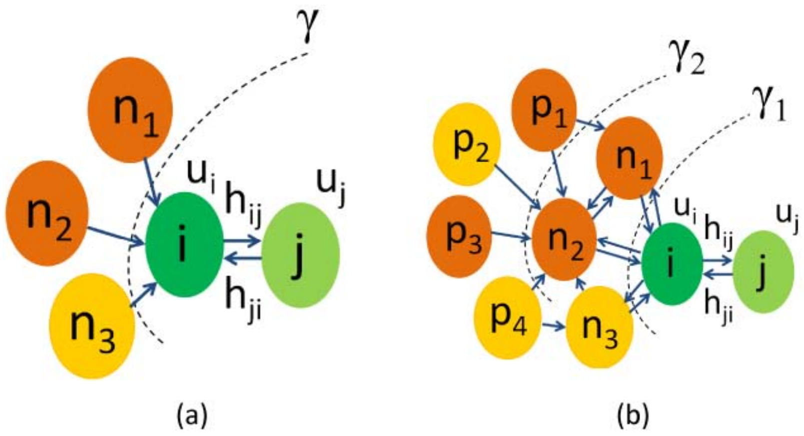


Figure 6.1 Parts of a network involved in estimating $s_{ij}(\mathbf{t})$.

inside one TA are assumed to have the same TAL. If $l_{ij} = 1$ then by the transitive relation in the standard TA scheme $l_{nj} = 1$ for all $n \in Q_i$, and hence $s_{ij}(\mathbf{t}) = 1$. If $l_{ij} = 0$ then again due to transitivity $l_{nj} = 0$ for all $n \in Q_i$, and hence $s_{ij}(\mathbf{t}) = 0$. This gives the logical conclusion that the parameter γ does not play any role in the $s_{ij}(\mathbf{t})$ calculation of the standard TA scheme and $s_{ij}(\mathbf{t}) = l_{ij}$, for all $i, j \in \mathcal{N}$.

From the equation, it is observable that when the TAL of each cell contains that cell, the values on the diagonal of $\mathbf{S}(\mathbf{t})$ is always equal to one.

6.1.2 Two-hops Calculation

To extend the calculation to two hops, the impact of the neighbors of neighbors should be also included in the calculation. This may increase the accuracy of $\mathbf{S}(\mathbf{t})$ estimation. In Equation (6.2) two hops are considered. The denominator is showing an estimation of the overall number of UEs in i. The overall number of UEs in cell i is estimated by the sum of cell load of i and the UEs entering i with i in their TAL. The numerator of (6.2) is giving the number of UEs in i estimated to have j in their TALs. Parameter l_{ij} is 1 if j is in the TAL of i and 0 otherwise.

$$s_{ij}(\mathbf{t}) = \frac{u_i l_{ij} + \gamma_1 \sum_{n\in Q_i} h_{ni} l_{nj} + \gamma_2 \sum_{n\in Q_i} \sum_{p\in Q_{ni}} min(h_{pn}, h_{ni}) l_{pj}}{u_i + \gamma_1 \sum_{n\in Q_i} h_{ni} + \gamma_2 \sum_{n\in Q_i} \sum_{p\in Q_{ni}} min(h_{pn}, h_{ni})} \tag{6.2}$$

The neighbors of i having i in their TAL form the Q_i set. Notation Q_{ni} is the set of neighbors of $n \in Q_i$ having both n and i in their TALs. In most of the cases, $i \in Q_{ni}$, because as long as the neighborhood definition is defined in both directions, i is considered as the neighbor of neighbor of i. While UEs move from the cells in Q_{ni} to n and thereafter to i, there will be no TAU.

- Example: In Figure 6.1(b) the impact of neighbors of neighbors is considered. The TAL of orange cells contain i and the connected cells from the first-hop neighbors containing i. By assuming $l_{n_1 i} = l_{i n_1} = l_{n_2 i} = l_{p_1 n_1} = l_{p_1 i} = l_{p_1 n_2} = l_{p_3 n_2} = l_{p_3 i} = l_{n_1 n_2} = 1$ in the figure, one concludes $Q_i = \{n_1, n_2\}$, $Q_{n_1 i} = \{i, p_1\}$, and $Q_{n_2 i} = \{n_1, p_1, p_3\}$.

Factor γ_1 represents the probability of UEs entering cell i having been updated in Q_i. Similarly, γ_2 is the fraction of UEs entering cell i and holding a TAL assigned by some cell in $Q_{n_k i}$ with $n_k \in Q_i$. The reason for picking the minimum value between h_{pn} and h_{ni} in the last term is to avoid overestimating the effects of UEs entering i.

It can be observed from definition (6.2), that:

$$\begin{aligned} \gamma_1 + \gamma_2 &\leq 1 \\ \gamma_2 &\leq \gamma_1 \end{aligned} \tag{6.3}$$

The constraints in (6.3) conclude that $\gamma_2 \leq 0.5$. It should be also considered that not all combinations would be practically reasonable. As an example $\gamma_1 = \gamma_2 = 0.5$ is not a valid assumption, because it represents that for each cell the impact of the second-hop cells (not including the cell itself) are as much as the first-hop cells.

6.2 How to Design TAL?

In this section, three algorithms are suggested for designing TAL using the available data from a cellular network. All the algorithms are designed based on the objective of improving the overall signaling overhead.

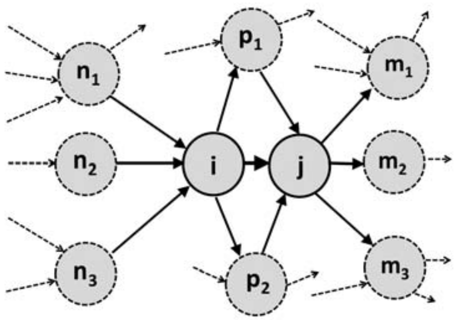


Figure 6.2 An example of the dependency between elements of $\mathbf{S(t)}$.

6.2.1 TAL Design Independent from UE Traces

For allocating and assigning TAL independent from UE traces, the only available information is the load of each cell and handover between cell pairs. In Section 6.1, simple formulas are defined for estimating all $s_{ij}(\mathbf{t})$ values of the $\mathbf{S(t)}$ matrix, considering one or two hops. The definition of the l_{ij} parameter in the formulas requires a TAL assignment. Thus, the problem is narrowed down to the challenge of finding a TAL assignment resulting in an estimated $\mathbf{S(t)}$ matrix which improves the overall signaling overhead calculated by Equation (2.2).

Because UEs in one cell can hold different TALs, it is clearly not possible to set each element of $\mathbf{S(t)}$ completely independently from the other elements. Also, it is already mentioned in Section 6.1 that the $s_{ij}(\mathbf{t})$ values in TAL can be fractional.

- Example: From Figure 6.2 it can be observed that by adding or omitting j from the TAL of i, not only s_{ij}, but s_{p_1j}, s_{p_2j}, s_{jm_1}, s_{jm_2} and s_{jm_3} may all change in the $\mathbf{S(t)}$ matrix.

The following local search algorithm is developed to make the TAL assignment of each cell considering the effects on the other cells.

Local Search Algorithm

The local search algorithm iteratively updates the TAL assignment of each cell. The basic operation of the algorithm is to modify the TAL of one cell at a time, by either deleting or adding one of the elements in the TAL. Then, depending on how many hops are considered in the

Algorithm 3 Local Search for TAL Allocation.

$tal^0 = t^*$
$tal^* = tal^0$; $c^*_{SO} = c_{SO}(tal^*)$
repeat
$\delta = 0$, $c^p_{SO} = c^*_{SO}$
for all $i \in \mathcal{N}$ **do**
for all $j \in \mathcal{N}$ **do**
$tal^l = tal^*$, $A^{tal^l} = A^{tal^*}$
if $j \in tal^l_i$ **then**
$tal^l_i = tal^l_i \setminus \{j\}$, Update $A^{tal^l}_i$ and s_{ij}
for all $p \in tal^l_i$ **do**
if $p \in tal^l_j$ **then**
Update s_{jp}
end if
if $j \in tal^l_p$ **then**
Update s_{pj}
end if
end for
if $c_{SO}(tal^l) < c^*_{SO}$ **then**
$tal^* = tal^l$, $c^*_{SO} = c_{SO}(tal^*)$, $A^{tal^*} = A^{tal^l}$
else
$tal^l = tal^*$, $A^{tal^l} = A^{tal^*}$
end if
end if
if $j \in A^{tal}_i$ **then**
$tal^l_i \leftarrow \{tal^l_i, j\}$, Update $A^{tal^l}_i$ and s_{ij}
for all $p \in tal^l_i$ **do**
if $p \in tal^l_j$ **then**
Update s_{jp}
end if
if $j \in tal^l_p$ **then**
Update s_{pj}
end if
end for
if $c_{SO}(tal^l) < c^*_{SO}$ **then**
$tal^* = tal^l$, $c^*_{SO} = c_{SO}(tal^*)$, $A^{tal^*} = A^{tal^l}$
else
$tal^l = tal^*$, $A^{tal^l} = A^{tal^*}$
end if
end if
end for
end for
$\delta = c^p_{SO} - c^*_{SO}$
until $\delta^* = 0$
return tal^l

algorithm, Equation (6.1) or (6.2) is used to estimate the $\mathbf{S(t)}$ matrix which in turn gives the overall signaling overhead. This is repeated until no additional change results in any improvement.

The local search algorithm is formalized in Algorithm 3, in which the solution given to and returned by the algorithm is denoted by tal^l, and the algorithm considers one-hop calculation. The optimal standard TA configuration can be used as the starting TAL assignment of the network. This means that the cells which belong to the same TA are given a list containing all the cells in the TA. Of course, with this solution, the $\mathbf{S(t)}$ matrix is binary, and by definition it is equal to the matrix obtained from the condition stated in (2.1).

At line 8, the algorithm checks whether j should be in the TAL of i or not. If j belongs to the list of i, the algorithm removes this cell at line 9, and if it does not belong to the list of i but it is adjacent to the TAL of i, the algorithm adds it to the list of i at line 25. All the neighbor cells to the TAL of i are stored in A_i^{tal}. When j is added or removed from the TAL of i, row A_i^{tal} of the matrix A^{tal} is updated by the algorithm.

According to the discussion in Section 6.2.1 and Figure 6.2, it can be concluded that by adding/removing only one cell j to/from the TAL of another cell i, there are three parts in the $\mathbf{S(t)}$ matrix which should be updated, in case of one-hop calculation:

- s_{ij}
- s_{pj} for all $p \in tal_i \cap tal_j$
- s_{jp} for all $p \in tal_i \cap tal_p$

The algorithm changes all these elements of $\mathbf{S(t)}$ and calculates the signaling overhead. If the change results in a lower signaling overhead, then the change is kept, and if it results in a higher signaling overhead, the previous configuration is again active. By repeating this procedure for every (i, j) pair, a modified TAL will be defined for each cell. The algorithm iterates until no more improvement is possible. Note that the algorithm is not achieving the optimum TAL design, but rather aiming at an improved configuration which should result in a lower overall signaling overhead compared to the standard TA scheme.

Advantages

- The scheme does not require any information regarding the UE traces. The same input data for designing a standard TA scheme can also be used here.

- The scheme considers the impact of adding or omitting a cell from the list of other cells on either one-hop or two-hops cells, therefore the calculation tends to take care of the challenge explained in Section 5.3.

Disadvantages

- The algorithm is based on the $s_{ij}(\mathbf{t})$ formulas suggested in (6.1) and (6.2). Each cell has its own true value of γ. However, in the formulation the average estimation of this value over the network is considered. It's hard to approach a good average estimation of γ, which influences the TAL design and the resulting signaling overhead. In the two-hops formula, a good estimation of the ratio between γ_1 and γ_2 is also important.

- Applying the algorithm considering two-hops formulation has a rather high complexity of calculations.

- The starting point in the algorithm has an impact on the final result. A logical available starting point is to use the optimum standard TA configuration. This requires to obtain the optimum standard TA design before starting the algorithm.

6.2.2 An Intuitive Rule of Thumb

The flexibility of TAL comparing to the standard TA scheme is that $\mathbf{S}(\mathbf{t})$ is not necessarily binary nor symmetric. If one considers a network with only two cells (i.e., cells i and j in Figure 6.1 and ignoring the rest). There are two choices for designing the TAL of cell i:

1. $tal_i = \{i\}$
   In this case, the signaling overhead resulted from cell i is $c^u h_{ij}$. This means that all the flow moving from i to j should have a TAU, because j is not included in the TAL of i.

2. $tal_i = \{i, j\}$
   Here, the signaling overhead resulted from cell i is $\alpha c^p u_i$, meaning that if a UE is paged in i, there will be paging in cell j, but there is no TAU for the UEs moving from i to j.

Thus, for minimizing the signaling overhead resulted from cell i, the following decision can be made:

$$l_{ij} = \begin{cases} 1 & \text{if } \alpha c^p u_i < c^u h_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

If $\alpha c^p u_i < c^u h_{ij}$, then it is desirable for i to include j in its TAL. The same logic can be applied for designing the TAL of cell j. The principle is easily generalized to the whole network.

Advantages

- The algorithm is simple and easy to be applied in a large-scale network. It usually gives a good TAL assignment. That is why it is called a *rule of thumb* for designing TAL.

- The scheme does not require any information more than the load and handover of the cells. It does not even depend on the standard TA configuration or tuning of any parameter.

- The algorithm has very low computational complexity.

Disadvantages

- The main disadvantage of the algorithm is that it only searches among the one-hop neighbors of a cell to be included in the TAL of that cell. Thus, the maximum length of a cell's TAL is the number of neighbors of the cell plus one. This limitation becomes critical if reducing the TAU overhead is prioritized over the paging overhead (i.e. $c^u \gg c^p$).

- In this algorithm, each cell is "selfishly" optimizing the signaling overhead according to its own data and does not consider the joint effect of the other cells' TALs.

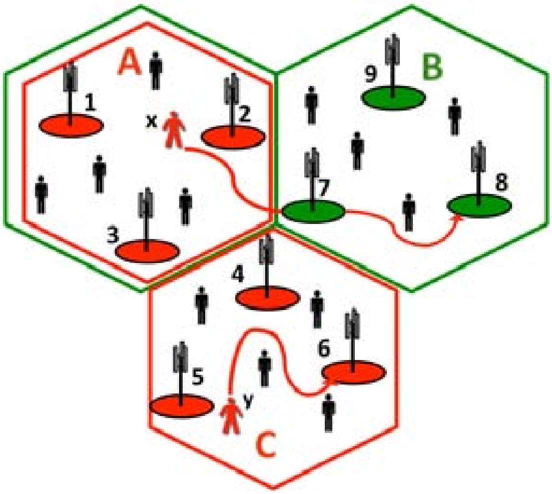


Figure 6.3 An example of how to collect part of UE traces.

6.2.3 TAL Design based on UE Traces

In the previous two sections, TAL design has been based on the load and handover of each cell for a time duration. If possible, it would be desirable to utilize UE traces in designing TAL. With the earlier cellular technologies, getting the traces of idle UEs was an extremely costly and unfeasible procedure. However in LTE, there are some possibilities of collecting a small part of UE traces. One is the existence of UEs using GPS-featured applications in the network. Another possibility is to apply the following TAL scheme adjusted inside the network.

Collecting UE traces

In the TAL concept, a cell is able to give different TALs to different UEs, and with this feature it is feasible to collect a few percentage of UE traces. If all cells in a network update a specific UE by giving it a list which only contains the updating cell, then the network can trace that specific UE. The idea is clarified by the following example.

- Example: All the UEs in Figure 6.3 are assumed to be idle UEs. UE x and UE y are marked red, which indicates that the traces of these two UEs will be collected. There are three TAs in this figure (A, B and C). Let's consider that TAL1 $= \{A, C\}$ and TAL2

$= \{A, B\}$. When each black UE enters a TA which is not included in its TAL, the UE updates to either TAL1 or TAL2. For the red UEs, every cell gives TAL of a single cell. Unlike other UEs, UE x and UE y update when they pass any cell, but not to TAL1 or TAL2. The TALs of UE x and UE y always consist of one cell, which is the current serving cell. Therefore, by all the updates, the trace 2-7-8 can be collected for UE x and the trace 5-4-6 can be collected for UE y.

The price of collecting a portion of UE traces is the amount of updates that the selected UEs create. Note that for these selected UEs, there is no paging overhead.

Optimization Algorithm

In this algorithm, unlike the other two, the load and handover of the cells are not used. Instead, a portion of UE traces is considered as the input data. The algorithm goes through these UE traces and considers whether adding or omitting a cell from the TAL of another cell will potentially improve the overall signaling overhead or not. The algorithm uses the same basic TAL-modification operation as in Algorithm 3. The difference is that the collected UE traces are used in signaling overhead estimation. The result is the optimized TA assignment for the available UE traces, and the solution can be applied to the entire network. Of course, by having 100% of the UE traces, a close-to-optimum TAL assignment for the network could be found.

Advantage

- The scheme explores the tracing possibility admitted by the flexibility of TAL.

Disadvantage

- The network has to trace some UEs at the accuracy of the cell level. This may cause too much signaling if the sampling percentage is large.

Chapter 7

Performance Evaluation of TAL Schemes

This chapter focuses on a comparative performance evaluation of the standard TA scheme and the three TAL schemes presented in Chapter 6. Here, the natural questions are: What is the potential of TAL in comparison to the standard TA scheme? Among all the TAL schemes, which one results in lowest signaling overhead, and which one is more practical to be applied in a large-scale network? A carefully designed evaluation framework is presented to answer these questions.

7.1 Generating UE-traces Scenario

The performance of a TAL scheme is always dependent on the UE traces inside the network. In order to have a fair comparison between the standard TA and TAL schemes, one proper way is to apply all the schemes to the same UE-traces scenario. A UE-traces scenario is a set of UE traces and call arrivals for a specific time period in a network, matching the cell load and handover. For one set of cell load and handover data, there can be uncountable numbers of matching UE-traces scenarios. UE-traces scenario is dependent on the chosen mobility model including the amount and speed of each UE's movement.

Assuming that cell load, handover and the call intensity factor (α) are the available data of the UEs' mobility behaviors of a network, to generate a proper UE-traces scenario, the following aspects should be considered.

- The UE is more probable to exist in cells with higher cell load. Equation (7.1), where u_i is the load of cell i, estimates the probability ratio of cell i being the starting cell of a UE in the specified time period.

$$P_{start}(i) = \frac{u_i}{\sum_{j \in \mathcal{N}} u_j} \tag{7.1}$$

- The UE tends to move to neighbors with high handover value. Equation (7.2) gives the probability of a UE moving from cell i to cell j, provided that the UE leaves cell i. The handover between i and j is defined by h_{ij} and the set of neighbors of cell i is denoted by A_i.

$$P_{move}(i, j) = \frac{h_{ij}}{\sum_{p \in A_i} h_{ip}} \tag{7.2}$$

- Among the UEs inside the UE-traces scenario, α of them are paged in the specified time period.

In order to store a UE-traces scenario, a scenario matrix is defined. The number of UEs in the UE-traces scenario is denoted by V and the time duration under investigation is denoted by T. Each row of the matrix represents one UE's movement during different time intervals of T. The serving cell of UE v in time interval τ is stored as the element at (v, τ) of the scenario matrix. The length of each time interval is denoted by $\Delta\tau$, and the dimension of the scenario matrix is defined by $V \times \frac{T}{\Delta\tau}$. If the UE remains in the same cell for some time, the corresponding elements are identical in the intervals.

- Example: Figure 7.1 illustrates an example of a row in the matrix. The time period T is divided into ten equal time intervals $(\tau_1, ..., \tau_{10})$. The UE stays in cell i for $3\Delta\tau$ and then it moves to cell j and stays there for $5\Delta\tau$. At τ_8 the UE makes another move to cell k and stays there for the rest of the time period.

The first column of the matrix is the starting cells of all UEs randomly generated based on Equation (7.1). The cell-to-cell movements

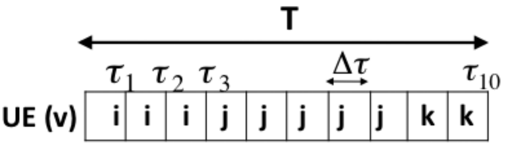


Figure 7.1 An example of a row in the scenario matrix.

of UEs are randomly generated according to (7.2). Based on the call intensity factor, $\lfloor \alpha V \rfloor$ UEs are randomly chosen to be paged in some intervals of T. The call duration for each paging is randomly chosen based on the rayleigh distribution. During a call, the UE is active and the network knows the UE's location at cell level. Thus, there is no update overhead during a call.

7.2 Aggregating Data from UE-traces Scenario

Because the UE-traces scenario is generated based on the probabilities given by cell load and handover, the implied cell load and handover of the UE-traces scenario is not exactly the same as the original data. Thus, it is required to aggregate the cell load and handover from the UE-traces scenario.

The cell load u_i is defined as the total number of UEs in cell i scaled by the time proportions that the UEs spend in cell i. Therefore, the load of each cell in the network is aggregated by the scaled values of UEs staying in the cell using all the elements of the scenario matrix. The aggregated handover value is the number of moves from one cell to another.

- Example: Considering the example in Figure 7.1 once more, the aggregated cell load and handover from UE v are: $u_i = 0.3$, $u_j = 0.5$, $u_k = 0.2$, and $h_{ij} = 1$, $h_{jk} = 1$ (where $\frac{\Delta\tau}{T} = 0.1$).

7.3 Calculating the Signaling Overhead

According to the previous section, it is possible to simulate a UE-traces scenario. The standard TA and TAL schemes can be applied to the same UE-traces scenario and their performance can be compared to each other. For most of the schemes, however, the aggregated data in load and handover are the input. There are two methods for calculating the signaling overhead resulted from a TA/TAL scheme for the UE-traces scenario. Method I, which gives the accurate overall signaling overhead, directly counts the total numbers of TAUs and pagings in the UE-traces scenario. Method II, which is merely a metric used for the estimation of signaling overhead in some of the schemes, calculates the true $\mathbf{S}(\mathbf{t})$ matrix based on the UE-traces scenario and then uses Equation (2.2).

Method I: Simulating UE-Traces Scenario

Considering that each cell has received the TAL of the starting cell, the TAL of each UE is known in all the time intervals by following the UE trace. Each time a UE moves from a cell which is not included in the UE's TAL, the TAU cost c^u is added to the overall signaling overhead. Depending on the length of the holding TAL, denoted by L_{tal}, at the time of paging, $(L_{tal} - 1) \times c^p$ is added to the overall signaling overhead. When the TAL of a UE consists of only the serving cell, the paging overhead for the UE is zero. Similarly, in longer TALs, the serving cell should be excluded from the paging overhead. This is why the value one is reduced from L_{tal} in the paging overhead computation. This procedure is repeated for all the UEs in the UE-traces scenario. Finally, the obtained overall signaling overhead is the exact value for that UE-traces scenario.

Method II: $\mathbf{S}(\mathbf{t})$ Matrix

Given the UE-traces scenario and a TAL design, one can calculate the exact $\mathbf{S}(\mathbf{t})$ matrix instead of using any estimation formulas. In the signaling overhead calculation of TAL discussed in Section 6.1, the $s_{ij}(\mathbf{t})$ element of the $\mathbf{S}(\mathbf{t})$ matrix is defined as the ratio of UEs in cell i having j in their TAL. By having the complete UE-traces scenario of the network, the TAL held by each UE is known, and therefore the true $\mathbf{S}(\mathbf{t})$ matrix can be computed given a TAL solution. The time scaling of the cell loads in the UE-traces scenario should be again considered in the $s_{ij}(\mathbf{t})$ calculation. Although the true fractional $\mathbf{S}(\mathbf{t})$ matrix is computed, the signaling overhead resulted from this matrix is still an estimation and does not necessarily match the signaling overhead obtained by the UE-traces scenario. The reason for considering this method is to study the accuracy of the signaling overhead computation by Equation (2.2) for various TAL schemes.

7.4 Performance Evaluation

The cellular network of the Lisbon downtown area provided by the EU MOMENTUM project [46] is considered. Ten UE-traces scenarios are generated resembling one set of data of cell load and handover in one hour. The number of UEs in all UE-traces scenarios is equal to $25,000$. The one-hour time period is divided into 60 equal time intervals. Thus, every time interval is equivalent to 1 minute.

The dimension of the scenario matrices in all ten scenarios is $25,000 \times 60$. The parameter α is 0.0167, which means 1.67% of the UEs are paged in the UE-traces scenarios. The possibility of a UE being called several times during the one hour time period is also considered. Among the elements of the scenario matrix, the number of active UEs scaled by the time proportion is 5% of the total cell load. While a UE is in the active state, there is no signaling overhead for that UE during the call. The average length of the call durations is assumed to be 3 minutes.

Based on the discussion in Section 7.2, the cell load and handover data are aggregated from each UE-traces scenario. The aggregated cell load and handover are comparable to those in the original data. The assumptions are that $c^u = 1$ and $c^p = 0.1$; this ratio is common in the literature [17, 27, 33]. The optimum standard TA configuration is computed by CPLEX [31] using the model in [62] for each UE-traces-scenario data set. TAL configuration is at the granularity of the cell level.

The three algorithms presented in Section 6.2 for designing TAL are implemented in MATLAB. The computations are run on a processor of type Intel Core 2 Duo with the clock speed of 2.1 GHz.

7.4.1 Standard TA Configuration

The signaling overhead of the standard TA (STA) configuration can be simply computed from the aggregated cell load and handover. For comparing the signaling overhead given by method II to the accurate result of method I in Section 7.3, both methods are applied to the STA configuration of the ten UE-traces scenarios. The results are presented in Table 7.1. The values for all UE-traces scenarios are very close to each other, because all of them are based on the same set of original cell load and handover data. The paging overheads from method I are slightly different from the ones from method II. The reason is that method I

Table 7.1 Signaling overheads of the STA configuration.

Scenario	Method I (exact)			Method II		
No.	TAU	Paging	Overall	TAU	Paging	Overall
1	421	598.7	1019.7	421	579.2	1000.2
2	428	549.6	977.6	428	560.0	998.0
3	382	607.2	989.2	382	602.8	984.8
4	449	498.9	947.9	449	508.8	957.8
5	357	622.3	979.3	357	632.0	989.0
6	331	616.1	947.1	331	618.6	949.6
7	325	622.6	947.6	325	618.0	943.0
8	483	494.4	977.4	483	490.0	973.0
9	382	620.4	1002.4	382	599.0	981.0
10	363	624.9	987.9	363	622.6	985.6

considers the actual pagings that have happened in the UE traces, but method II only accounts that α of the UEs are paged.

7.4.2 Trace-independent TAL Configuration

A TAL configuration (TAL1) based on the algorithm in Section 6.2.1 considering the two-hops calculation is designed for each UE-traces scenario. The algorithm took in average 4 minutes for designing TAL for one UE-traces scenario. In order to calculate the overall signaling overhead, both methods are applied. The results are presented in Table 7.2. The observations from the table can be summarized as follows.

- The overall signaling overhead from the TAL design is 54% to 58% better than the ones obtained from the optimal STA configurations.
- The overall signaling overheads from method II are 8% to 16% lower than the direct computation using all UE traces. It should be recalled that Algorithm 3 is doing a rather coarse estimation of the $\mathbf{S}(\mathbf{t})$ matrix. Hence, this observation is reasonable.
- The TAL algorithm significantly reduces the TAU overhead in respect to the paging overhead. By the exact evaluation of method I, the TAU overheads are reduced by 83% to 90% comparing to the values obtained from the STA configuration.

Table 7.2 Signaling overheads of TAL1 configuration.

Scenario	Method I (exact)			Method II		
No.	TAU	Paging	Overall	TAU	Paging	Overall
1	71.0	348.8	419.8	118.0	252.5	370.5
2	49.0	382.4	431.4	92.0	285.3	377.3
3	64.0	382.4	446.4	88.7	276.9	365.6
4	64.0	340.2	404.2	115.9	254.9	370.8
5	48.0	380.2	428.2	65.3	291.8	357.1
6	43.0	372.8	415.8	69.3	292.5	361.8
7	36.0	366.2	402.2	62.2	288.1	350.3
8	49.0	356.5	405.5	90.8	270.0	360.8
9	64.0	381.1	445.1	85.9	284.3	370.2
10	33.0	375.2	408.2	71.5	293.8	365.3

- In method II, each element of the $\mathbf{S}(\mathbf{t})$ matrix is representing the exact ratio of UEs inside a cell having another cell in their TAL. However, unlike method I which is exact, method II considers the average behavior of the UEs. This is the reason for obtaining different overhead values from the two methods.

7.4.3 TAL Configuration based on Rule of Thumb

A TAL configuration (TAL2) based on the rule of thumb in Section 6.2.2 is designed for each UE-traces scenario. It took around 10 seconds to design TAL for one UE-traces scenario. Let us use both methods for signaling overhead calculation and have a comparison. The results of the overhead values given by the two methods are presented in Table 7.3. The observations from this table and the comparison to Table 7.1 can be summarized as follows.

- The overall signaling overheads from the TAL2 design are 49% to 56% better than the ones of the optimal STA configurations.

- The rule of thumb in designing TAL significantly reduces the paging overhead in respect to the TAU overhead. By the exact evaluation from method I, the paging overheads are reduced 67% to 73% comparing to the values obtained from the STA configuration.

Table 7.3 Signaling overheads of TAL2 configuration.

Scenario	Method I (exact)			Method II		
No.	TAU	Paging	Overall	TAU	Paging	Overall
1	317	160.0	477.0	451.0	176.7	627.7
2	295	162.4	457.4	467.4	179.4	646.8
3	257	163.6	438.6	436.6	180.2	616.8
4	312	162.9	474.9	476.6	180.7	657.3
5	274	165.0	439.0	455.4	182.1	637.5
6	267	163.0	430.0	439.7	180.1	619.8
7	285	162.7	447.7	438.5	179.5	618.0
8	268	157.1	425.1	440.9	176.4	617.3
9	306	162.8	468.8	461.1	180.3	641.4
10	306	163.9	469.9	470.4	181.8	652.2

This is expected because the rule of thumb tends to create small TALs.

- The overall signaling overheads from method II are 23% to 29% higher than the ones obtained by method I. The difference between the signaling overheads from the two methods is mostly due to the TAU overheads. The reason is that method II considers the average behavior, whereas method I is exact.

7.4.4 TAL Configuration based on UE Traces

An optimized TAL configuration (TAL3) based on 5% of the overall number of UE-traces is designed for each UE-traces scenario. It took around 2 hours for the algorithm to design the TAL for the 1250 UE traces. For these UEs, the signaling overheads obtained from the STA configuration and TAL3 are given in Table 7.4. The signaling overheads are calculated using method I. The significant reduction in the signaling overheads for the TAL solution shows that TAL3 has a potential in comparison to the STA scheme.

The last column of Table 7.4 gives the cost of collecting these 1250 UE traces. Recalling the discussion in Section 6.2.3, all these 1250 UEs are given TALs which only contain the updating cell. Thus, there will be no paging overhead, and the cost of UE-trace collection is the amount of TAU.

Table 7.4 Signaling overheads of TAL3 configuration for 1250 UEs.

Scenario	STA			TAL3			Cost
No.	TAU	Paging	Overall	TAU	Paging	Overall	TAU
1	29	29.8	58.8	1	4.2	5.2	389
2	25	30.3	55.3	0	5.9	5.9	355
3	19	30.1	49.1	1	6.6	7.6	359
4	19	34.3	53.3	3	8.0	12.0	365
5	13	27.7	40.7	3	4.8	7.8	368
6	17	35.3	52.3	4	7.2	11.2	351
7	28	28.2	56.2	0	7.1	7.1	355
8	25	19.2	44.2	4	5.9	9.9	383
9	22	14.5	36.5	1	5.5	6.5	355
10	21	33.5	54.5	2	7.0	9.0	387

For each scenario, TAL3 is applied to the entire UE-trace data. Let us use both methods for signaling overhead calculation and have a comparison. The results are presented in Table 7.5, and the observations from this table and Table 7.4 can be summarized as follows.

- The overheads given by method I show that except for scenario 8, there is a reduction in the overall signaling overhead. However, the improvement is much smaller in comparison to the results of the previous two TAL schemes.
- From Table 7.4, it is observable that by having 100% of the UE traces a high-quality TAL configuration is attainable. However, the results from Table 7.5 show that for most of these UE-traces scenarios, an optimized TAL based on 5% of UE traces is not a convincing design. A higher percentage of UE-traces have to be considered in order to improve the performance of the scheme.
- Considering the very time-consuming process of designing TAL3 for even a small portion of the UEs, and the fact that TAL3 is not achieving much improvement in the overall signaling overhead, this scheme is not recommended.
- Comparing the amount of improvement in the overall signaling overhead to the cost of collecting the UE traces, it is apparent that collecting the UE traces by TAL is not advantageous.

Table 7.5 Signaling overheads of TAL3 configuration for 25000 UEs.

Scenario	Method I (exact)			Method II		
No.	TAU	Paging	Overall	TAU	Paging	Overall
1	267	611.3	878.3	281.3	594.2	875.5
2	282	585.0	867.0	296.7	590.0	886.7
3	276	624.7	900.7	287.3	609.8	897.1
4	321	521.9	842.9	337.1	521.0	858.1
5	294	642.4	936.4	300.7	635.0	935.7
6	386	340.2	726.2	444.0	479.7	923.7
7	385	469.0	854.0	430.5	570.8	1001.3
8	635	375.9	1010.9	672.8	436.9	1109.7
9	315	504.8	819.8	350.8	559.0	909.8
10	468	427.7	895.7	522.6	534.9	1057.5

7.5 Conclusions

In this chapter, the performance of the three TAL-design approaches discussed in Chapter 6 are evaluated, and compared to each other and also to the optimal standard TA scheme. The numerical results from this study show that generally the TAL schemes can significantly reduce the signaling overhead compared to the standard TA scheme in a large-scale network. The first algorithm which designs TAL independent from UE traces is the most recommended one for large-scale networks. The rule of thumb is a very fast approach to obtain a good TAL assignment. However, because it only considers the one-hop neighbors of the TAL of each cell, the design may not perform well for some scenarios, like in the train scenario for which it is more reasonable to include more cells forming paths in the TAL. Another conclusion is that designing TAL based on UE traces may result in a configuration which only performs well for the traces collected, and could not be generalized to a large-scale network.

Chapter 8

A Comparative Study of Dynamic and Static TAs

Conventionally, TAs are manually configured and the configuration is static. While network conditions change, this approach is not efficient for reconfiguring TAs. The static configuration often does not perform very well in signaling overhead, but it has the advantage of low computational complexity. In contrast, a dynamic approach that frequently updates the configurations may achieve better results, while requiring a higher degree of computational effort [19].

Automatic dynamic configuration is a key aspect for Self-Organizing/Optimizing Networks (SON). In this chapter, the dynamic and static approaches are applied to both standard TA and TAL schemes, and their performance are analyzed and compared to each other. The work presented here has been partially published in [44].

8.1 Self-Organizing Networks

Although the concept of automated reconfiguration is not new in cellular networks and there are already extensive uses of automated processes in performance engineering, introducing a higher level of automation remains one of the key topics in cellular communications. In Releases 8 and 9, 3GPP is standardizing self-optimizing and self-organizing capabilities for LTE. This is a continuation of the natural evolution of automation in cellular networks by extending the scope deeper into the network for LTE [6]. SON techniques aim at doing both planning and reconfiguration in a semi-autonomous fashion.

Until now, cellular operators have been using an off-line approach to configure TAs. Due to the complexity of TA reconfiguration, the operators mostly decide about the TA of each cell at the time of network deployment, and changes are made only in case of extreme performance degradations. In SON, the network continuously collects UE statistics and monitors performance indicators, and there is a potential of improving the TA designs or the TALs in short time intervals *without any cost in terms of service interruption* [54].

8.2 The Performance Evaluation Framework

The performance evaluation framework in this chapter consists of three parts: One is to compare static and dynamic TAs for the standard TA (STA) scheme. The other one is to compare static and dynamic TALs, and the last part is to compare the STA scheme to the TAL scheme within the static and dynamic frameworks. In the static framework, one static STA or TAL configuration is applied and evaluated for the entire evaluation period T. The period T is divided into equal time intervals. In the dynamic framework, the STAs or TALs are reconfigured for each time interval of T.

In this chapter, for the sake of simplicity, the rule of thumb is the chosen scheme in configuring TALs. To design an optimum STA configuration and a TAL design based on rule of thumb presented in Section 6.2.2, the only required data are the cell loads and handovers. In order to analyze the behavior of each scheme in the static and dynamic frameworks, it is assumed that one set of data is given for each time interval.

In this section, acronyms in signaling overhead evaluation are introduced and explained. Table 8.1 defines the acronyms, which are used for compactness.

- ISO-DSTA is the ideal value one could achieve by the STA scheme. It represents the signaling overhead of applying the STA design made for data of time interval τ and re-evaluate it for the same interval. The unreal assumption here is the possibilty of getting back in time and applying the optimum design to the same data which is only available by the end of the time interval. Because there is a

Table 8.1 Acronyms used for various signaling overhead results.

ISO-DSTA	Ideal Signaling Overhead of Dynamic Standard TA
PSO-DTAL	Potential Signaling Overhead of Dynamic TAL
ASO-DSTA	Actual Signaling Overhead of Dynamic Standard TA
ASO-DTAL	Actual Signaling Overhead of Dynamic TAL
SO-SSTA	Signaling Overhead of Static Standard TA
SO-STAL	Signaling Overhead of Static TAL

reconfiguration in each time interval, the acronym belongs to the dynamic framework.

- PSO-DTAL is the minimum potential value one can achieve by the TAL obtained from the rule of thumb. PSO-DTAL represents the signaling overhead of applying the TAL design made for data of time interval τ to the same data. It is similar to ISO-DSTA. However as the rule of thumb is not reaching the exact optimal signaling overhead for TAL, the word "ideal" is replaced by "potential". Because there is a reconfiguration in each time interval, the acronym belongs to the dynamic framework.

- ASO-DSTA and ASO-DTAL represent the signaling overhead of applying the design made for data of time interval τ to the data of time interval $\tau + 1$. These schemes are practically feasible and result in the "actual" signaling overhead. However, the optimum design based on the history of UEs' behaviors might not perform well in the next time interval. Because there is a reconfiguration in each time interval, the acronyms belong to the dynamic framework.

- SO-SSTA and SO-STAL represent the signaling overhead of applying the configuration designed for the average data of the entire period T to all time intervals. Because there is no reconfiguration, these acronyms belong to the static framework.

8.3 A Case Study

The same network of Lisbon down-town area considered in the previous chapters is used in this study. The network consists of 60 sites and 164

Table 8.2 Static and dynamic STA comparison.

	Total TAU	Total Paging	Overall
ISO-DSTA	1.7607×10^4	1.3780×10^4	3.1387×10^4
ASO-DSTA	2.0062×10^4	1.3740×10^4	3.3788×10^4
SO-SSTA	1.9114×10^4	1.3931×10^4	3.3045×10^4

cells. As putting the cells of one site in different TAs is not typical, the performance evaluation has been done based on sites.

The time period is the 24 hours of one day and each time interval is 15 minutes. A predefined set of UE location and mobility pattern that varies over the day has been used. The UE activities are very low during the night and higher during the day with two peaks at the beginning and the end of the office hours. This gives varying cell loads and handovers over the 96 time periods, each being 15 minutes.

The cost ratio of a single update over a single paging (c^u/c^p) is set to be 10. The call intensity factor α is assumed to be 0.05.

8.3.1 A Comparison of Static and Dynamic STA

For each set of data, the optimum STA configuration is computed by CPLEX [31] using the model in [62]. The optimum design for time interval τ is first applied to the data set of τ to get ISO-DSTA, and then to the data set of $\tau+1$ to obtain ASO-DSTA. Equation (2.2) is used for calculating the signaling overhead.

For static STA, the evaluation takes the average of cell load and handover of all the data sets of the entire day, and an optimum STA configuration is designed based on the average data. This is used as a static TA configuration. Figure 8.1 illustrates all three signaling overheads of the STA scheme for all time intervals of the day. The curves in this figure show that the performance of static and dynamic STA schemes are very close. Also, ASO-DSTA is only slightly higher than ISO-DSTA. This means that there is a correlation between the data of the adjacent time intervals.

The total signaling overheads of the 24-hours time period are given in Table 8.2. The results in the table show that ASO-DSTA is actually slightly higher than SO-SSTA. This suggests that while applying the STA scheme to a network, a static STA is sufficient and close to optimum. This is a valuable observation, because under the STA scheme,

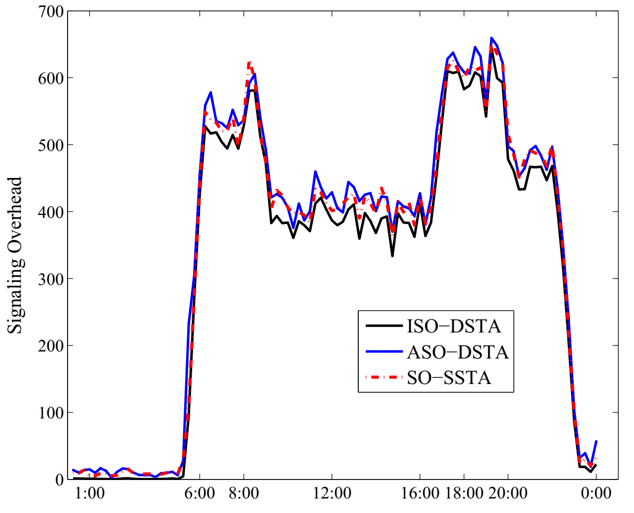


Figure 8.1 Signaling overhead comparison of STA configurations.

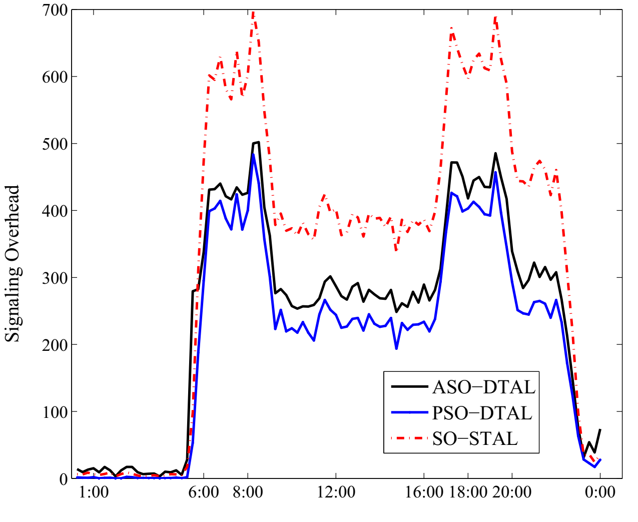


Figure 8.2 Signaling overhead comparison of TAL configurations.

Table 8.3 Static and dynamic TAL comparison.

	Total TAU	Total Paging	Overall
PSO-DTAL	1.2782×10^4	7.5795×10^3	2.0362×10^4
ASO-DTAL	1.6234×10^4	7.5567×10^3	2.3805×10^4
SO-STAL	2.7412×10^4	5.3835×10^3	3.2796×10^4

reconfiguration is currently a costly exercise (not considering SON).

8.3.2 A Comparison of Static and Dynamic TAL

For each time interval, a TAL configuration based on the rule of thumb presented in Section 6.2.2 is derived. The TAL design derived for time interval τ is first applied to the data set of τ to get PSO-DTAL, and then to the data set of $\tau + 1$ to obtain ASO-DTAL. Equation (6.2) is used for computing the signaling overheads, with $\gamma_1 = 0.75$ and $\gamma_2 = 0.15$.

A static TAL configuration based on the average data of cell load and handover of the entire time period T is applied to all the data sets, and the corresponding SO-STAL is computed.

Figure 8.2 illustrates signaling overheads of the static and dynamic TAL configurations. For the dynamic configuration, the ASO-DTAL curve is slightly higher than PSO-DTAL. Again it shows that the data has a correlation between the adjacent time intervals. Another observation from the figure is that in most of the time intervals the signaling overhead of the static TAL is significantly higher than the signaling overheads of the dynamic TAL.

The total signaling overheads are given in Table 8.3. From the results in the table, it can be seen that the total actual signaling overhead of the dynamic TAL is 27.4% lower than the total signaling overhead of the static TAL. Note that in SON, reconfiguration is a cost free process in terms of service interruption. Thus, the dynamic framework is highly recommended for the TAL scheme.

8.3.3 A Comparison of STA and TAL

This section compares the performance of STA and TAL schemes in the static and dynamic frameworks. Figure 8.3 compares ISO-DSTA to ASO-DTAL. The graphs show that except for the very low traffic hours (midnight to 6am), the signaling overhead of the dynamic TAL is

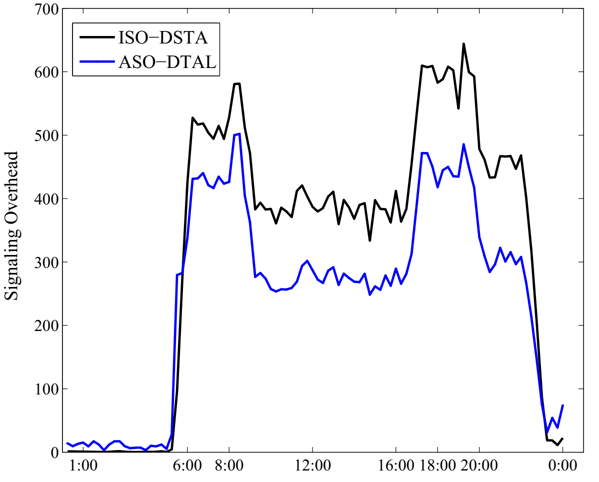


Figure 8.3 Signaling overhead comparison of dynamic STA and TAL.

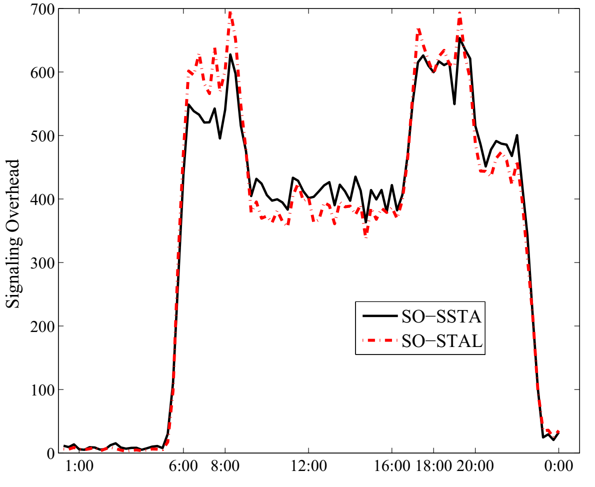


Figure 8.4 Signaling overhead comparison of static STA and TAL.

considerably lower than the ideal performance of the dynamic STA.

Figure 8.4 illustrates the signaling overhead of static STA and TAL configurations. The figure shows that for the static framework, there is

Table 8.4 Signaling overhead comparison of STA and TAL.

Total Overhead	STA	TAL	Improvement
I/PSO-D	3.1387×10^4	2.0362×10^4	35.1%
ASO-D	3.3788×10^4	2.3805×10^4	29.6%
SO-S	3.3045×10^4	3.2796×10^4	0.8%

no definite conclusion about the preference of STA or TAL. There are some time intervals in which TAL is performing better, and for the other time intervals TAL is performing worse.

Table 8.4 gives the total signaling overheads of each configuration in the static and dynamic frameworks. The second column of the table gives the signaling overheads of STA and the third column presents the signaling overheads of TAL. According to the values in the table, for both static and dynamic frameworks, TAL has an improved performance compared to the STA scheme. The overall signaling overheads are improved by 29.6% and 0.8% by TAL compared to STA for the dynamic and static frameworks, respectively. The graphs and the numerical results show that for the dynamic framework, TAL has a significant improvement in comparison to the STA scheme, while in the static case the improvement is not significant.

8.3.4 Justification of the Evaluation

The numerical results obtained in Chapter 7 showed that the signaling overhead computed by method II is more than 20% higher than the ones obtained by method I, in the case of rule-of-thumb TAL design. Recall that method I is accurate. Thus, it can be expected that generally the results presented here are over-estimations, and the true values can be lower than these.

Until now all results and figures for TAL are obtained by using $\gamma_1 = 0.75$ and $\gamma_2 = 0.15$. To have a better perspective towards the performance of TAL, the $\mathbf{S}(\mathbf{t})$ matrix is calculated for all combinations of $\gamma_1 = [0, 1]$ and $\gamma_2 = [0, 0.5]$ by a step size of 0.1 with the constraints in (6.3).

Figure 8.5 shows the distribution of the dynamic TAL overhead for all combinations of γ_1 and γ_2. It can be seen that the maximum is $2.4778 * 10^4$, which is still 26.8% better than the corresponding value of the STA scheme.

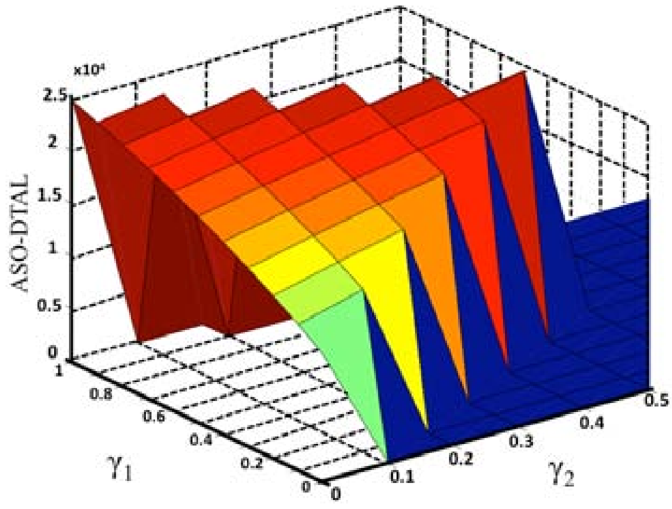


Figure 8.5 ASO-DTAL based on various combinations of γ_1 and γ_2.

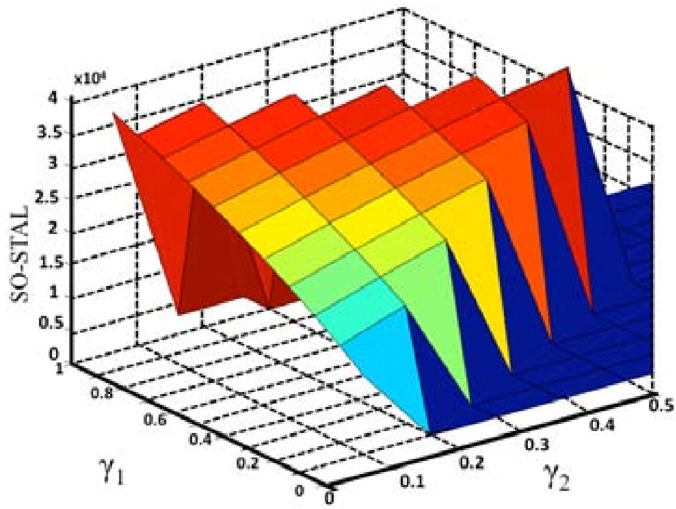


Figure 8.6 SO-STAL based on various combinations of γ_1 and γ_2.

Figure 8.6 is the same type of graph as Figure 8.5, only this time the total overhead of static TAL is considered. The maximum overhead in this figure is at point $\gamma_1 = \gamma_2 = 0.5$ and it is equal to 3.5757×10^4, which is 8.2% higher than the corresponding overhead of the STA scheme. This figure shows again that static TAL in some situations may not be as

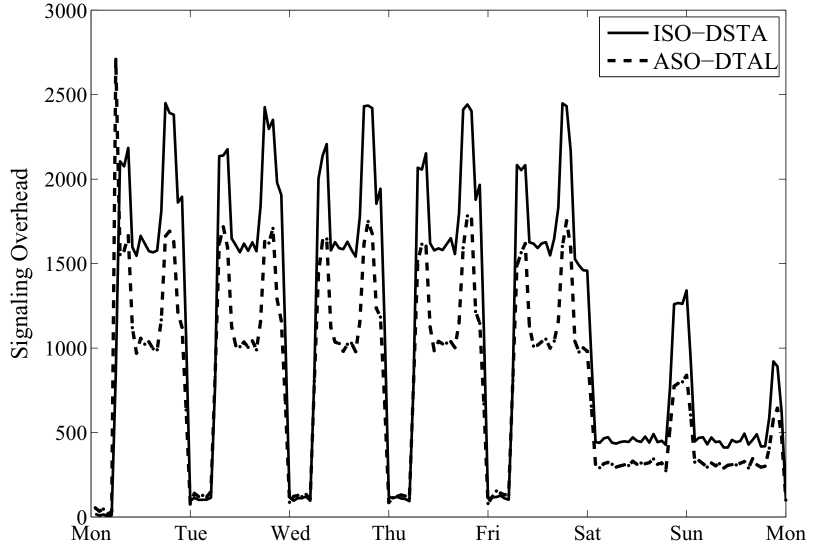


Figure 8.7 Signaling overhead comparison of dynamic STA and TAL for one-week data.

efficient as the STA scheme.

8.3.5 Addition Performance Comparison

Another set of experiments has been conducted for the Lisbon network with the time frame of one entire week. The traffic is assumed to vary over different times of each day and different days of the week (especially when comparing weekdays to weekends). The time interval is one hour, which is a more practical value to be used in a large-scale network.

Figure 8.7 illustrates the signaling overhead of the STA and TAL schemes for the dynamic framework, and Table 8.5 gives the corresponding ideal/potential and actual signaling overheads of the STA and TAL schemes. The numerical results in the table and the curves in Figure 8.7 show that for the dynamic framework, TAL performs clearly better than what can be ideally obtained by the STA scheme. This comparison indicates the potential of TAL in SON.

Table 8.5 Performance comparison on one-week data.

	Total TAU	Total Paging	Overall
ISO-DSTA	1.0956×10^5	8.2412×10^4	1.9197×10^5
ASO-DSTA	1.1702×10^5	8.2670×10^4	1.9963×10^5
PSO-DTAL	7.5623×10^4	4.8676×10^4	1.2430×10^5
ASO-DTAL	8.6208×10^4	4.8254×10^4	1.3452×10^5

8.4 Conclusions

In this chapter, the performance of STA and TAL schemes are examined under time-vary data within the static and dynamic frameworks. The results illustrate that by dynamic TAL, the performance of the network is significantly improved by reconfiguration. Another conclusion follows from the study of the static framework using averaged data: Unlike the STA scheme that performs close to optimal in the static framework with short time intervals, TAL works best if dynamic and frequent reconfigurations are applied for the whole time duration. Fortunately, this is possible due to the automatic reconfiguration feature in LTE. The numerical results from the one-week-data experiment demonstrate that dynamic TAL improves the performance of the network in the long run.

Chapter 9

Conclusions and Future Research

The thesis work has dealt with three themes. The *first* theme is TA design re-optimization considering a budget cost, and finding the pareto-optimal solutions for the trade-off between the signaling overhead and the reconfiguration cost (Chapters 3-4). Although these problems have been studied in the TA context, the results can be generalized to the study of LA and RA optimization. The *second* theme deals with the TAL scheme and its potentials compared to the standard TA scheme (Chapters 5-7). TAL is still a rather unexplored area, and it requires more investigation. The thesis gives some insight into the performance of TAL. The *third* theme is the dynamic framework explored in Chapter 8. LTE supports SON, which is one of the visions in future network management. The thesis examined the standard TA scheme and TAL under a dynamic evaluation framework, in order to investigate the aptness of the schemes for SON.

9.1 Conclusions

The work presented in the thesis justifies the benefit of tracking area planning and optimization for improving the performance in cellular networks. There are also some detailed conclusions from each specific theme studied in the thesis.

Once a TA design is in use, adopting a new solution of green-field optimization does not typically pay off in real networks. The repeated-local-search algorithm which is developed to solve the re-optimization problem in the thesis is able to approach high-quality solutions. The novelty of the approach is the consideration of the reconfiguration-cost budget.

Before applying any reconfiguration, a decision maker can be provided by a set of pareto-optimal solutions representing potential tradeoffs between the signaling overhead and the reconfiguration cost. The proposed integer programming model provides the exact pareto-optimal solutions, and the suggested GA algorithm gives close to optimal solutions for large-scale networks in short time.

The signaling overhead obtained from the TAL assigned by the presented local-search algorithm is half of the signaling overhead resulted from the optimal standard TA scheme. The rule of thumb in the thesis is a very simple and quick approach for assigning a reasonably good TAL for a large-scale network.

TAL works best when a dynamic frequent reconfiguration is applied. For the standard TA scheme, the difference between the dynamic and static TA is not significant, as long as there is not a major change in the mobility behavior of the UEs.

9.2 Suggestions for Future Works

There are still many open problems in the study of TA management of cellular networks. Some related topics deserving further research are summarized below.

In the thesis, the TAL scheme has been only examined for the improvement of the overall signaling overhead. Exploring the scheme by considering other parameters, such as load balancing, forms a future line of research.

The re-optimization problem explored in the thesis for the standard TA scheme can be extended to the TAL scheme. Even though in the TAL scheme the problem of service interruption in reconfiguration is solved, from the network standpoint it is still more suitable to avoid major changes between two consecutive configurations. Hence, another extension is to introduce a "change budget" in reconfiguring the TAL in each time interval of the dynamic framework.

In the performance evaluation of the static and dynamic frameworks, the aggregated data is based on short and equal-length time intervals.

Due to the effort required for data collection, it is of relevance to evaluate the overall signaling overheads resulted from considering data with higher level of aggregation.

Additional experimental analysis on larger networks and various topologies can give more insights into the performance of the proposed algorithms. Another topic is the investigation of alternative and better optimization algorithms, especially for the TAL scheme. One example can be to extend the idea of rule of thumb to neighbors other than the first-hop ones in order to overcome the existing limitation.

Bibliography

[1] 3GPP. TS 23.002 – v9.1.0, Network Architecture, 3G Release 9, 2009.
http://www.3gpp.org/ftp/specs/archive/23_series/23.002/.

[2] 3GPP. TS 23.401 – v8.6.0, General Packet Radio Services (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access, 3G Release 8, 2009.
http://www.3gpp.org/ftp/specs/archive/23_series/23.401/.

[3] 3GPP. TS 23.882 – v1.9.0, System Architecture Evolution: Report on technical options and conclusions, 3G Release 7, 2007.
http://www.3gpp.org/ftp/specs/archive/23_series/23.882/.

[4] 3GPP. TR 25.913 – v7.3.0, Requirements for EUTRA and EUTRAN, 3G Release 7, 2006.
http://www.3gpp.org/ftp/specs/archive/25_series/25.913/.

[5] 3GPP. TR 36-series, 3G Release 8, 2008
http://www.3gpp.org/ftp/specs/archive/36_series/.

[6] 3GPP TS 36.902, Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions, 2009.
http://www.3gpp.org/ftp/specs/archive/36_series/36.902/.

[7] 3G Americas, The Benefits of SON in LTE, White Paper, 2009.

[8] E.H.L. Aarts and J.K. Lenstra, *Local search in combinatorial optimization*, Wiley, 1997.

[9] D.K. Agrafiotis, Multiobjective optimization of combinatorial libraries, Technical Report in *IBM Journal of Research and Development*, 2001.

[10] I.F. Akyildiz, J.S.M. Ho and Y.B. Lin, Movement-based location update and selective paging for PCS networks, *ACM/IEEE Transactions on Networking*, vol. 4, pp. 629-638, 1996.

[11] I.F. Akyildiz, J. McNair, J.S.M. Ho, H. Uzunalioglu, and W. Wang, Mobility management in next generation wireless systems, *Proc. of IEEE*, vol. 87, pp. 1347-1384, 1999.

[12] S.Z. Ali, Design of location areas for cellular mobile radio networks, in *Proc. of IEEE Vehicular Technology Conference (VTC '02)*, pp. 1106-1110, 2002.

[13] A. Bar-Noy and I. Kessler, Tracking mobile users in wireless communications networks, *IEEE Transactions on Information Theory*, vol. 39, pp. 1877-1886, 1993.

[14] Y. Bejerano, M.A. Smith, J. Naor, and N. Immorlica, Efficient location area planning for personal communication systems, *ACM/IEEE Transaction on Networking*, vol. 14, pp. 438-450, 2006.

[15] P.S. Bhattacharjee, D. Saha, A. Mukherjee, and M. Maitra, Location area planning for personal communication services networks, in *Proc. of the 2nd ACM international workshop on modeling, analysis and simulation of wireless and mobile systems (MSWiM '99)*, pp. 95-98, 1999.

[16] E. Cayirci and I.F. Akyildiz, Optimal location area design to minimize registration signaling traffic in wireless systems, *IEEE Transactions on Mobile Computing*, vol. 2, pp. 76-85, 2003.

[17] M. Canales, J.R. Gallego, A.H. Solana, and A. Valdovinos, An adaptive location management scheme for mobile broadband cellular systems, in *Proc. of the 6th International Symposium on Wireless Communication Systems (ISWCS '09)*, pp. 333-337, 2009.

[18] A. Charnes and W.W. Cooper, Goal programming and multiple objective optimization, *European Journal of Operational Research*, vol. 1, pp. 39-45, 1977.

[19] J. Cowling, *Dynamic location management in heterogeneous cellular networks*, Thesis, University of Sydney, 2004. http://people.csail.mit.edu/cowling/hons/jcowling-dynamic-Nov04.pdf.

[20] D. Chung, H. Choo and H.Y. Youn, Reduction of location update traffic using virtual layer in PCS, in *Proc. of the International Conference on Parallel Processing (ICPP '01)*, pp. 331-338, 2001.

[21] I. Demirkol, C. Ersoy, M.U. Caglayan, and H. Delic, Location area planning in cellular networks using simulated annealing, in *Proc. of IEEE Conference on Computer Communications (INFOCOM '01)*, pp. 13-20, 2001.

[22] I. Demirkol, C. Ersoy, M.U. Caglayan and H. Delic, Location area planning and cell-to-switch assignment in cellular networks, *IEEE Transactions on Wireless Communications*, vol. 3, pp. 880-890, 2004.

[23] M. Ehrgott and X. Gandibleux (editors), *Multiple Criteria Optimization: State of the Art Annotated Bibliographic Surveys*, Kluwer Academic Publishers, 2002.

[24] G. Fan, I. Stojmenovic, and J. Zhang, Elimination of generalized ping-pong effects using triple-layers of location areas in cellular networks, *Computer Science and Information Systems*, vol. 5, pp. 1-16, 2008.

[25] P.C. Fishburn, Lexicographic orders, utilities and decision rules: a survey, *Management Science*, vol. 20, pp. 1442-1471, 1974.

[26] R. Fourer, D.M. Gay, and B.W. Kernighan, *AMPL - A Modeling Language for Mathematical Programming*, Boyd & Fraser, 1993.

[27] P. Garcia, V. Casares, and J. Mataix, Reducing location update and paging costs in a PCS network, *IEEE Transactions on Wireless Communications*, vol. 1, pp. 200-209, 2002.

[28] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Welsey, 1989.

[29] P.R.L. Gondim, Genetic algorithms and location area partitioning problem in cellular networks, in *Proc. of IEEE Vehicular Technology Conference (VTC '96)*, pp. 1835-1838, 1996.

[30] Gurobi 3.0.1, Reference Manual, 2010.

[31] ILOG CPLEX 10.0, User's Manual, 2006.

[32] H. Isermann, Proper efficiency and the linear vector maximum problem, *Operations Research*, vol. 22, pp. 189-191, 1974.

[33] K. Kyamakya and K. Jobmann, Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis, *IEEE Transactions on Vehicular Technology*, vol. 54, pp. 687-708, 2005.

[34] C.Y. Lee, S.J. Kim and T. Park, A design of multi-layered location registration areas in microcellular systems, *Telecommunication Systems*, vol. 14, pp. 107-120, 2000.

[35] B. Liang and Z. Haas, Predictive distance-based mobility management for PCS networks, in *Proc. of IEEE Conference on Computer Communications (INFOCOM '99)*, pp. 1377-1384, 1999.

[36] Y.B. Lin, Reducing location update cost in a PCS network, *IEEE Transactions on Networking*, vol. 5, pp. 25-33, 1997.

[37] G.L. Lyberopoulos, J.G. Markoulidakis, D.V. Polymeros, D.F. Tsirkas, and E.D. Sykas, Intelligent paging strategies in future mobile telecommunication networks, *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 534-554, 1995.

[38] T. Mach and R. Tafazolli, Mass mobility signaling congestion avoidance mechanism using randomized time distribution of cell reselections, in *Proc. of International Conference on Telecommunications (ICT '09)*, pp. 238-242, 2009.

[39] S. Mishra and O.K. Tonguz, Most recent interaction area and speed-based intelligent paging in PCS, in *Proc. of IEEE Vehicular Technology Conference (VTC '97)*, pp. 505-509, 1997.

[40] Mitsubishi Electric, Tracking areas sizes and tracking area list optimization, Technical Report in 3GPP TSG RAN WG3 Meeting, R3-071931, 2007.

[41] S. Modarres Razavi and D. Yuan, Performance improvement of LTE tracking area design: a re-optimization approach, in *Proc. of the 6th ACM International Workshop on Mobility Management and Wireless Access (MobiWac '08)*, pp. 77-84, 2008.

[42] S. Modarres Razavi, D. Yuan, F. Gunnarsson, and J. Moe, Optimizing the tradeoff between signaling and reconfiguration: a novel bi-criteria solution approach for revising tracking area design, in *Proc. of IEEE Vehicular Technology Conference (VTC '09)*, 2009.

[43] S. Modarres Razavi, D. Yuan, F. Gunnarsson, and J. Moe, Exploiting tracking area list for improving signaling overhead in LTE, in *Proc. of IEEE Vehicular Technology Conference (VTC '10)*, 2010.

[44] S. Modarres Razavi, D. Yuan, F. Gunnarsson, and J. Moe, Dynamic tracking area list configuration and performance evaluation in LTE, in *Proc. of IEEE Global Communications Conference Workshop (GLOBECOM Workshop '10)*, 2010.

[45] S. Modarres Razavi, D. Yuan, F. Gunnarsson, and J. Moe, Performance and cost trade-off in tracking area reconfiguration: a pareto-optimization approach, submitted for journal publication, 2010.

[46] IST-2000-28088 Momentum project. http://momentum.zib.de.

[47] Motorola, Long Term Evolution (LTE): A technical overview, Technical White Paper, 2007.

[48] Z. Naor and H. Levy, Minimizing the wireless cost of tracking mobile users: an adaptive threshold scheme, in *Proc. of IEEE Conference on Computer Communications (INFOCOM '98)*, pp. 720-729, 1998.

[49] NEC, Tracking area concepts and influence on paging load, Technical Report in 3GPP TSG RAN2 Meeting, R2-070655, 2007.

[50] C.H. Park and S. Soni, A simulated annealing based solution approach for the two-layered location registration and paging areas partitioning problem in cellular mobile networks, *European Journal of Operational Research*, vol. 178, pp. 579-594, 2007.

[51] K.E. Parsopoulos, D.K. Tasoulis, and M.N. Vrahatis, Multiobjective optimization using parallel vector evaluated particle swarm optimization, in *Proc. of IASTED International Conference on Artificial Intelligence and Applications (AIA '04)*, pp. 823-828, 2004.

[52] J. Plehn, The design of location areas in a GSM network, in *Proc. of IEEE Vehicular Technology Conference (VTC '95)*, pp. 871-875, 1995.

[53] G.P. Pollini and C.L. I, A profile-based location strategy and its performance, *IEEE Journal on Selected Areas in Communications*, vol. 15, pp. 1415-1424, 1997.

[54] QUALCOMM Europe and T-Mobile, New SON user case: tracking area optimization, Technical Report in 3GPP TSG RAN WG3 Meeting, R3-071594, 2007.

[55] C. Rose and R. Yates, Minimizing the average cost of paging under delay constraints, *Wireless Networks*, vol. 1, pp. 211-219, 1995.

[56] D. Saha, P.S. Bhattacharjee, and A. Mukherjee, Time efficient heuristics for cell-to-switch assignment in quasi-static/dynamic location area planning of mobile cellular network, *Computer Communications*, vol. 30, pp. 326-340, 2007.

[57] S.K. Sen, A. Bhattacharya and S.K. Das, A selective location update strategy for PCS users, *Wireless Networks*, vol. 5, pp. 313-326, 1999.

[58] R.E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*, John Wiley & Sons, 1986.

[59] R.E. Steuer, L.R. Gardiner, and J. Gray, A bibliographic survey of the activities and international nature of multiple criteria decision making, *Journal of Multi-Criteria Decision Analysis*, vol. 5, pp. 195-217, 1996.

[60] S. Tabbane, An alternative strategy for location tracking, *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 880-892, 1995.

[61] E.G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, 2009.

[62] D.W. Tcha, T.J. Choi, and Y.S. Myung, Location-area partition in a cellular radio network, *Journal of the Operational Research Society*, vol. 48, pp. 1076-1081, 1997.

[63] G. Wan and E. Lin, A dynamic paging scheme for wireless communication systems, in *Proc. of the 3rd ACM International Conference on Mobile Computing and Networking (MobiCom '97)*, pp. 195-203, 1997.

[64] W. Wang, I.F. Akyildiz, G.L. Stüber and B. Chung, Effective paging schemes with delay bound as QoS constraints in wireless systems, *Wireless Networks*, vol. 7, pp. 455-466, 2001.

[65] T. Winter, Mobility management and network design for UMTS, in *Proc. of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC '04)*, 2004.

[66] V.W.S. Wong and V.C.M. Leung, Location management for next generation personal communication networks, *IEEE Network*, vol. 14, pp. 18-24, 2000.

[67] V.W.S. Wong and V.C.M. Leung, An adaptive distance-based location update algorithm for PCS networks, in *Proc. of IEEE International Conference on Communications (ICC '01)*, pp. 2001-2005, 2001.

[68] E. Zitzler and S. Kunzli, Indicator-based selection in multi-objective search, in *Proc. of the 8th International Conference on Parallel Solving from Nature (PPSN '04)*, pp. 832-842, 2004.

[69] ZTE, Self optimization for tracking areas, Technical Report in 3GPP TSG RAN WG3 Meeting, R3-071415, 2007.