This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

Learning-based Tracking Area List Management in 4G and 5G Networks

Jessica Moysen and Mario García-Lozano

Abstract—Mobility management in 5G networks is a very challenging issue. It requires novel ideas and improved management so that signaling is kept minimized and far from congesting the network. Mobile networks have become massive generators of data and in the forthcoming years this data is expected to increase drastically. The use of intelligence and analytics based on big data is a good ally for operators to enhance operational efficiency and provide individualized services. This work proposes to exploit User Equipment (UE) patterns and hidden relationships from geo-spatial time series to minimize signaling due to idle mode mobility. We propose a holistic methodology to generate optimized Tracking Area Lists (TALs) in a per UE manner, considering its learned individual behaviour. The k-means algorithm is proposed to find the allocation of cells into tracking areas. This is used as a basis for the TALs optimization itself, which follows a combined multi-objective and single-objective approach depending on the UE behaviour. The last stage identifies UE profiles and performs the allocation of the TAL by using a neural network. The goodness of each technique has been evaluated individually and jointly under very realistic conditions and different situations. Results demonstrate important signaling reductions and good sensitivity to changing conditions.

Index Terms—Mobility Management, Tracking Area Lists, Mobile Networks, Big Data Analytics, Multi-Objective Optimization

1 Introduction

Mobility management is one of the core procedures of cellular networks. It allows to track user equipment (UE) and deliver communication services in a seamless manner. Being in continuous evolution and research of new ideas, it is a key element in the development of mobile networks towards their fifth generation (5G).

Mobility is managed at two different levels depending on whether the UE is in connected or idle mode. In the first case, UEs inform about events in which neighbour base stations provide a better reception level than the current server. This means UEs are located at cell level and handover is the procedure in charge of transferring the connections between base stations. On the other hand, when UEs are in idle mode they do not report all cell reselections. They just inform about changes in registration areas conformed by several cells. This is done to minimize uplink signaling, power consumption and interference. It also reduces the processing load in the Mobility Management Entity (MME) in the core network. In the context of pre Long Term Evolution (LTE) networks, such areas are named location or routing areas for circuit and packet switched domains respectively. Each cell belongs to a single location/routing area and continuously broadcasts the corresponding identifier. Thus, every time a UE detects an area change, it reports the event to the core network by sending an update message (uplink signaling). Whenever a new incoming connection arrives, it is possible to locate the UE since all the cells in its area will send a paging message (downlink signaling) until the UE answers back or a timer expires.

From a planning viewpoint, the number of cells in each registration area is a compromise between the number of area updates and the volume of paging messages, i.e. between uplink and downlink signaling load. Hence, it is a typical and well-studied optimization problem [1]. Area updates depend on UE spatial density and their mobility level and, on the other hand, paging depends on traffic load.

Since each cell belongs to a single routing/location area, the resulting groups are non-overlapping. Consequently, one of the problems with this mobility management approach is the existence of border cells. They are in charge of transmitting the update messages from all users entering into the area, thus needing extra signaling channels and so reducing their capacity to carry data traffic. Collision rates in the uplink random access channel are also increased, which impacts on collision resolution time. This is even more important in environments with a massive number of devices such as railways [2].

LTE tackled this problem by introducing the concept of Tracking Area List (TAL) [3]. Tracking areas (TAs) is the name of registration areas in the context of LTE and New Radio (NR). The particularity in these systems is that a UE in idle mode can handle a list of them. This way, when the UE enters a TA that is not contained in its current list, it reports the event to the network and gets back a new TAL. Since different UEs can be configured with different lists, the signaling associated with updates is distributed among several cells and area borders become blurred. Also, the newly allocated list may be overlapped with the previous one. This solves the problem of frequent updates due to UEs moving in the border between two TAs or due to sharp changes in channel conditions.

Our work lies in the context of 4G and 5G networks, where radio access network densification is one of the key ingredients to increase network capacity. A new layer of small cells is expected to serve the massive number of UEs from machine type communications. In this context, handling mobility is a very challenging issue with most of

J. Moysen and M. García-Lozano are with the Department of Signal and Theory Communications, Universitat Politècnica de Catalunya-UPC, Barcelona, Spain. Email:{jessica.moysen, mariogarcia}@tsc.upc.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

the efforts being set in handover mechanisms. Idle mode mobility requires imaginative ideas and improved management so that signaling is kept minimized and far from congesting the network. Also, artificial intelligence is expected to play a key role in 5G network self-optimization thanks to the availability of big data provided by users and network elements.

Cellular networks have become massive generators of data, which can be smartly exploited by operators thanks to new processing and storage capabilities favored by multi-access edge computing and fog/cloud Radio Access Network (RAN) architectures. Indeed, many recent works have proposed frameworks for big data driven mobile network optimization [4] [5].

This work proposes to exploit big data analytics discovering particular UE patterns from geo-spatial time series. In particular, we propose a mechanism that generates optimized TALs in a per UE manner considering its learned individual behaviour.

The remainder of the paper is organized as follows: Section 2, describes the state of the art in this area and the main contributions. Section 3, deals with UE pattern characterization and classification. Section 4 presents the procedure to assign cells into TAs and the optimization of TAs into TALs for the different types of UEs. We present in Section 5 the scenario considered. Section 6 focuses on the analysis and performance evaluation of each stage of the framework. Then, Section 6.4 analyzes the results obtained when the complete strategy operates in a network in exploitation. Finally, we draw our conclusions and future work in Section 7.

2 Related work and Contributions

Mobility management for idle mode terminals is composed of two procedures, paging and TAL updates. The latter being formed by two sub problems, TA planning and TAL allocation. So we can find different groups of works depending on where they put the main focus.

2.1 Paging Schemes

Several authors have formulated new paging schemes [6]–[8] to reduce the amount of signalling from the MME. Examples are: Blanket polling, shortest distance first, sequential paging, velocity paging [8]. Others adapt to ultra dense deployments and act hierarchically so that macro-cells redistribute paging messages to pico-cells via the X2 interface as a means to reduce signalling load at the core network [9]. Other interesting approaches exploit UE mobility and traffic patterns [6], [7]. However, current commercial networks only use blanket polling, in which a paging message is distributed in all the cells of the UE TAL. Hence, operators can only reduce signaling by optimizing the registration areas design and this is the context of the present work.

2.2 TAL Updates: TA planning and TAL allocation

Research on location update strategies started with Global System for Mobile Communications (GSM) systems and the optimization of its *location areas*. There has been extensive work on the matter and two big strategies can be identified, static and dynamic. In the following paragraphs we review the different approaches, most of them being also summarized in Table 1 in a comparative manner.

In the first case, the areas are static and common for all users. There are several out-of-standard approaches but the only possible implementation today is the one based on common registration areas, as previously explained. The problem is an NP hard combinatorial optimization one. Indeed, the sub-problem of assigning cells to location areas is mathematically equivalent to the bin-packing problem [10], which is known to be NP-complete [11]. Location areas (or TAs in LTE and NR) correspond to bins due to their paging capacity limit. We try to pack the cells to them in such a way that the number of areas is minimized so as to have a network with minimum signaling. Given the NP-completeness, meta-heuristics have been identified as an effective tool to tackle the problem. The authors in [12] review several of the works that deal with this strategy following a static approach, without taking lists into account.

In [13], [14], M. Razavi et al. propose interesting heuristics named the *local search* algorithm and the *rule of thumb* method to assign a proper TAL to each cell. Therefore, TALs are static but overlapped. This means that all UEs requiring a TAL update in a certain cell get the same list. Since the original cells are different for each UE, updates are distributed and TAL diversity is achieved. On the other hand, there is no per UE adaptation and each TA is composed of one cell. These designs improve the conventional use of registration areas, i.e. without creating lists and their low computational cost would allow a fast adaptation of TALs to different periods of time. The authors in [15] go one step ahead and optimize TAL allocation at the cell level by using a variable-order Markov model. This allows to choose the length of the UE routes to be considered so that the problem complexity is kept bounded. However, since the TAL size is assumed to be 1 TA, it would not be possible to perform a per-UE allocation under current standards.

The distribution of cells into location areas allows computing globally good solutions. However, since they are common to all UEs, they cannot adapt to the user behaviour in terms of mobility and traffic activity. Therefore, the signalling cost is not minimized. The present work optimizes both TAs and TALs and pursues a per UE allocation, thus falling into the category of dynamic approaches.

Dynamic approaches would allow to allocate TALs in a per UE basis. In many works, areas are created once a specific threshold is reached which can be based on distance, movement, time, profile or velocity [16]. Some works also add a reset in the TAL after an incoming call [17]. Once the process is launched, a new location area is allocated, usually as rings around the current cell. These mechanisms were widely investigated in the context of 2G and 3G. But, many of them required out-of-standard functionality. Even needing the mobile terminal to know the network topology in some cases. Alternatively, some works extend these schemes to LTE and allocate a TAL around the TA [18]–[20].

Many of these works fully rely on cells creating a perfect hexagonal plane tessellation since this also simplifies the analytic treatment (see Table 1). In fact, all the TAs in a TAL are also identical in size. However, hexagonal analysis

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

is useful for pre-planning tasks but at the end, coverage shapes are very irregular, with certain cells having a lot and irregular neighborhoods. The concept of *ring* is also very unclear in hierarchical cell structures and in environments with base stations having different transmission powers. This work proposes a strategy that is agnostic to the RAN layout and it is evaluated in a realistic network having cells of different sizes.

Dynamic approaches also require to treat the UE movement. In this sense, random walk and fluid flow model are very commonly used, other group of works use Markov chains. TAs are often assumed to be composed of 1 cell [18], [19], [21]. Then, a probabilistic cell residence time can be considered along with the probability of crossing to each neighbour. This allows doing an analytic treatment of the problem. This is improved in [21] by assuming a fluid flow model to describe the dependence between the cell and the area residence time. Therefore, signaling cost can be evaluated by means of an embedded Markov chain, though they are tested in a regular hexagonal layout. The authors focus on the analysis of the total signaling cost for UEs with a fixed trajectory, or at least, strong regularity. With a similar Markov approach, the authors in [20] consider UEs exhibiting weak regularity and include the TAL residence time in the study. They consider that the number of TAs in a TAL can vary, but all TAs are identical and shaped by concentric rings comprising multiple cells.

Another set of works deal with linear 1D scenarios, used to design strategies that are specific for railway scenarios. In this case, the route of UEs can be anticipated and overlapping TALs allow to distribute signaling among TAs. In this sense, the work [2] proposes that each cell manages a list of TALs and allocates them after finding the optimum proportions of UEs. In subsequent works [22], based on min-max optimization approach the authors proposed two separate solutions to decrease the TAU and paging signaling messages. The first one tries to minimize the TAU overhead while setting paging as a constraint, and the second one minimizes the paging overhead while fixing the TAU overhead as a constraint. In this model no assumption of UE trajectories is required.

On the other hand, a small subset of works rely on real traces, which require a simulation methodology. This is the context of the current work, in which we use the Google Maps API [23] to generate real UE mobility traces considering speed vectors that change along time and depending on the means of transportation. The evaluation is then based on very realistic simulations.

The utilization of ring shaped areas/lists allows adaptation to the UE speed, but they might well involve cells having a very low probability of being in the UE route. Hence, some authors have improved the adaptation to the specific route by using sectors of rings that partially adapt to the predicted route [24]. In [26], the average mobility behaviour of UEs is analyzed to eliminate or add specific cells not strictly included in the entire ring. In the current work, we use machine learning to learn repetitive UE routes and identify/predict whether the UE is following it. This way, the TAL can adapt as much as possible to the UE direction and speed so as to minimize signaling.

The benefits of TALs can only be achieved with a dynamic and adaptive assignment of the lists in a per UE basis. In this respect, some works argue about an increase in signaling. However, current standards indicate the tracking area list to every UE in non-access stratum messages from the MME. This happens no matter if the TAL is common for all UEs in the cell or if it is specific for each one. Hence, there is no extra complexity nor changes required if a per UE allocation is used. This work is developed in this context, we pursue not only a per UE optimization, but an optimization based on different per UE patterns and learned profiles.

Alternatively, as shown on Table 1, several works assume that TAs have a single cell. Note that using one cell per TA allows the maximum flexibility in TAL design. However, the TAL update response uses 40 bits per TA identifier, so there might be an important increase in the signaling volume if an upper bound is not considered. In fact, LTE imposes a maximum TAL size of 16. Therefore, the approaches cannot be applied in real networks unless TALs are very small, which means they cannot fully optimize signaling cost. Note that in some of the approaches, up to 4 rings are considered to create TALs [19], which would exceed the LTE limit. Even pedestrian UEs can go through 16 cells in dense urban networks in a rather short time. In the present work, we do not assume a fixed TA size to build TALs. We propose the use of a pre-optimization task based on k-means clustering [27] to find a suitable set of cells to include in each TA. So, in this sense the approach is holistic and offers a unified method. Our comparisons with other approaches, are done by assuming that this pre-optimization task was also executed in them.

The adaptation of TALs along time allows to design a dynamic framework that adjusts to the network dynamics. Thus, several works have posed the problem of TAL allocation as a Self-Organizing Network (SON) function. In [25], the authors introduced a dynamic framework to adapt the TAL optimization model to SON. TALs are recomputed along time as handover statistics and traffic change. Complexity is reduced by just considering congested areas with high number of TAUs at their boundaries and not to the whole network. The authors highlight that by introducing more history of network's behaviour, SON can significantly improve the network performance. The work in [26] also proposes a SON use case to minimize the signaling overhead. The cell to TAL problem is solved using a previous model based on pooling schemes [28]. The model is modified and divided into a bi-objective minimization problem that allows a dynamic approach for the optimal assignment of cell-to-TAL. This work also seeks to achieve load balancing through different MMEs. Finally, the work in [29] optimizes the management of TALs by introducing a two-stage based framework. The first one is executed offline and assigns TAs to TALs. The second stage assigns TALs to UEs in an online manner. In the online process, they take into account the UE behaviour in terms of mobility and connection frequency. In the offline process they proposed three different schemes, the first one is for small cities with high density populations, the second one is for networks with high mobility, and the third one, is for any kind of networks based on Nash Bargaining (NB) games to find a trade-off among TAU and paging messages.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

Table 1: Related work

Ref.	TA size	TAL size	Per UE allocation considering: UE velocity	Predicted route	Individual traffic	Technique	Mobility model	Network layout
[13]	1 cell	Static (per cell)	×	×	×	Heuristic. Local search	Handover statistics	Real city with 60 sites
[14]	1 cell	Static (per cell)	×	×	×	Heuristic. Rule-of-thumb	Handover statistics	Real city with 60 sites
[15]	Any	1 TA	×	×	×	Heuristic	Variable-order Markov based on learned paths	Realistic
[17]	Equal sized	Equal sized	×	×	×	Distance based, shortest-distance-first paging	1D and 2D random walk	1D and 2D Manhattan layout
[18]	1 cell	Dynamic	✓	×	✓	Movement based	Random walk	Hexagonal
[19]	1 cell	Dynamic for high speed UEs	✓	×	×	Speed based	Markov based for hexagonal topology	Hexagonal
[21]	1 cell	Dynamic	✓	×	×	Movement based	Fluid flow for analysis. Random walk for simulations	Hexagonal
[20]	Equal sized	Dynamic	✓	×	×	Movement based	Fluid flow for analysis. Random walk for simulations	Hexagonal
[24]	1 cell	8 or 12	×	✓	×	Distance based	Realistic traces	Hexagonal
[2]	Equal sized	Static pool of TALs per cell	×	✓	×	Linear programming and allocation based on optimum proportions	1D uniform	1D railway
[22]	Single-sited	Any	×	×	×	Min-Max Linear Programming (MMLP)	Handover statistics	Real city with 60 sites
[25]	Any	Any	×	×	×	Linear programming repeated along time	Handover statistics along time	Real city with 60 sites
[26]	1 cell	Static pool of TALs per cell	×	×	×	Decomposition in 2 sub-problems. Heuristic	Fluid flow model	Hexagonal
[⋆]	Any	Any	✓	✓	✓	Machine learning with clustering and metaheuristic	Real traces	Real city

2.3 Contributions

Given the previous paragraphs, we summarize our contributions next, being most of them already anticipated when explaining the existing research. The last row of Table 1 also contains the features of our proposal, denoted as [⋆].

Different from previous works, we intensively exploit the data already generated in the network to find patterns from geo-spatial time series and perform a per UE allocation considering its individual traffic activity and its mobility profile, both considering the speed and predicted route. Operator's data is used to bring the capability of predicting the UE behaviour. Thus, enabling mobile network operators to allocate optimized TALs in real time.

The proposed approach consists of two main steps. First, it estimates the type of UE based on UE's reports. We perform this estimation through supervised learning tools. This results in an appropriately tuned prediction model, which is then integrated in the next step, which assigns TALs considering paging and Tracking Area Update (TAU) signaling costs. More specifically, we perform this optimization by means of metaheuristics (genetic algorithms) and a multi-objective approach, which takes into account the following considerations:

1) Lists are adapted to learned repetitive routes that UEs may have for example while commuting on labor days. Generating TALs that adapt to them allows minimizing the number of updates the UE has to perform and hence, saving network resources and allowing a more energy efficient idle mode operation. This adaptation considers both the geographical route and speed changes. It is intuitive that UEs moving at high speed benefit from larger areas since this implies less update messages.
2) However, lists do also need to take into account UEs traffic activity. So UEs receiving many incoming sessions at certain periods of time will generate a lot of paging messages. Such signaling is spread among all cells in the UE TAL, so such profile of highly active UEs should be allocated smaller lists to keep network signaling low. On the other hand, UEs having very few paging messages could be allocated larger TALs, since their impact in the downlink is low and their uplink is improved. Such sizes can adapt to the UE activity in a spatio-temporal manner. Thus, depending on the time of day, weekday... and UE position, lists adapt their size to the predicted UE activity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

Given the previous two considerations, we propose a combined multi-objective and single-objective strategy to group TAs into TALs. The multi-objective approach is applied to UEs having different levels of traffic activity but a non-predictable route. Multi-objective analysis allows to present signalling tradeoffs to the operator, in charge of choosing one or another solution in specific areas of the network and given its particular constraints. The single objective optimization finds the most suitable TAL for UEs whose route is repetitive in a certain geographic and time coordinate. Metaheuristics are executed offline and the optimized TALs are applied following the real time decisions of a back-propagation Neural Networks (NN), whose design and error evaluation is also a matter of analysis.

Every time a new TAL update is required, the approach allocates customized TALs in a per UE manner. Both the dynamics and traffic of UEs are considered in the decision. Recall, that in static approaches, the areas are static and common for all users and this is clearly not the case. The strategy can be summarized as identification/classification plus allocation of TALs. Optimized TALs are computed offline and they are considering the learned routes and traffic patterns. The allocation itself is executed following the standard procedures from 3rd Generation Partnership Project (3GPP), using information elements in non access stratum messages.

The main benefit for the learning method is that it allows self-awareness. That is, by learning UE's behaviour the network is capable to perceive its current status, interpret them and then act in the future to provide personalized services. The objective is to improve the way operators address the network today by exploiting intelligent control decisions procedures.

It is important to note that TALs cannot be optimized without a previous assignment of cells into TAs. As previously indicated, we propose the use of a pre-optimization task to find the most suitable number of cells per TA based on k-means clustering.

From the previous subsection, we can observe that only the approaches presented in [22], [25], [26], [29] consider large-scale scenarios with different UEs mobility patterns over time and velocities. However, none of them consider a perfect classifier technique to assign TALs in a per UE manner. This allows to introduce self-awareness into the network by learning UE's behaviours.

In summary, the proposed strategy is a novel holistic approach composed of two main stages:

1) Per User Equipment (UE) pattern identification and classification.
2) Tracking area lists optimization.

The first stage collects, analyses and exploits operator's data to perform pattern identifications to decide about the type of each UE. This results in a predicted movement pattern that is fed into the second stage. Then, the TAL allocation is optimized according to the learned model. Depending on whether the UE can be associated with a predicted route at the time of TAL allocation, we applied single or multi-objective strategy to group TAs into TALs. Multi-objective strategy is applied for UEs not having a predictable route. These UEs are optimized in an aggregated manner. For UEs having a fixed or permanent route, single objective strategy is applied, so the optimization is applied in a per UE manner. If the optimization was multi-objective, that would mean having a Pareto front *per UE*, which is not practical. Then, a solution is chosen for each type of UE, in different parts of the network and depending on operator needs. Each process is depicted in Figure 1, and described in the following sections.

3 UE Pattern characterization and classification

The weight of designing a TAL for each UE requires to learn mobility patterns for each UE and compute optimal lists. Nevertheless, such tasks are computed offline and have already been considered in the context of 3GPP [30]. With the upcoming of 5G, one of the main targets is to provide personalized services to customers and the application of artificial intelligence to the wireless area seems to be the only path. The mobility pattern concept has been specified and can be used to predict UE's mobility pattern and their associated TALs or TAs depending on the time of the day and geographical position. Thanks to emergence of distributed machine approaches, such complexity can be currently handled. Indeed, operators and specialized companies are putting important research efforts on machine learning applied to the wireless area.

Two features are required to characterize a UE pattern and include its behaviour in the TAL optimization task. The first one is the volume of incoming traffic which directly influences the number of paging messages generated by the presence of that UE in a certain TAL and the second is the level of mobility and route of UEs.

Because behaviour patterns can be highly variable, a reliable discrimination between activities must take several sources of evidence into account. Currently, operators may perform such characterization from many sources. The LTE MME may keep track of UE movement events using TAL updates, including both the updates due to movement and periodic updates. This simply implies a sampled and rough analysis of the route. Looking at more precise characterizations, operators can also get cell identifiers when the UE is actively connected to the network. Nowadays, a large percentage of UEs switch to connected mode very often due to frequent updates of background applications such as e-mail or social networks. Even if that is not the case, Call Data Records (CDRs) provide very useful information to operators [31].

Nevertheless, information about mobility is becoming more and more detailed and most users can check their full location history timeline stored in services such as those from Google. Information about origin and destiny, means of transportation, route preferences, mean required time, etc. can be data mined and combined with incoming data traffic at the mobile network. Sources of information both from the operator itself or external services do allow a precise characterization of mobility and incoming data traffic. In this context, one of the more delicate issues to leverage big data analytics is a proper management of users privacy and security.

Following the previous ideas, we indeed take advantage of Google Maps API [23] to generate real UE mobility traces

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

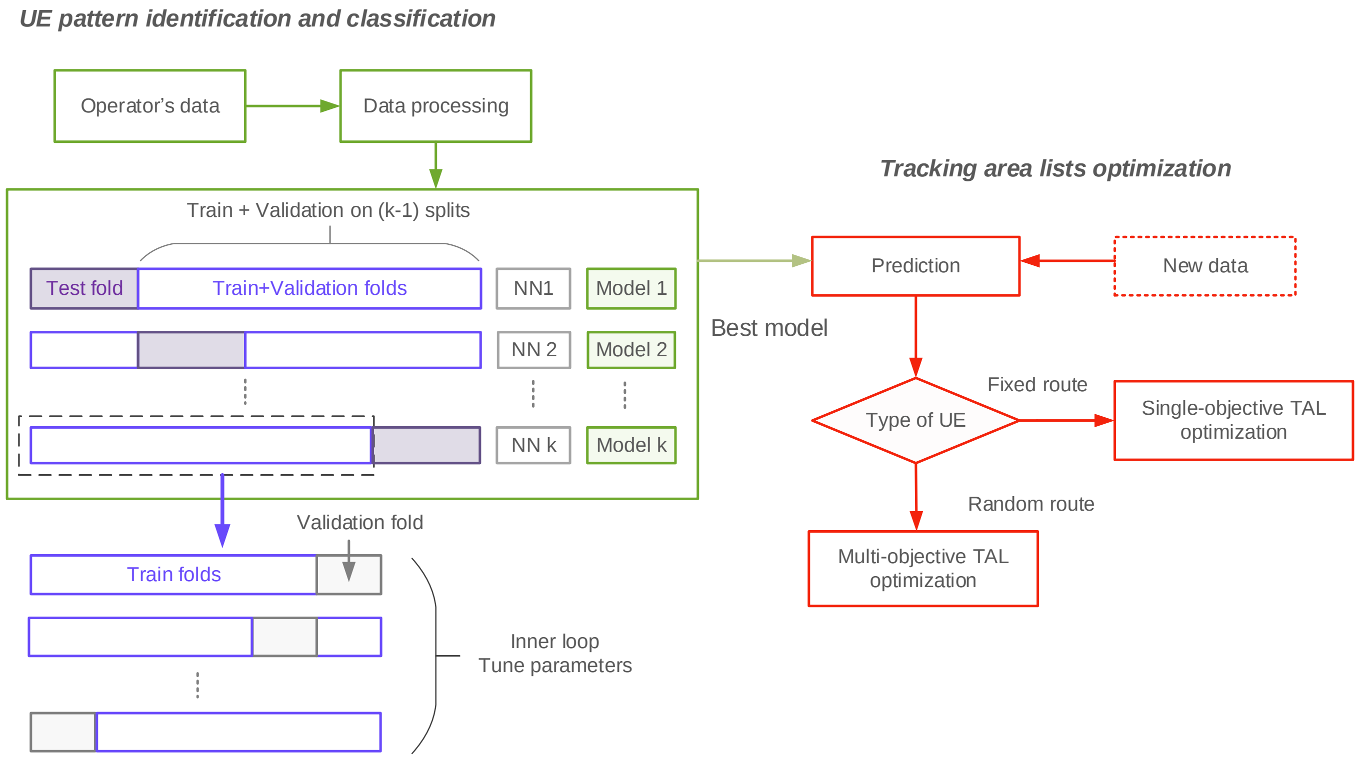


Figure 1: Holistic approach

over an arbitrary period of time and examine a variety of patterns and UE behaviours. Section 5 provides more information in this respect.

A certain percentage of UEs may not follow repetitive routes, or at least not with enough frequency to establish a pattern to be included in the TAL optimization. However, even *random* UEs may be analyzed in an aggregated manner so that TALs are optimized for the general movement trends, for example hotspot generation at certain hours in commercial areas. Such aggregated movement will also determine the best allocation of TAs to TALs, both by determining their size and geographic distribution.

Without loss of generality, we have considered a taxonomy of four possible types of UEs:

- **FHT**: UEs with an identified **F**ixed or permanent route and having **H**igh incoming **T**raffic at a certain time.
- **FLT**: UEs with an identified **F**ixed route but having **L**ow **T**raffic.
- **RHT**: UEs that cannot be identified as following a repetitive route in that time, i.e. having a **R**andom movement, and having **H**igh incoming **T**raffic.
- **RLT**: UEs showing **R**andom movement and **L**ow **T**raffic.

In this process, we exploited the data generated within the network itself to find mobility patterns of UEs through machine learning techniques. A back-propagation NN [32] is proposed for this purpose. The NN consists of two layers plus its corresponding input layer, that receives the data from which predictions are to be made. The output layer is in charge of providing predictions and feedback training and an intermediate hidden layer stores the characteristics of the learned patterns. Figure 2 shows the structure of

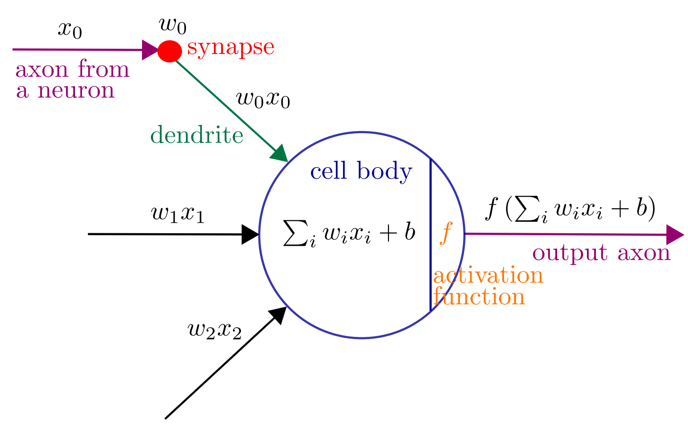


Figure 2: Structure of one neuron in the network.

one of the neurons in the network. Regarding activation functions, the logistic Sigmoid function is used in the hidden layer ($f(x) = \frac{1}{1-e^{-x}}$) and a linear identity function ($f(x) = x$) is used in the output one [33]. The network is trained using the gradient descent algorithm, which is widely used in back-propagation NNs to minimize the error between outputs and targets. In machine learning, a fundamental issue is to estimate the generalization error of a model after being trained. Partitions between training and testing sets must be chosen so that the NN is not overfit. In that case, the NN would be very well adjusted to the particular training data but would fail with new unseen data. We have strengthened the network against over-fitting by performing the K-fold cross validation procedure, as shown in Figure 1 and Algorithm 1.

The technique performs K rounds of training plus vali-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

Algorithm 1: Model evaluation

input : X, Y: data set (mobility and signaling UE's characteristics)

output : model* : built training model

// Split X, Y into K folds random
1 **for** $k = 1 : K$ **do**
2 Xtest,Ytest $\leftarrow$ fold k for test;
3 Xtrainval,Ytrainval $\leftarrow$ remaining folds (all except k);
// Split train+validation into K-1 folds random
4 **for** $k2 = 1 : K - 1$ **do**
5 Xval,Yval $\leftarrow$ fold k2 for validation;
6 Xtrain,Ytrain $\leftarrow$ remaining folds (all except k2);
// Train with each parameter on Xtrain,Ytrain and evaluate on Xval,Yval, then choose the best hyperparameter setting over the K2 folds
7 best$_{hp}$ $\leftarrow$ `tunner`(Xtrain,Ytrain, Xval,Yval)
8 **end**
// Get a model with the training and validation sets
9 model $\leftarrow$ `NN`(Xtrain+Xval,Ytrain+Yval, best$_{hp}$);
// Predict types of UEs based on the trained network
10 $\hat{y} \leftarrow$ `Predict`(model, Xtest);
// Performance against the actual value
11 performances $\leftarrow$ `NRMSE`(Ytest, $\hat{y}$);
12 **end**
// return the model with the best performance
13 model* $\leftarrow$ `best`(performances, model);

dation and testing over different non-overlapping sets (line 1 in Algorithm 1). It can be observed that a single subset is used to test the model (line 2), and the rest of them (K-1) are used for training and validation to find the best model (line 3). Within each iteration, there is an extra K-1 fold cross validation (inner folds) to optimize the hyper-parameters momentum update and learning rate [34] of each model (lines 4-8). The outer cross validation is used to estimate the error on a test set that the model has never seen before (lines 9-11). Finally, we select the best model with the best performance (line number 13). For our particular case, we set K=10 [35], that means that samples of UEs are created and partitioned into 80% for training, 10% validation and 10% testing. The performance of the NN will also be a matter of evaluation in the results section.

Operator's data is used to bring the capability of predicting the behaviour of each UE, so, enabling mobile network operators to react in real time. That is, we collect, analyze and exploit this data to get experience from it, find patterns of users, and take better decisions. We consider the distribution of different mobility characteristics. In particular, the number of handovers and all the cell IDs that each UE has crossed. Notice that since Release 12, LTE includes a functionality which allows operators to get more information about UE's mobility. The UEs can store the information related to the last 16 visited cells, no matter if they were in idle or connected mode [36]. The UE may send this information when changing its state from idle to connected mode. We also consider the number of paging and TAU overhead generated by that UE. Both mobility and signaling act as input-output pairs to train the predictor, i.e., this kind of data is what the predictor is supposed to be able to produce once trained. The built model is then use to optimize the TALs of each UE. Once we have found the NN weights (w_i) and bias (b) values (see Figure 2), the model will be used with new data to predict the type of UE. So, depending on this outcome, a different strategy is used to allocate the TAL. In particular, we applied single or multi-objective strategy to group TAs into TALs.

4 Tracking Area Lists Optimization

Given a network area with many cells, the objective of this process is to find an optimal TAL design and assignment to UEs so that the network signaling is kept minimized.

The design of TALs requires an underlying existing set of TAs already planned. As it was previously mentioned, it is not uncommon that new strategies for TAL optimization assume that one TA is just composed of one cell. As previously explained, such approach may not be the best in networks implementing current standards. For this reason, prior to optimize the allocation of TAs into TALs a new method to group cells into TAs is proposed.

4.1 Grouping cells into TAs

In order to create TAs, a clustering technique is used. The objective here is to find a basis of TAs to be used by the TAL optimization itself. We propose the use of k-means clustering [27] as a way to obtain meaningful TAs that can be used as a basis for the TAL optimization. k-means aims at grouping cells and creating the optimal Voronoi distribution of TAs. This is a very fast method that allows to generate and compare feasible TA distributions.

The complete process is depicted in Algorithm 2 and its specific lines are referenced along the following explanation. The algorithm is initialized with N_{TA} initial *means* and creates N_{TA} TAs dictated by the resulting Voronoi diagram. Once all the cells are assigned to a TA, the TA centroids (new *means*) are computed from cell coordinates. Next, each cell is re-assigned to the TA whose centroid has the least squared Euclidean distance. The Voronoi distribution is iteratively updated until it converges to the lowest sum squared distance possible $d(\mathbf{T})$. So, the algorithm aims at solving:

$$\arg\min_{\mathbf{T}} \; d(\mathbf{T}) = \sum_{k=1}^{N_{\text{TA}}} \sum_{i \in \mathbf{t}_k} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|^2 \,, \tag{1}$$

where,

- $\mathbf{T} \in \mathcal{B}^{N_{\text{cell}} \times N_{\text{TA}}}$ with $\mathcal{B} = \{0, 1\}$ is a matrix defining the allocations of cells into TAs. In particular, $t_{ik} = 1$ if cell i belongs to TA k, and 0 otherwise.
- N_{TA} represents the number of tracking areas.
- $\mathbf{t}_k = \{i : t_{ik} = 1\}$ is the set of cell indices in TA k.
- $\mathbf{x}_i \in \mathbb{R}^2$ are the coordinates of cell i.
- $\boldsymbol{\mu}_k = 1/|\mathbf{t}_k| \sum_{i \in \mathbf{t}_k} \mathbf{x}_i$ is the centroid of TA k.

The algorithm sweeps among all the possible numbers of TAs (line 2). For each case, it is executed in parallel for 100 different initializations (line 3), therefore obtaining 100 different solutions and choosing the best one. The initial N_{TA} centroids are chosen following the k-means++ method [37] (line 4) just before the previously described k-means procedure (line 5). At the end, the algorithm chooses the N_{TA} and $\mathbf{T}$ matrix that minimize signaling cost ($N^\star_{\text{TA}}, \mathbf{T}^\star$) (lines 15-18). In particular, for each TA configuration, an objective

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

function is computed representing the signaling overhead in the whole network. That is to say, the signaling that would be needed if such TAs were directly planned in the network, without no subsequent TAL optimization. Given this, the contribution of TAUs to the network signaling overhead is (lines 11-12):

$$TAU_{\text{cost}}(\mathbf{T}) = \sum_{i=1}^{N_{\text{cell}}} \sum_{j=1}^{N_{\text{cell}}} \phi_{ij} \times m_{ij}, \tag{2}$$

where,

- N_{cell} is the number of cells in the network.
- ϕ_{ij} are the elements of matrix $\Phi_{\mathbf{T}} \in \mathcal{B}^{N_{\text{cell}} \times N_{\text{cell}}}$ generated from $\mathbf{T}$. In particular, $\phi_{ij} = 1$ if cell i and cell j belong to different TAs, and 0 otherwise (line 11).
- m_{ij} are the elements of matrix $\mathbf{M} \in \mathbb{N}^{N_{\text{cell}} \times N_{\text{cell}}}$. It represents the number of UEs moving from cell i to cell j during the evaluation time. When $i = j$, it represents the number of UEs staying in cell i. Note that in this case, the contribution of such UEs to TAU_{cost} would be zero. There would be no change in cell, and so there would be no possible change in TA ($\phi_{ii} = 0$).

On the other hand, paging overhead is given by (line 13):

$$P_{\text{cost}}(\mathbf{T}) = \sum_{k=1}^{N_{\text{TA}}} \left(\mathbf{c} \cdot \mathbf{T} \cdot \sum_{i=1}^{N_{\text{cell}}} t_{ik} \right), \tag{3}$$

where,

- $\mathbf{c} \in \mathbb{N}^{N_{\text{cell}}}$ is a vector such as c_i contains the number of incoming connections in cell i, computed as the rate of incoming connections of each UE during the observation time (connections/h) multiplied by the total time cell i appears in the UE's route.
- Thus, $\mathbf{c} \cdot \mathbf{T} \in \mathbb{N}^{N_{\text{TA}}}$ is a vector containing the aggregation of incoming connections in the cells of each TA.
- $\sum_{i=1}^{N_{\text{cell}}} t_{ik} \in \mathbb{N}^{N_{\text{TA}}}$ is the number of cells in each TA.
- Hence, $\left(\mathbf{c} \cdot \mathbf{T} \cdot \sum_{i=1}^{N_{\text{cell}}} t_{ik} \right) \in \mathbb{N}^{N_{\text{TA}}}$ is the number of paging messages received in each TA during the observation time.

The minimization of TAUs is very important from UEs viewpoint since it directly impacts on power consumption and so, battery time. In order to capture this effect, the contribution to the cost of one TAU is β times greater than a paging procedure. It has been considered the commonly accepted value for this penalizing factor, $\beta = 10$ [25]. The final aim is minimizing the total signaling overhead cost (line 14):

$$\begin{aligned} \underset{\mathbf{T}}{\text{minimize}} \quad & f_{\text{cost}}(\mathbf{T}) = \beta \times TAU_{\text{cost}}(\mathbf{T}) + P_{\text{cost}}(\mathbf{T}) \\ \text{subject to} \quad & N_{\text{TA}} \leq N_{\text{cell}}, \\ & 1 \leq \sum_{i=1}^{N_{\text{cell}}} t_{ik} \leq N_{\text{cell}}, \quad \forall k, \\ & \sum_{k=1}^{N_{\text{TA}}} t_{ik} = 1, \quad \forall i. \end{aligned}$$

An additional and obvious restriction is that all TAs must have a closed boundary.

Algorithm 2: Clustering algorithm for allocation of cells into TAs

input : $\{\beta, N_{\text{cell}}\}$
$\mathbf{x} \in \mathbb{R}^{N_{\text{cell}} \times 2}$: coordinates of all cells
$\mathcal{R}$: set of routes of all UEs
$\mathbf{c} \in N_{\text{cell}}$: incoming connections in each cell

output : $\{\mathbf{T}^\star, f^\star, N_{\text{TA}}^\star\}$

// Number of UEs crossing neighbour cells
1 $\mathbf{M} \leftarrow$ NumCross($\mathcal{R}, \mathbf{x}$);
2 **for** $n = 1 : N_{\text{cell}}$ **do**
// 100 solutions are evaluated in parallel
3 **for** $r = 1 : 100$ **do**
// Determine 1st set of centroids with k-means++
4 $\boldsymbol{\mu} \leftarrow$k-means++($\mathbf{x}, n$);
// Unsupervised clustering with k-means
5 $\{\mathbf{T}, d(\mathbf{T})\} \leftarrow$k-means($n, \mathbf{x}, \boldsymbol{\mu}$);
// Save best solution
6 **if** $r = 1$ **then**
7 $\{\mathbf{T}', d(\mathbf{T}')\} \leftarrow \{\mathbf{T}, d(\mathbf{T})\}$;
8 **else if** $d(\mathbf{T}) < d(\mathbf{T}')$ **then**
9 $\{\mathbf{T}', d(\mathbf{T}')\} \leftarrow \{\mathbf{T}, d(\mathbf{T})\}$;
10 **end**
11 $\Phi_{\mathbf{T}} \leftarrow$ EvalSameTA($\mathbf{T}'$);
12 $TAU \leftarrow$ TAUCost($\Phi_{\mathbf{T}}, \mathbf{M}$); // Compute TAU cost
13 $P \leftarrow$ PagCost($\mathbf{T}', \mathbf{c}, \mathcal{R}$); // Compute paging cost
14 $f \leftarrow$ SigCost(TAU, β, P); // Compute signaling cost
// Save solution with minimum cost
15 **if** $n = 1$ **then**
16 $\{\mathbf{T}^\star, f^\star, N_{\text{TA}}^\star\} \leftarrow \{\mathbf{T}', f, n)\}$;
17 **else if** $f < f^\star$ **then**
18 $\{\mathbf{T}^\star, f^\star, N_{\text{TA}}^\star\} \leftarrow \{\mathbf{T}', f, n)\}$;
19 **end**

Regarding the complexity of the strategies, k-means is fast in practice, its worst-case running time is exponential in the number of data points. Therefore, the run time of the clustering is $O(knT)$ where k is the number of clusters, n is the number of points and T the number of iterations. For NN training complexity, we need to process all the weights and all the neurons. Then, the complexity for learning m examples where each gets repeated ϵ times, is $O(Wm\epsilon)$, where W is the number of weights. With respect to the cross validation, since we are using K-fold cross-validation (outer loop) and K-1 fold cross validation (inner loop), it is required a quadratic number of models to be trained to the number of K folds. Indeed, this option is computationally more expensive than the normal cross validation, but it selects the set of hyper-parameters that seem to provide the best estimated performance.

4.2 Grouping TAs into TALs

Once the cells have been assigned into TAs, we propose a: 1) single-objective and 2) multi-objective strategy to group TAs into TALs. One or the other approach is used depending on whether the UE can be associated with a predicted route at the time of TAL allocation (see Figure 1).

1) The single-objective strategy is applied to UEs that were identified as having a fixed or permanent route. In this case, the UE is allocated a TAL that is optimized in an ad-hoc manner for its specific predicted movement pattern. That is to say, the optimization is applied in a per UE manner. Thus, for FHT and FLT UEs, we propose an evolutionary (genetic) method but following a single objective approach that will yield a solution.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

In particular, we propose the use of a classic Genetic Algorithm (GA) [38]. In GAs, a solution is called an individual, in our case, it is a vector **s** containing the allocations of TAs into TALs. We create a $N^{\star}_{\text{TA}}$-by-1 vector containing TAL indices of each TA. We define a set of lower and upper bounds on the TAL indices, so that a solution is found in the range $1 \leq \mathbf{s} \leq N^{\star}_{\text{TA}}$.

The complete process is summarized in Algorithm 3 and explained in the next paragraphs. The specific lines of the algorithm are referenced along the text.

Algorithm 3: Genetic algorithm operation

input : $|\Sigma|$: population size
π_c: crossover probability
π_m: mutation probability
N_{elite}: number of elitist solutions

output : $\mathbf{s}^{\star}$: solution having minimum signaling

// Random initial set of valid solutions
1 $\Sigma = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|\Sigma|}\} \leftarrow$ `Initialize`$(|\Sigma|)$;
// Compute costs for all solutions
2 $\mathbf{f} \leftarrow$ `SigCost`$(\mathbf{s})$;
// Copy elitist solutions to future population Σ'
3 $\{\mathbf{s}'_1, \mathbf{s}'_2, \ldots, \mathbf{s}'_{N_{\text{elite}}}\} \leftarrow$ `BestSolutions`$(\mathbf{s}, N_{\text{elite}})$;
4 **while** *elitist solutions do not converge* **do**
5   **for** $n = N_{\text{elite}} : 2 : |\Sigma|$ **do**
    // Select parent solutions
6     $\{\mathbf{s}_{\text{p1}}, \mathbf{s}_{\text{p2}}\} \leftarrow$ `Roulette`$(\Sigma, \mathbf{f})$;
    // Crossover parents with probability π_c
7     $\{\mathbf{s}_{\text{child1}}, \mathbf{s}_{\text{child2}}\} \leftarrow$ `CrossOver`$(\mathbf{s}_{\text{p1}}, \mathbf{s}_{\text{p2}}, \pi_c)$;
    // Mutate genes with probability π_m
8     $\{\mathbf{s}'_n, \mathbf{s}'_{n+1}\} \leftarrow$ `Mutation`$(\mathbf{s}_{\text{child1}}, \mathbf{s}_{\text{child2}}, \pi_m)$;
9   **end**
  // Offspring becomes current population
10   $\Sigma = \Sigma'$;
  // Compute costs for all solutions
11   $\mathbf{f} \leftarrow$ `SigCost`$(\mathbf{s})$;
  // Copy elitist solutions to future population Σ'
12   $\{\mathbf{s}'_1, \mathbf{s}'_2, \ldots, \mathbf{s}'_{N_{\text{elite}}}\} \leftarrow$ `BestSolutions`$(\mathbf{s}, N_{\text{elite}})$;
13 **end**
14 $\mathbf{s}^{\star} \leftarrow \mathbf{s}_1$

The algorithm manages a collection of individuals or *population* which is randomly initialized with valid solutions (line 1-2). Then, the GA performs an intelligent search aiming at the solution having minimum cost [38]. This is done by using the three basic genetic operators:

a) Selection: Pairs of parent solutions are iteratively selected and combined to generate a complete new population or offspring. The selection procedure in this work follows the Roulette mechanism combined with elitism [39]. The concept of elitism establishes that the best solutions are directly copied to the new population, this prevents losing them and increases the performance of GA rapidly. In the present work two elitist solutions are considered (see lines 2-3 for the initial population and lines 11-12 for the rest). This number must be chosen small enough to promote solution diversity and re-combinations and thus a thorough exploration of the space of solutions. On the other hand, the Roulette criteria establishes that each solution is chosen with a probability proportional to its cost, in this case inversely proportional because minimization is desired (lines 5-6).
b) Crossover: Once a pair of solutions is selected, their combination (crossover) is done with probability π_c (line 7). If there is no crossover, parent solutions are directly transferred to the new population. Crossover can be performed in multiple ways [39]. In this case, since each solution is an array with the allocation of TAs into TALs, the single point crossover strategy is suitable. A dividing point is chosen randomly and the first half of each parent solution is combined with the second half, thus originating two new children solutions. The sensitivity of cost to π_c variations is analyzed in Section 6.2.
c) Mutation: This operator introduces random modifications into the new children solutions. This is required since initial solutions are unlikely to contain all the information needed to find the optimum via crossover operations alone. The idea is to maintain diversity within the population and inhibit premature convergence. Each element in the solution vector is mutated with probability π_m (line 8). It is recommended that this parameters is kept very low, because the higher π_m is, the closer the algorithm behaves as a random search. We have followed general recommendations and $\pi_m = 0.01$ [40].

2) The multi-objective strategy is applied to UEs not having a predictable route at the time when the TAL must be allocated, RHT and RLT: Random high and low traffic, in the taxonomy previously described. These UEs are optimized in an aggregated manner and so a Pareto front can be obtained for each type. Thus, for RHT and RLT UEs, we focus on multi-objective evolutionary (genetic) algorithms which aim at finding the set of solutions that form the Pareto front. A solution is Pareto optimal if it is a non-dominated one, meaning it cannot improve the performance of an individual cost function without degrading the other one. Note that each solution represents an assignment of TAs into TALs.

In our case, the cost functions represent the TAU and paging signaling volume, but different to equations 2 and 3, now they are aggregated in the TAs conforming every TAL, which is different for each UE.

The rationale of using a multi-objective approach is that it proposes a set of solutions representing the trade-off between both cost functions. That is to say, when visualizing the Pareto front, the sensitivity of one objective with respect to the other is clearer. For example, if TAU_{cost} gets much worse from a certain P_{cost}, operators can decide to stay with a solution just before entering in the area where the Pareto Front will have a high slope. This way, the operator may choose one or another solution depending on their priorities. Note that it is important to update the TAL designs over time in order to adapt to changes in UE location and mobility patterns. If the mechanisms to gather information from the network were specific for such purpose, that extra signaling overhead should be considered to assess the real cost of network reconfiguration [1], [41]. However, as previously explained, data availability is nowadays huge. It seems not reasonable to assume that specific

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

data collection is needed.

Given the previous paragraphs, and starting with the multi-objective approach, we shall consider both objectives independently TAU_{cost} and P_{cost}. Note that multi-objective optimization involves several conflicting objectives simultaneously. Hence, an optimal decision can be taken from Pareto optimal solutions in the presence of trade-offs. As a consequence, the cost functions are not combined but treated separately. So the optimization formulation is given by,

$$\begin{aligned} \underset{\mathbf{s}}{\text{minimize}} \quad & TAU_{\text{cost}}(\mathbf{s}), P_{\text{cost}}(\mathbf{s}) \\ \text{subject to} \quad & \mathbf{s} \in \mathbf{S} \end{aligned}$$

Where $\mathbf{S}$ is the solution space created by all valid solutions, $\mathbf{s}$. The only restriction to create the TALs is that they have closed boundaries. Note that there is no restrictions in the number of TALs because there are more potential TALs than TAs. This is due to the multiple potential sizes and combinations that can be created.

Among the multi-objective formulations available in the literature, we focus on Non-dominated Sorting in Genetic Algorithms II (NSGA-II) initially proposed in [42] as an improved version of the original NSGA algorithm [43]. The three main features of NSGA-II are the use of non-dominated sorting, meaning that candidate solutions are iteratively sorted according to their Pareto dominance ranking. Second, the definition of crowding distances that emphasize on less crowded solutions to maintain the diversity and spread of the Pareto front. And third, the use of elitism, indeed, as it can be inferred from its name, NSGA-II takes advantage of GA operators already described above. NSGA-II is a well-known algorithm that requires the tuning of a low number of parameters to consistently provide good results, a formal treatment can be found in [44]. There has been a lot of research in multi-objective algorithms in the last decade, but NSGA-II still stands out as one of its best exponents.

5 Scenario

We consider the city of Vienna, within an area of $2.5 \cdot 10^4 \times 2 \cdot 10^4$ km^2. The network consists of 207 base stations. The number of UEs is computed considering the three main operators plus the Mobile Virtual Network Operators (MVNOs) that contribute altogether as as a fourth one. The capital city area is also considered (414.65 km^2) and a subscriber density calculated as the inhabitants density (4326 inhabitants/km^2) multiplied by the service penetration factor. This is assumed to be equal to the LTE coverage, expected to be 85% in 2015 [45]. So, we estimate the number of clients per operator υ:

$$\upsilon = \frac{\textit{area}(\text{km}^2) \times \textit{density}(\text{subscribers/km}^2)}{\textit{number\_of\_operators}}.$$

As a result, the estimated number of clients per operator is 500,000. In subsequent simulations we consider a sample of more than 1% of the clients of one operator, which corresponds to N_{UE} =5410 UEs. We focus on modelling rush

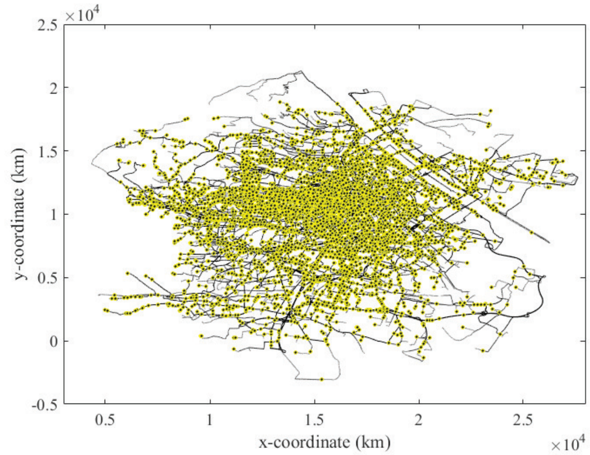


Figure 3: UE's distribution

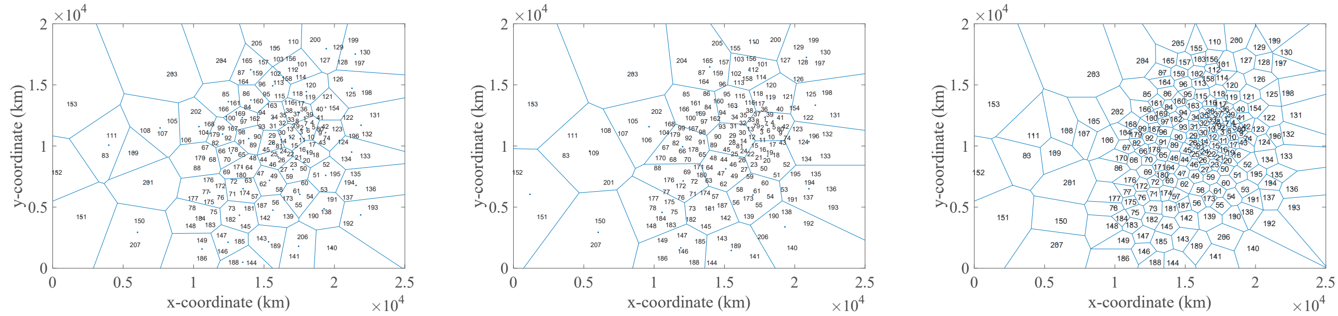
hour traffic conditions, including the morning rush-to-work and evening get-off-from-work traffic in the city of Vienna. Mobility is generated from realistic traces downloaded from the Google Maps APIs. So, we generate mobility based on real-world streets, highways, etc. by queering the Google Maps Places service to find start and endpoints. This API allows us to get information, such as, the location history time-line, which stores the routes a user has travelled with time information. Indeed, an exact movement and position of UEs is very important for providing meaningful evaluation results of TAL assignment.

Ns3 has been used to interact with Google Maps API and to convert downloaded information into way-points. Ns3 is an open simulation environment for networking research. It accounts for an important amount of models that are widely validated and maintained [46]. Hence, the ns3 way-point mobility model has been used to generate realistic UE mobility, based on such real world locations and as described in [47]. Figure 3, shows the visualization of the UEs distribution, and the mobility trace generated automatically. We consider both, vehicular and pedestrian UEs. The UE speed is given by the different conditions (traffic, road configuration, etc). The mobility traces generated provide the Cartesian coordinates and the velocity vector in specific waypoints at a given time.

6 Analysis and performance evaluation

We consider that for UEs with low traffic, the average connection arrival rate per hour is $\lambda = 1$, whereas for UEs with high traffic, $\lambda = 50$. Also, three levels of paging volume are defined by means of parameter $\alpha = \{0.05, 0.2, 0.4\}$ which indicates the percentage of UEs being paged. Hence, the worst case is $\alpha = 40\%$ having a load intensity of 50 pages/h. That would represent an average paging intensity of 20 pages/h per subscriber. This value is in line with the data considered in other investigations [48] or real study cases [49]. It is important to note that with the widespread use of smartphones and push messages, paging volumes have increased significantly [50]. So, scenarios with high paging intensity are specially interesting to analyze.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing



(a) (b) (c)

Figure 4: Figure 4a represents the distribution for the optimum number of TAs found ($N^{\star}_{\text{TA}} = 56$), figure 4b represents a solution in which N_{TA} is a 50% lower than $N^{\star}_{\text{TA}}$, whereas figure 4c represents a direct allocation of one cell to one TA

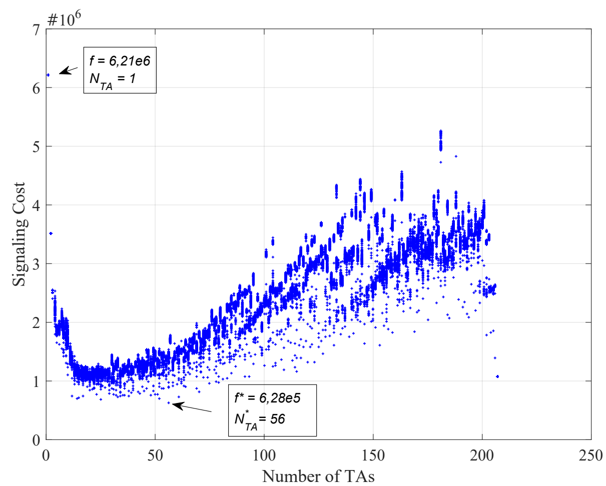


Figure 5: Total signaling cost that has been obtained by the 100 instances of k-means at each N_{TA}.

6.1 Assigning cells into TAs

In this section, we present how the cells have been distributed into TAs, and how the overall signaling overhead increases if cells are not correctly allocated.

Figure 4 represents the allocation of cells into TAs after applying the k-means based procedure. Figure 4a shows the distribution for the optimum number of TAs found ($N^{\star}_{\text{TA}} = 56$). In this case the signalling cost is found to be $f^{\star} = 6.28e5$. Figure 4b is an example of a solution in which N_{TA} is a 50% lower than $N^{\star}_{\text{TA}}$ and Figure 4c corresponds to the direct allocation of one cell to one TA $N_{\text{TA}} = N_{\text{cell}}$. Such situations have a cost of $f = 1.07e6$ and $f = 1.078e6$ respectively so the degradation may be important if cells are not correctly allocated.

Figure 5 represents the signaling cost of each of the 100 solutions obtained by k-means for each N_{TA}. From this figure, we observe that the worst case in terms of the signaling cost is when we consider the allocation of 207 cells into one TA, i.e., $N_{\text{TA}} = 1$. Due to the contribution of the P_{cost}, we also observe that the signaling cost decreases up to $N^{\star}_{\text{TA}} = 56$, which is the optimum number of TAs found. After that, it increases back again, in this case due to the contribution of TAU_{cost}.

6.2 Assigning TAs into TALs

We present the performance of TAL optimization for the different approaches according to the type of UE. In case of FHT and FLT UEs, we optimize the TALs following the GA approach, otherwise we perform multi-objective optimization with NSGA-II. For both cases, the $N^{\star}_{\text{TA}}$ configuration depicted in Figure 4a is considered as an input.

6.2.1 *TAL optimization for FLT and FHT types of UEs*

As we stated earlier, for UEs having a predictable route, the optimization is applied in a per UE manner by following a single objective approach. In this case, the UE is allocated a TAL that is optimized in an ad-hoc manner for its specific predicted movement pattern. In particular, the GA acts over all the TAs in the UE route. The performance of this GA based approach is compared against three benchmarks:

- OneTAL scheme: All the TAs in the UE route are grouped in one single TAL.
- IOTA scheme: Gradient-based method that starts from the OneTAL assignment, iteratively isolates the highest overloaded TA: Once the most overloaded TA has been identified, the TAL is split in that point and signaling overhead is re-calculated for each cut. If the resulting total cost is less than the OneTAL scheme, we keep IOTA's configuration, otherwise we maintain only one TAL. The process is repeated until no improvement is obtained.
- $N_{\text{TA}} = N_{\text{TAL}}$ scheme, which corresponds to the direct allocation of one TA to one TAL.

The performance among these approaches can be appreciated in Figures 6 and 7. These figures represent the contribution of FHT and FLT type of UEs to the total signaling overhead in the network. It can be observed that for UEs having low traffic (Figure 6) the worst allocation is to perform a direct assignment of TAs into TALs. This is because in such conditions, the total signaling is mainly conditioned by TAUs. On the other hand, for UEs having high incoming traffic (Figure 7), the worse approach would be allocating one single TAL to the route they follow, since in this case paging is the limiting factor. Also, the $N_{\text{TA}} = N_{\text{TAL}}$ scheme becomes the best among the three benchmarks but still having margin for improvement since it neglects TAUs impact. IOTA is able to improve oneTAL particularly for HT UEs, but performs worse than the $N_{\text{TA}} = N_{\text{TAL}}$ scheme. This

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

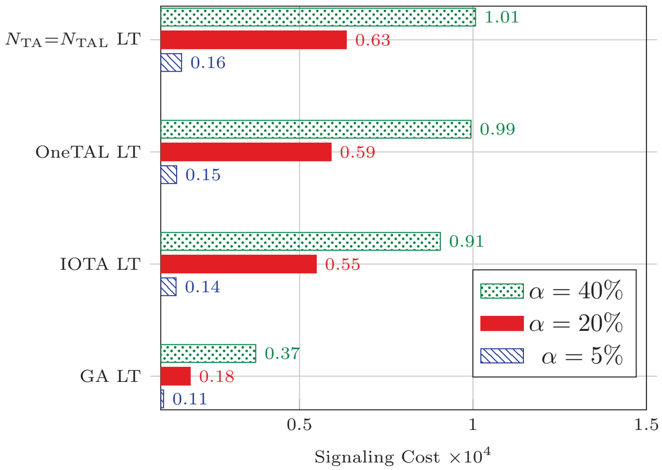


Figure 6: Total signaling cost after TAL optimization for different number of UEs being paged and belonging to group FLT

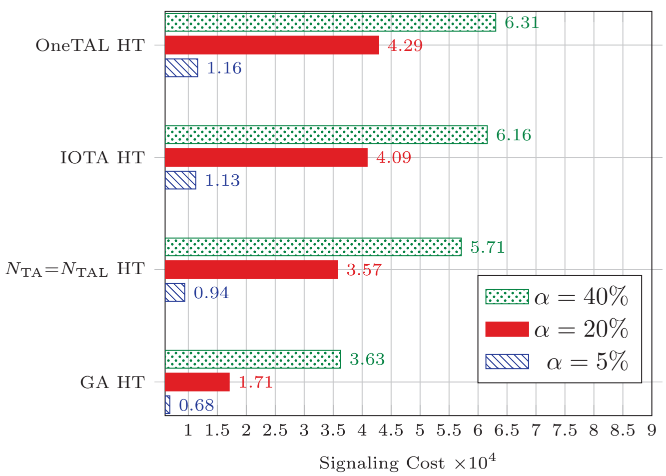


Figure 7: Total signaling cost after TAL optimization for different number of fixed UEs being paged and belonging to group FHT

is because it is gradient based and gets trapped in a local minimum with high probability. For low traffic conditions IOTA provides almost the same results as OneTAL. The reason is that IOTA keeps the original single TAL in most cases.

It is clear that a different treatment is required, a strategy that is able to perform well no matter the traffic conditions of the UE. From both figures, we observe that GA provides the best results with signaling reductions of up to 67% and 52% with respect to the second best case for LT and HT UEs respectively. The reason behind this is that GA performs as a more intelligent search, accepting worse solutions with a certain probability so that other areas in the space of solutions can be explored. Gains are always maximum for an intermediate number of UEs being paged and are lower for the extreme cases (5 and 40%), as shown in Table 2. Finally, it is obvious that HT users cause more cost to the network than LT users. On the other hand, the price to pay when using the GA approach is the need for a correct tuning

Table 2: GA gains with respect to second best scheme

$\alpha \longrightarrow$	5%	20%	40%
FLT	21.4%	67.2%	59.3%
FHT	27.6%	52.1%	36.4%

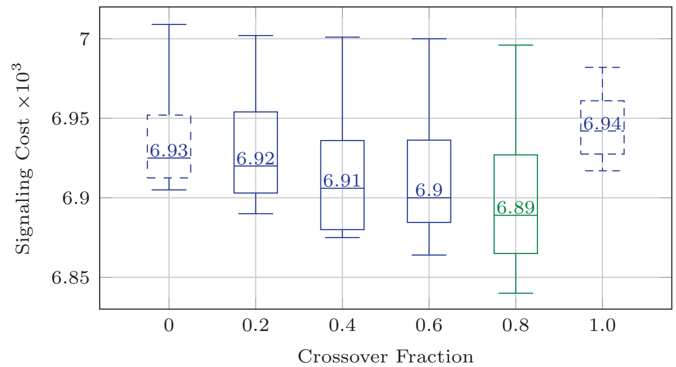


Figure 8: Evolution of total signaling cost for different values of crossover probability. FHT UEs case with $\alpha = 5\%$

of its internal parameters. Figure 8 shows the evolution of cost as a function of crossover probability for the FHT case with $\alpha = 5\%$. The plot represents the mean values (also with numbers), standard deviations (square limits) and extremes (line limits) over 50 generations. We can observe that the algorithm is quite robust to wrong adjustments, and the gain is only changed in their second decimal digit for the extreme cases, 0 and 1. We observe that setting the crossover fraction to 0.8 yields the best result.

6.2.2 TAL optimization for RLT and RHT types of UEs

As we stated previously, this type of UE does not have a predictable route, therefore, we cannot assign a specific TAL, since the information regarding their final destination is un-known. Hence, this type of UEs are optimized in an aggregated manner based on NSGA-II multi-objective formulation. Here, we also consider as an input the TA configuration represented in Figure 4a.

The performance of the whole network can be observed in Figures 9 and 10, where a Pareto front is obtained for each type of UE. These figures illustrate the performance achieved by the NSGA-II. They show the set of points along the entire Pareto front for each scenario, i.e., when the percentage of UEs being paged is: 5%, 20% and 40%. These figures also show the performance with respect to two approaches, namely $N_{TA} = N_{TAL}$ and min-max linear programming (MMLP). The first one is a direct allocation of one TA to one TAL, whereas the second one follows the procedure presented in [22], where the authors propose a scheme using overlapped TALs and min-max optimization. This is formulated and computed by means of linear programming and dealing with both congestion mitigation in TAU and congestion mitigation of paging. The authors propose two separate solutions to decrease the TAU_{cost} and P_{cost} via TAL management. One solution tries to minimize the TAU overhead while setting paging as a constraint, and the other one tries to minimize the paging overhead while fixing the TAU overhead as a constraint.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

Table 3: NSGA-II improvement (signalling cost reduction in %) when compared with $N_{\text{TA}} = N_{\text{TAL}}$ and MMLP-TAU/PA

	RLT	RHT
α	**NSGA-II** vs. $N_{\text{TA}} = N_{\text{TAL}}$	**NSGA-II** vs. $N_{\text{TA}} = N_{\text{TAL}}$
5%	68.95%	65.08%
20%	60.26%	51.99%
40%	66.90%	64.03%
α	**NSGA-II vs. MMLP(TAU)**	**NSGA-II vs. MMLP(PA)**
5%	16.83%	52.42%
20%	2.72%	21.50%
40%	0.33%	20.77%

It can be seen that the multi-objective approach is capable of finding a set of solutions representing the tradeoff between TAU and paging. As previously explained, the operator should decide which is the solution that best suits the specific part of a network depending on the MME available capacity. We observe that, for each point on the Pareto front, the TAU_{cost} can be improved only by degrading the P_{cost} and vice versa, i.e., the paging overhead improves only by degrading the TAU overhead. Each Pareto front can be compared with the solution provided by $N_{\text{TA}} = N_{\text{TAL}}$ (squares in the picture). It can be seen how that approach penalizes the TAU cost a lot and still, it is not able to find a non dominated solution (it does not optimizes paging), thus falling on the right of the Pareto front. On the other hand, MMLP is more interesting since it solves better the tradeoff (there is no extreme penalization of one of the costs in most of the cases) but it also falls on the left of each front.

For the sake of completeness, Table 3 summarizes the total signaling cost improvement compared to $N_{\text{TA}} = N_{\text{TAL}}$ and the best MMLP case. In particular, for RLT UEs, the comparison is done taking the MMLP solution that minimizes TAU overhead, whereas for RHT, the comparison is done with the solution that tries to minimize the paging overhead. For the sweep of situations that has been evaluated, it can be observed that improvement with respect to $N_{\text{TA}} = N_{\text{TAL}}$ ranges from 51.99% to 68.95% and this is basically due to a much better management of paging signaling. Gains over MMLP range from 0.33% to 52.42% but in this case the solutions proposed by that technique are more balanced than $N_{\text{TA}} = N_{\text{TAL}}$ and in some cases it is able to find solutions falling in the Pareto Front, with a gain of NSGA-II of just 0.33%.

Therefore, we can conclude about the effective performance of the multi-objective approach for UEs moving with random (or non predictable) routes.

Regarding NSGA-II tuning, Figure 11, illustrates the selection of the population size ($|\Sigma|$) and the number of generations in terms of the signaling cost. Note, that after 900 generations, there is almost no difference in terms of signaling cost. Therefore, we set the number of generations equal to 900 and the $p_{size} = 300$.

Given that GA and NSGA-II perform the best for the situations they were applied, the success of the approach depends on a correct classification of UEs and individual allocation of TALs into UEs, which is analyzed subsequently.

6.3 Assigning TALs to UEs

We consider the mobility and incoming data traffic (inputs), as well as, the type of UEs (output) to build the learning model. Once the NN has been trained to distinguish between different types of UEs, it is used to generate predictions for new data with $\alpha = 20\%$, i.e., once the NN forms a generalization of the input-output relationship, it generates outputs for inputs it was not trained on.

The performance of the predictor can be observed in Table 4, which summarizes the results by the error matrix. In this table, the columns correspond to the predicted groups, and the rows show the real ones. The diagonal shows the

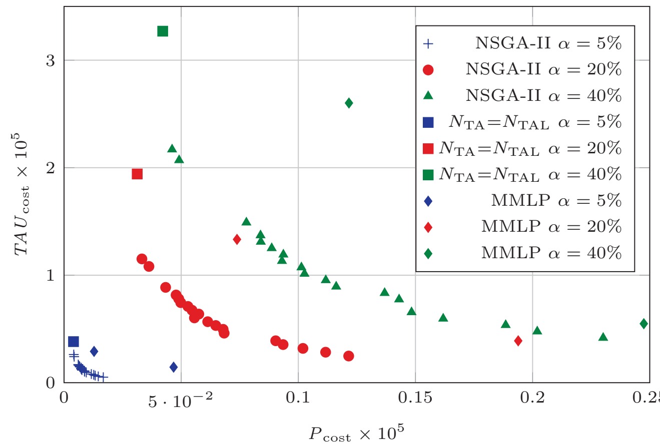


Figure 9: Pareto Front for RLT type of UEs

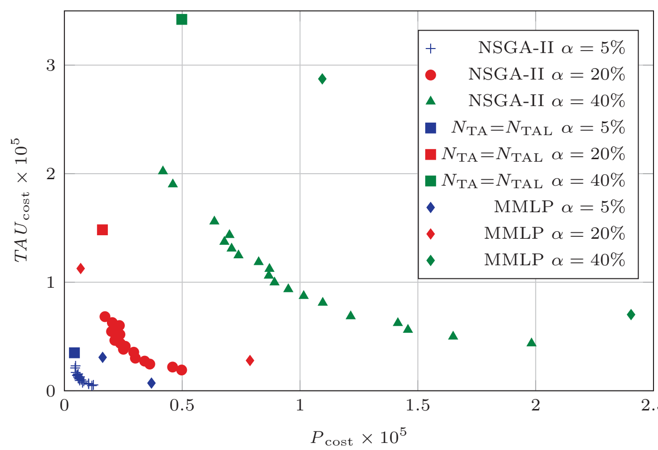


Figure 10: Pareto Front for for RHT type of UEs

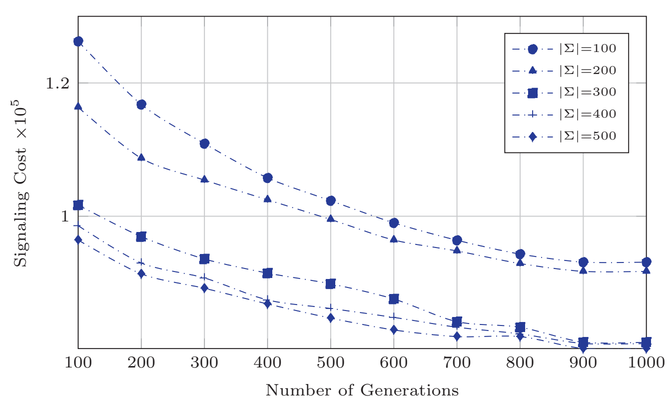


Figure 11: Total signaling cost. Computed after tuning the number of generations and the population size for RHT type

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing

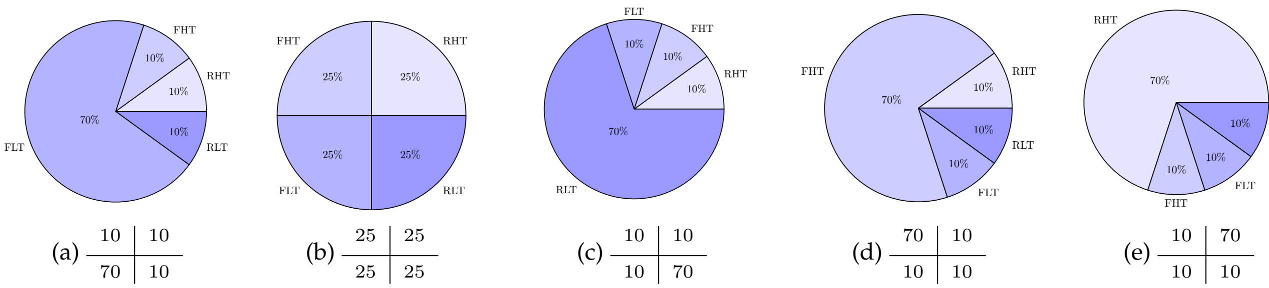


Figure 12: Different combinations of UE types

Table 4: Confusion Matrix

Output	Target RLT	RHT	FLT	FHT	Accuracy
RLT	**267**	0	21	0	92.7%
RHT	0	**271**	0	0	100%
FLT	4	0	**250**	0	98.4%
FHT	0	0	0	**271**	100%
	98.5%	100%	92.3%	100%	**97.7%**

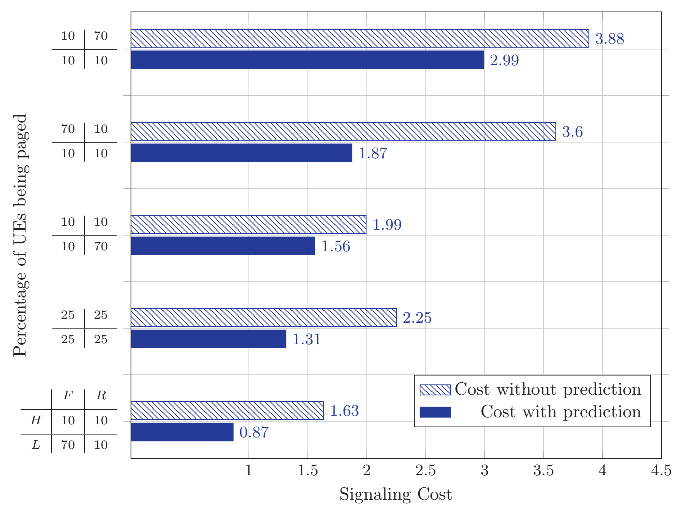


Figure 13: Total signaling cost after TAL optimization for different combinations of type of UEs

number and percentage of correct predictions. That is, 267 of UEs belonging to random UEs with low traffic conditions (RLT) are correctly classified. This corresponds to 98.5% of all RLT UEs. 4 of the RLT type of UEs are incorrectly classified as fixed UEs with low traffic conditions (FLT) and this corresponds to 1.5% of all RLT type of UEs. Similarly, 21 FLT type of UEs are incorrectly classified as random UEs with low traffic conditions (RLT) and this corresponds to 7.7% of this type of UEs. Finally, 271 of the UEs belonging to fixed UEs with high traffic (FHT) and 271 of the UEs belonging to random UEs with high traffic (RHT) are correctly classified. This corresponds to 100% of FHT and RHT type of UEs.

Summarizing, out of group of UEs with RLT predictions, 92.7% are correct and 7.3% are wrong. Out of 271 group of UEs with RHT predictions, 100% are correct. Out of 271 group of UEs with FLT predictions, 98.4% are correct and 1.6% are predicted as group of UEs with RLT predictions. Finally, out of 271 group of UEs with FHT predictions, 100% are correctly classified. Overall, 97.7% of the predictions are correct and 2.3% are wrong classifications.

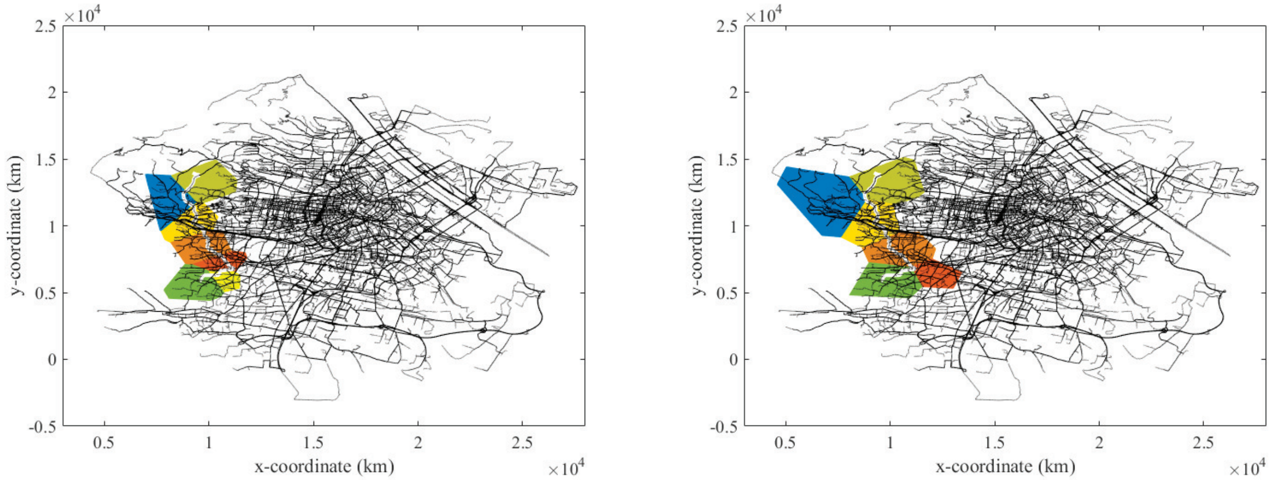
6.4 Performance of complete approach

In this section, we analyze the results obtained when the complete strategy operates in a network in exploitation. In this evaluation, we consider 5410 UEs, where 2705 UEs correspond to FHT and FLT types, and the rest of them correspond to RHT and RLT. It has been analyzed the total signaling overhead associated with paging and TAU in the whole network. This has been done when the TAL assignment considers the UE type prediction in terms of mobility and incoming traffic. Five cases have been considered, they are graphically depicted in Figure 12 and indicated in the vertical axis of Figure 13. It can be understood that the columns of each matrix indicate the kind of traffic conditions, low (L) or high (H), whereas the columns represent the trajectory of each UE, fixed (F) or random (R). For example, for the upper bars, 70% of the UEs have a predictable route and show low traffic conditions (FLT), the rest of UEs are equally distributed into the other three types.

In order to compare the signaling reduction achieved by the proposed mechanism, Figure 13 shows the overhead obtained with and without prediction. The case without prediction treats all UEs as RLT or RHT, there is optimization but only traffic variations are considered and route prediction is not performed. As it can be seen, by predicting UE profiles, the total signaling overhead is reduced drastically. Gains appear in all the situations and they increase with the number of FHT UEs. Thus the case with more outstanding gain is the second one, when 70% of UEs are FHT, and lower gains appear when 70% of UEs is RLT.

Figures 14a and 14b show an example of TAL optimization for one specific UE, with and without prediction. The white line represents the route of the UE, and each box corresponds to one TAL for this particular UE, i.e. each box is composed by a set of TAs, which are composed by a set of cells. On one hand, from Figure 14a, we observe that, if we are able to predict the UE's activity in a spatio-temporal manner, the number of TAs that comprises one TAL follows pretty much the route of the UE. On the other hand, if the prediction is not able to identify a specific route, the TAL optimization based on NSGA-II is used. Then, the region of each TAL covers more than the required to follow the UE's route. This means an increase in overhead due to extra paging.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2915079, IEEE Transactions on Mobile Computing



(a) (b)

Figure 14: TAL optimization. Figure 14a represents the optimized TALs in a per-UE manner with prediction, whereas Figure 14b illustrates the optimized TALs without prediction

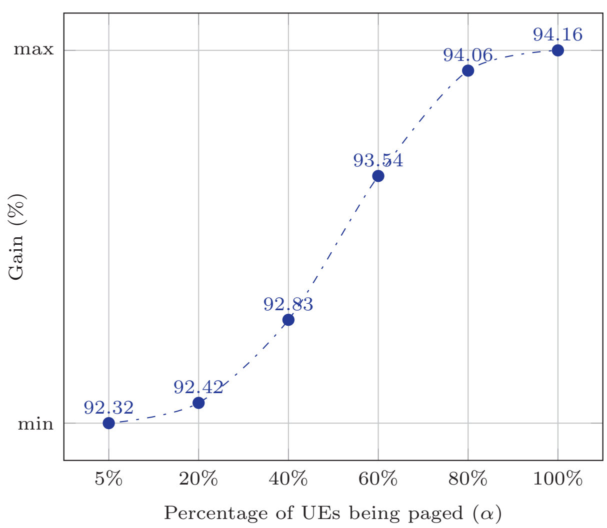


Figure 15: Sensitivity of the prediction gain with the percentage of fixed UEs having high paging traffic

Furthermore, in order to evaluate the sensitivity of the gain as a function of the number of UEs, Figure 15 shows the $Cost_{\mathrm{T}}$ versus the number of UEs providing more gain to the prediction, the FHT type. From this figure, we observe how the gain in the prediction increases as a function of the percentage of UEs being paged. Therefore, mitigating signaling overhead due to profile prediction not only is significant, but also it increases with the number of UEs.

The main advantage of considering this approach, is that it enables the assignment of customized TALs in a per UE manner, and such TALs adapt their sizes in a spatio-temporal manner, i.e., depending on the time of day, or the weekday. As a result, the operator could be able to find the TAL configuration that best meets the real state of the network.

We close this section with a discussion about the impact of specific UEs changing their behaviour and not being correctly identified by the NN. Indeed, it is expected that this situation happens for a certain percentage of UEs. For example, a UE that starts a new route that the NN has not yet learnt. In this case, the UE is predicted to have a random route and so the TAL is allocated following the aggregated UEs pattern in the region. Thus, signaling cost would not be fully minimized and would fall in between the two values in each pair of bars in Figure 13.

Alternatively, if a UE is predicted to have a certain repetitive route but it changes its behaviour, then uplink signaling would be higher for that UE. The most extreme case would happen if only one cell is common between the predicted and the real route. In that case, the UE would perform a TAU as soon as it leaves the current cell. After this event, a new TAL is allocated and so a new evaluation by the NN would happen. At this point, the UE would be identified as having a random route and a general TAL (not route specific) would be assigned, so the impact on the signaling would be very low and localized in time.

For UEs having low traffic but being identified as high traffic, paging cost would be reduced but TAU cost would be increased, and vice versa. In this case, the reaction time would equal the time window in which the traffic intensity is evaluated.

A small percentage of UEs not being correctly identified generates low impact on the network signaling. Indeed, the NN that has been used in this work already makes mistakes of up to 7.3% in certain types of UEs and still the signaling reduction was shown to be very noticeable.

Regarding scalability issues, we must first note that the approach has been applied to a city that accounts for 207 sites, 5410 users, 530,180 routes and an observation time of 3600 seconds. With such configuration, it was required a processing time of 8 hours to compute optimal solutions using core i7 6700 at 3.40 GHz with 4 cores and no parallel programming. When compared to other planning processes (e.g. radio planning), this is a very modest time, and so there is a large margin to increase the data set and complexity of the network. Nevertheless, it is a fact that having a centralized approach may pose scalability problems as larger parts of the network are considered. In this sense, the computation should be distributed. The usual practice based on monolithic data sets and learning processes that generate a unique model might soon be phased out with distributed approaches. In our particular case, this would include not only distributed machine learning, but also parallelization of GA / NSGA [51].

7 Conclusions

In this paper we have presented a holistic approach for a dynamic TAL optimization. The proposed strategy aims at exploiting big data analytics and UE patterns from geospatial time series to generate optimized TALs in a per UE manner. This is done by proposing a different optimization treatment depending on whether the route of the UE can be predicted. Thus, a combination of multi-objective and single-objective optimizations is proposed and evaluated under very realistic conditions and for different traffic situations. The optimization is completed by a pre-optimization work in which the k-means algorithm is proposed to generate the basis allocation of cells into TAs. Results indicate that solutions with better cost than a direct allocation of one cell into one TA can be found, and thus overcoming the limitations of that approach in terms of signaling. Without loss of generality, a taxonomy of four possible types of UEs has been considered and a NN has been used as pattern classifier so that TALs can be applied in a per UE manner. All the mechanisms have been evaluated individually and operating jointly and results demonstrate the ability of the proposed scheme to reduce signaling related to idle mode mobility. The approach is complete and offers to the operator a closed solution.

Competing Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgment

This work was supported by the Spanish National Science Council and ERFD funds under projects TEC2014-60258-C2-2-R and RTI2018-099880-B-C32.

References

[1] M. Toril, S. Luna-Ramírez, and V. Wille, “Automatic replanning of tracking areas in cellular networks,” *IEEE Transactions on Vehicular Technology*, vol. 62, pp. 2005–2013, June 2013.

[2] S. M. Razavi, D. Yuan, “Mitigating mobility signaling congestion in LTE by overlapping tracking area lists,” *14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSMWiM)*, pp. 285–292, 2011.

[3] 3GPP, “Evolved universal terrestrial radio access (E-UTRA) and evolved universal terrestrial radio access network (E-UTRAN); Overall description; Stage 2,” 3GPP, TS 36.300 v13.6.0, 2016.

[4] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, “Big data-driven optimization for mobile networks toward 5g,” *IEEE Network*, vol. 30, no. 1, pp. 44–51, January 2016.

[5] S. Han and C. Lin I and G. Li and S. Wang and Q. Sun, “Big data enabled mobile network design for 5G and beyond,” *IEEE Communications Magazine*, vol. 55, no. 9, pp. 150–157, July 2017.

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

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

[8] S. M. Razavi, “Planning and optimization of tracking areas for long term evolution networks,” Ph.D. dissertation, Linköping University Institute of Technology, 2014.

[9] J. Ferragut , “Traffic and mobility management in large-scale networks of small cells,” Ph.D. dissertation, Universitat Politècnica de Catalunya, 2014.

[10] I. Demirkol, C. Ersoy, M. U. Caglayan, and H. Delic, “Location area planning in cellular networks using simulated annealing,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 1, April 2001, pp. 13–20 vol.1.

[11] M. R. Garey, D. S. Johnson, “Computers and intractability; A guide to the theory of NP completeness,” *W. H. Freeman and Company, New York*, 1979.

[12] Z. A. Dahi, E. Alba, and A. Draa, “A stop-and-start adaptive cellular genetic algorithm for mobility management of GSM-LTE cellular network users,” *Expert Systems with Applications*, vol. 106, pp. 290 – 304, 2018.

[13] S. M. Razavi, D. Yuan, F. Gunnarsson, J. Moe, “Exploiting tracking area list for improving signaling overhead in LTE,” *IEEE Vehicular Technology Conference*, pp. 1–5, 2010.

[14] S. M. Razavi, D. Yuan, F. Gunnarsson, J. Moe, “Dynamic tracking area list configuration and performance evaluation in LTE,” *IEEE GLOBECOM Workshops*, vol. 4, pp. 49–53, 2010.

[15] S.-L. Wu, J.-J. Chen, and W.-C. Chou, “Cell-related location area planning for 4G PCS networks with variable-order markov model,” *Journal of Systems and Software*, vol. 86, no. 10, pp. 2688 – 2699, 2013.

[16] 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, no. 2, pp. 687–708, March 2005.

[17] R. H. Liou, Y. B. Lin, and S. C. Tsai, “An investigation on LTE mobility management,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 1, pp. 166–176, January 2013.

[18] Y. W. Chung, “Adaptive design of tracking area list in LTE,” *International Conference on Wireless and Optical Communications Networks (WOCN)*, pp. 1–5, 2011.

[19] J. Ferragut and J. Mangues-Bafalluy, “A self-organized tracking area list mechanism for large-scale networks of femtocells,” in *2012 IEEE International Conference on Communications (ICC)*, June 2012, pp. 5129–5134.

[20] T. Deng, X. Wang, P. Fan, and K. Li, “Modeling and performance analysis of a tracking-area-list-based location management scheme in LTE networks,” *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 6417–6431, August 2016.

[21] X. Wang, X. Lei, P. Fan, R. Q. Hu, and S. Horng, “Cost analysis of movement-based location management in PCS networks: An embedded markov chain approach,” *IEEE Transactions on Vehicular Technology*, vol. 63, no. 4, pp. 1886–1902, May 2014.

[22] S. M. Razavi, D. Yuan, “Mitigating signaling congestion in LTE location management by overlapping tracking area lists,” *Computer Communications*, vol. 35, pp. 2227–2235, November 2012.

[23] Google Maps API. [Online]. Available: https://developers.google.com/maps/documentation/directions/

[24] E. Grigoreva, J. Xu, and W. Kellerer, “Reducing mobility management signaling for automotive users in LTE advanced,” in *2017 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, June 2017, pp. 1–6.

[25] S. M. Razavi, D. Yuan, F. Gunnarsson, J. Moe, “On dynamic signaling congestion mitigation by overlapping tracking area lists,” *Network and Computer Applications*, vol. 53, pp. 164–72, July 2015.

[26] E. Aqeeli, A. Moubayed and A. Shami, “Dynamic SON-enabled location management in LTE networks,” *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1511–1523, July 2018.

[27] T. Hastie, R. Tibshirani, J. Friedman, *The elements of statistical learning data mining, inference, and prediction*. Springer series in statistic, 2009.

[28] E. Aqeeli, A. Moubayed, and A. Shami, "Towards intelligent LTE mobility management through MME pooling," *IEEE 58th Global Communications Conference*, pp. 1–6, 2015.

[29] M. Bagaa, T. Taleb and A. N. Ksentini, "Efficient tracking area management framework for 5G networks," *IEEE Transactions on Wireless Communications*, vol. 16, pp. 4117–4131, June 2016.

[30] ETSI, "5G; System architecture for the 5G system," 3GPP, TS 23.501 V15.2.0, 2018.

[31] H. Zang and J. Bolot, "Mining call and mobility data to improve paging efficiency in cellular networks," *The 13th annual international conference on Mobile computing and networking, MobiCom 07*, 123-134 2007.

[32] C. Bishop, "Pattern recognition and machine learning," *Springer Science and Business Media, LLC.*, 2006.

[33] B. Karlik, A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, 2011.

[34] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, Springer Berlin Heidelberg*, p. 437–478, 2012.

[35] S. Borra, A. Di Ciaccio, "Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods," *Computational Statistics and Data Analysis*, vol. 54, no. 12, pp. 2976–2989, December 2010.

[36] J. F. Monserrat, M. García, J. J. Olmos and N. Cardona, *3GPP LTE-Advanced y su evolución hacia la 5G móvil*. Marcombo, 2017.

[37] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," *Proc. of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007*, January 2007.

[38] J. H. Holland, "Adaptation in natural and artificial systems," *University of Michigan Press, Ann Arbor*, 1975.

[39] J. C. Spall, *Introduction to stochastic search and optimization*. Wiley-Interscience, 2003.

[40] M. García-Lozano, M.A. Lema, S. Ruiz, F. Minerva, "Metaheuristic procedure to optimize transmission delays in DVB-T single frequency networks," *IEEE Transactions on Broadcasting*, vol. 57, no. 4, December 2011.

[41] E. Grigoreva, J. Xu, and W. Kellerer, "Reducing mobility management signaling for automotive users in LTE advanced," *IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, pp. 1–6, June 2017.

[42] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Computational Intelligence Society*, vol. 6, pp. 182–197, April 2002.

[43] Srinivas, N., & Deb, K., "Multiobjective optimization using non dominated sorting in genetic algorithms," *Journal Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[44] Coello, C. A., Lamont, G. B. and Van Veldhuizen, D. A., *Evolutionary algorithms for solving multi-objective problems*, 2nd ed. Springer: Genetic and Evolutionary Computation Series, 2007.

[45] TeleGeography. [Online]. Available: https://www.telegeography.com/products/commsupdate/articles/2015/05/07/3-austria-expands-lte-coverage-to-85-plans-98-by-summer-2015/

[46] ns-3. Discrete-event network simulator. [Online]. Available: https://www.nsnam.org/

[47] T. Cerqueria. Routes mobility model. [Online]. Available: https://www.nsnam.org/wiki/RoutesMobilityModel

[48] J. Ferragut, J. Mangues-Bafalluy, "A distributed paging mechanism over the X2 interface for all-wireless networks of small cells," *Proc. of the 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2014.

[49] M. Ali, "LTE TAL size planning in live networks (improving paging success rate)," *Netmanias Tech-Blog*, December 2016.

[50] Huawei, "Huawei, smartphone ecosystem R&D support team. "Smartphone solutions," *White Paper*, July 2012.

[51] Y. Y. Liu and S. Wang, "A scalable parallel genetic algorithm for the generalized assignment problem," *Parallel Computing*, vol. 46, pp. 98 – 119, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167819114000519

Jessica Moysen Jessica Moysen received her M.Sc degree from the Universidad Carlos III de Madrid in 2011. Further, in 2012 she joined at the Centre Tecnològic Telecomunicacions Catalunya with a grant of the Spanish Ministry of Science and Innovations to pursue a PhD degree. She received her PhD. in Telecommunications Engineering in 2016 from the Universitat Politècnica de Catalunya (UPC). She is currently a researcher at the Wireless Communications and Technologies Research Group of the Signal Theory and Communications Department of the UPC. Her work has been disseminated through research papers to the field of wireless cellular networks, contributing to new concepts for the autonomic management of current and future mobile networks.

Mario García-Lozano Mario Garcia-Lozano received his M.Sc. and Ph.D. in Telecommunications Engineering from the Universitat Politecnica de Catalunya (UPC, Barcelona-TECH), Spain, in 2001 and 2009 respectively. Dr. Garcia-Lozano has more than 15 years of experience in different radio network planning and optimization activities both at the academia and industry. He is currently an associate professor at UPC and his research activities are focused in the field of radio resource management and the application of machine learning and artificial intelligence to wireless networks. He has actively participated in >25 competitive research projects and contracts with the industry. He currently leads the Spanish DEFINE-5G project at UPC. He is recipient of 3 best paper awards and was the advisor of the student team that won the international competition for mobile network planning organized by the company ATDI.