

Self-Tuning Spectral Clustering for Adaptive Tracking Areas Design in 5G Ultra-Dense Networks

Brahim Aamer⁽¹⁾, Hatim Chergui^(1,2), Nouamane Chergui⁽³⁾, Kamel Tourki⁽⁴⁾,
Mustapha Benjillali⁽²⁾, Christos Verikoukis⁽⁵⁾ and Mérouane Debbah⁽⁴⁾

⁽¹⁾ INWI, Casablanca, Morocco

⁽²⁾ Communication Systems Department, INPT, Rabat, Morocco

⁽³⁾ Huawei Technologies, Casablanca, Morocco

⁽⁴⁾ Huawei Tehnologies, Paris Research Center, France

⁽⁵⁾ CTTC, Barcelona, Spain

Contact Email: brahim.aamer@inwi.ma

arXiv:1902.01342v1 [cs.IT] 4 Feb 2019

Abstract—In this paper, we address the issue of automatic tracking areas (TAs) planning in fifth generation (5G) ultra-dense networks (UDNs). By invoking handover (HO) attempts and measurement reports (MRs) statistics of a 4G live network, we first introduce a new kernel function mapping HO attempts, MRs and inter-site distances (ISDs) into the so-called similarity weight. The corresponding matrix is then fed to a self-tuning spectral clustering (STSC) algorithm to automatically define the TAs number and borders. After evaluating its performance in terms of the Q -metric as well as the silhouette score for various kernel parameters, we show that the clustering scheme yields a significant reduction of tracking area updates and average paging requests per TA; optimizing thereby network resources.

Index Terms—5G, self-tuning spectral clustering, tracking area planning.

I. INTRODUCTION

A KEY component in wireless networks is user location management. Such a function is achieved using the concept of tracking area (TA); similarly to location area (LA) in GSM and routing area (RA) in GPRS. To track users, a mobility management entity (MME) records the TA in which each user is registered. We consider TA design of cells managed by a single MME. When a user moves into a new TA, an update message is sent to the MME. This causes a signaling overhead, referred to as the update overhead. A second type of signaling overhead exists in the reverse direction. In order to place a call to a user, MME broadcasts a paging message in all cells of the TA in which the user is currently registered. Having TAs of very small size virtually eliminates paging, but leads to excessive update, whereas very large TAs give the opposite effect [1]. Intuitively, an optimal TA design tends to group new radio node Bs (gNBs) having large numbers of users roaming between them, e.g., gNBs along a road with

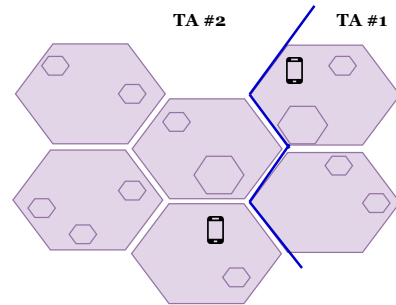


Figure 1. Tracking areas are adopted to manage user location.

much traffic, into the same TA. Nonetheless, the dynamic user behavior and traffic patterns in urban environments make that TAs, initially optimized for certain user statistics (or forecasts), become inaccurate and therefore urge to implement machine learning-driven adaptive TA design algorithms as part of the so-called self-organizing networks (SON) framework [2].

A. Related Work

In [3], the authors presented a re-optimization approach for revising a given TA design, which translates into a NP -hard optimization problem solved via repeated local search. Also, they presented in [4] a “rule of thumb” method to allocate and assign tracking areas lists (TALs) for a network and compare the performance of an optimum conventional TA design with the suggested TAL design for a large scale network in Lisbon, Portugal. The results clearly showed the ability of dynamic TAL in reducing the signaling overhead and maintaining a good performance due to reconfiguration compared to the conventional TA design. On the other hand, the performance of 4G TAL-based location management has been analyzed in [5] using a Markov chain approach. The provided closed-form

$$s_{i,j} = \exp \left[- \left(\alpha \frac{d_{ij}}{\max(d_{ij})} + (1 - \alpha) \left(1 - \beta \frac{a_{ij}}{\max(a_{ij})} - (1 - \beta) \frac{m_{ij}}{\max(m_{ij})} \right) \right) \right] \quad (1)$$

Table I
SAMPLES FROM THE LIVE NETWORK DATASET

Source Site ID	Target Site ID	ISD (Km)	HO Attempts (Two Directions)	MRs Count	Paging Requests for Source Site
45	45	0	30 520	161 194	79 274
45	294	0.428617	7 275	108 443	79 274
56	298	0.66769	1 968	113 536	97 725

formulas highlight the effect of different network parameters on the signaling cost. In particular, the total signaling cost has been shown to be a downward convex function of the radius of a TA in terms of cells. In [6], the authors resorted to K -means clustering algorithm to perform automatic TA planning. This approach ensures the adaptation of the network to the changing user trends. Once a TA re-plan has been triggered, a graph partitioning algorithm is used to build the new TA plan.

B. Contributions

In this paper we investigate the following aspects:

- First, we construct a dataset featuring source/target sites relations, inter-site distances (ISDs), bidirectional HO attempts, events A3 MRs count and paging requests per source site. These features stem from automatic neighbor relation (ANR) statistics retrieved from the SON platform of a large 4G live network.
- We introduce a new Gaussian kernel function that involves the three aforementioned features, and we use it to define an inter-site similarity matrix that is fed to the self-tuning spectral clustering (STSC) algorithm.
- We show that the presented algorithm leads to the reduction of tracking area updates and average paging requests per TA, which optimizes radio network resources while automating TA design.

II. DYNAMIC TA DESIGN ALGORITHM

A. Live Network Dataset

The dataset is retrieved from the performance monitoring platform of a commercial 4G network. It specifies—for each couple of source-target sites—the daily bidirectional HO attempts as well as A3 event measurement reports (MRs). It also includes the paging requests per source site. The sites global positioning system (GPS) coordinates are used to calculate inter-site distances. The total number of measured neighboring relations is $N = 149769$ corresponding to $M = 387$ sites. Note that the adopted quantities, i.e., measurement reports and HO attempts are still viable in 5G.

B. Similarity Matrix

A clustering algorithm operates on the so-called *similarity matrix* whose entries measure the logical correlations between

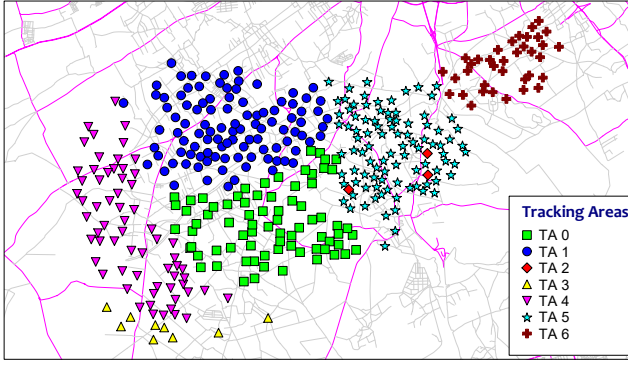
each couple of the dataset samples. In this regard, we adopt a radial basis function (RBF)-based precomputed matrix \mathbf{S} that involves the three features, namely, the ISD, HO attempts and MRs count. The kernel parameter γ is set to 1. As such, the (i, j) th matrix element is given by Eq. (1) on top of this page, where d_{ij} , a_{ij} and m_{ij} stand for the pairwise distance, handover attempts and MRs count from site i to site j . The parameters $\alpha, \beta \in [0, 1]$ are controlling the dependency to each feature. The features are normalized with respect to their maximum. Hence, sites with low ISD or high attempts/MRs present a similarity weight near to 1.

C. Self-Tuning Spectral Clustering

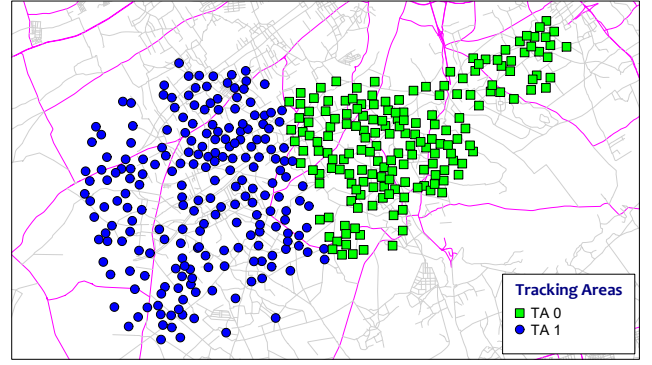
As new cells are added and removed every day, we target a fully adaptive TA design algorithm that can process a high number of observations as well. In this respect, since basic clustering algorithms generally require that the number of clusters be specified in the input, we resort to the well-established self-tuning spectral clustering (STSC) scheme. It was initially introduced in [7], where the authors studied a number of then open issues in spectral clustering:

- Selecting the appropriate scale of analysis, i.e., the parameter γ of the RBF kernel,
- Handling multi-scale data,
- Clustering with irregular background clutter,
- Finding automatically the number of clusters.

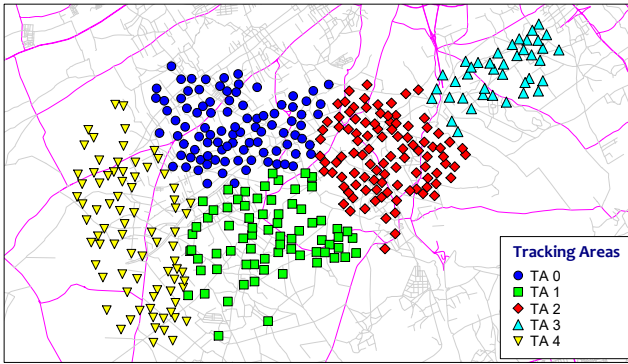
Spectral clustering algorithms [8], by definition, use the eigenvalues and eigenvectors (i.e., the spectrum) of the similarity matrix to cluster a dataset. After normalizing matrix \mathbf{S} , STSC uses the corresponding C largest eigenvectors that are then stored in a matrix \mathbf{X} , where the first column of \mathbf{X} is the biggest eigenvector. One of the tasks of STSC is to get a cost for each possible number of clusters in the dataset to find the most probable number of groups. This computation is done incrementally. One starts with the minimum number of possible clusters c_{\min} by taking the first c_{\min} columns of \mathbf{X} , we rotate them by applying a Givens rotation in an stochastic gradient descent scheme [7] to find the optimal rotation matrix \mathbf{R} . The point of comparison when doing the rotation is the



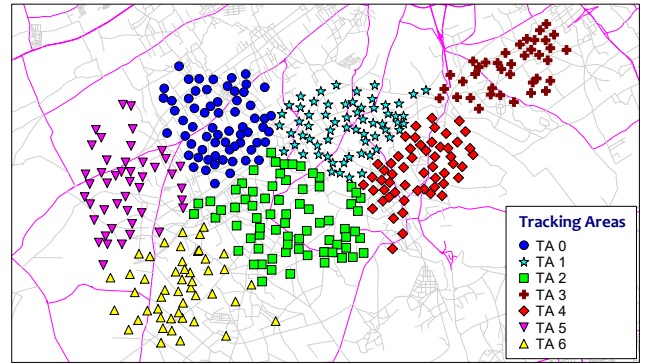
(a) Live network TA design.



(b) $\alpha = 0.4$ and $\beta = 0.8$, STSC-based TA design.



(c) $\alpha = 0.4$ and $\beta = 0.5$, STSC-based TA design.



(d) $\alpha = 0.5$ and $\beta = 0.5$, STSC-based TA design.

Figure 2. Adaptive TA design using self-tuning spectral clustering (STSC). For confidentiality considerations, the sites are displayed according to their shifted GPS coordinates.

minimization of the cost function J given by,

$$J = \sum_{i=1}^M \sum_{j=1}^C \frac{z_{i,j}^2}{\mu_i^2}, \quad (2)$$

with $\mathbf{Z} = \mathbf{X}\mathbf{R}$ and $\mu_i = \max_j z_{i,j}$. The parameters are updated in the opposite direction of the gradient ∇J following a learning rate. To that end, the STSC implementation [9] adopts automatic differentiation packages `autograd` [10] and `pymanopt` [11] to implement gradients.

The adopted STSC algorithm is detailed in Algorithm 1, and the selected parameters are listed in Table I.

Table II
SPECTRAL CLUSTERING PARAMETERS

Parameter	Value
kernel	rbf
gamma	1
Automatic differentiation	autograd

Algorithm 1: Adaptive Tracking Areas Design

Inputs: A dataset of sites $P = \{p_1, \dots, p_M\}$ to be clustered.

Outputs: Tracking area labels for the sites.

- 1: Construct a RBF-based similarity matrix \mathbf{S} according to (1).
 - 2: Set the optimization method to `autograd`.
 - 3: Define \mathbf{D} to be the diagonal matrix with elements $\delta_{i,i} = \sum_{j=1}^M s_{i,j}$ and normalize matrix \mathbf{S} as $\mathbf{L} = \mathbf{S}\mathbf{D}^{-1}$.
 - 4: Initialize STSC [9] with the normalized similarity matrix \mathbf{L} .
 - 5: STSC yields an array of vectors corresponding to the clusters.
 - 6: Assign a TA label to the elements of each output vector.
-

Table III
TAU AND PAGING PERFORMANCE GAINS

α	β	Number of TAs	TAUs	Paging Requests
Live Network		7	1 413 898	684 569 043
0.5	0.5	7	1 302 330	477 110 605
0.3	0.7	6	1 701 626	654 860 531
0.4	0.5	5	1 072 844	971 173 567
0.3	0.5	4	593 046	1 432 542 174
0.7	0.3	3	726 496	2 544 265 641
0.4	0.8	2	177 554	5 702 369 104

III. PERFORMANCE ASSESSMENT

In this section, we compare our STSC-based TA planning with the live network manual TA design depicted in Fig. 2-(a). The latter consists of 7 tracking areas with non-uniform sites distribution. Moreover, it includes an inaccurately planned TA 2. Before delving into the qualification of our clustering algorithm, let us study its convergence across different settings of parameters α and β and assess the potential gains in terms of tracking area updates (TAUs) and paging requests.

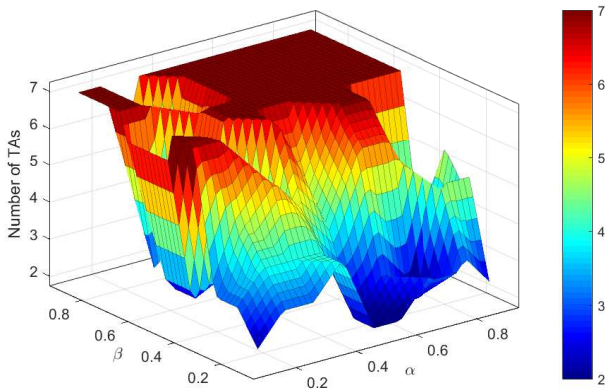


Figure 3. Number of TAs vs. parameters α and β .

A. Number of Clusters

A key remark, revealed by Fig. 2 and confirmed by Fig. 3, is that STSC converges to 7 clusters when α and β are greater or equal to 0.5, i.e., when ISD—on one hand—and the mixture of HO attempts and MRs count—on the other hand—are fairly taken into account in the construction of the similarity matrix. This is also a reasonable criteria to avoid clustering distant overshooting sites just because their MR count is high, or e.g., grouping close sites with low HO events. Under this setting, the STSC algorithm finds a trade-off between minimizing average paging requests per cluster and reducing the inter-cluster HO attempts and thereby tracking area updates. In contrast, relying on the ISD only does not consider the clutter effect that is more pronounced in the HO attempts and MRs count.

In practice, we note that for $\alpha \geq 0.5$, the number of clusters increases linearly with β . In this regime, radio network planning engineers may fix $\alpha = 0.5$ and control the number of output TAs by fine-tuning β .

B. TAU and Paging Performance Gains

Table III depicts the TA updates and average paging requests per TA for various combinations of α and β . In this regard, we remark that—as expected—increasing the number of clusters leads to the augmentation of TAUs and decrease of paging requests per TA. With 7 clusters, our STSC-based scheme achieves a better performance compared with the live network, with a reduction of 8% in TAUs and 30% in paging requests. In practice, radio network planning engineers may fine-tune α and β to control the load of either the tracking area updates or paging in such a way to optimize radio resources (e.g., downlink (DL) physical resource blocks (PRBs) consumed by the S1 paging).

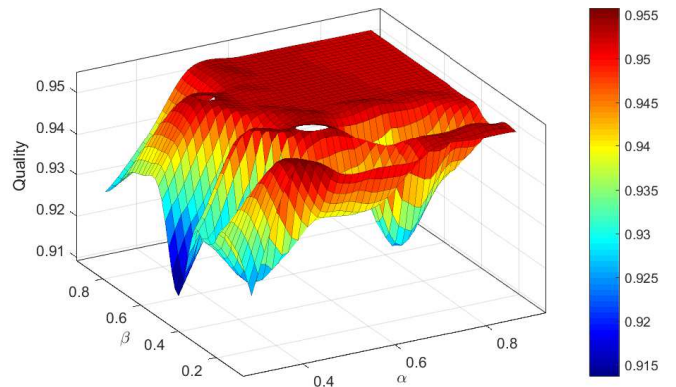


Figure 4. STSC clustering quality vs. parameters α and β .

C. STSC Quality

Upon the convergence of STSC, the minimum cost J_{\min} is used to define the quality Q of the clustering with values

ranging between 0 and 1 [7], where

$$Q = 1 - \frac{(J_{\min}/M) - 1}{C}. \quad (3)$$

A high quality clustering means that almost every site is assigned to the closest cluster, minimizing thereby the cost function J and approaching to $Q = 1$. In contrast, rural sites—that usually present a sparse distribution—increase the cost function and reduce the global clustering quality. In this respect, STSC generally groups suburban sites in the same TA. This is the case of TA 3 in Fig. 2-(c).

In Fig. 4, we plot the Q -metric versus α and β , and we notice that the best quality is obtained when $\alpha, \beta \in [0.5, 1]$, i.e., when the number of clusters is 7.

D. Silhouette Score

The silhouette score σ displays a measure of how close each point in one cluster is to the points in the neighboring clusters, and thus provides a way to assess parameters like number of clusters visually. This measure has a range of $[-1, 1]$ and is readily available in `scikit-learn` package [12]. The corresponding expression for site i reads,

$$\sigma_i = \frac{\bar{d}_i - d_i}{\max(\bar{d}_i, d_i)}, \quad (4)$$

where d_i and \bar{d}_i stand for the average distance of site i to the sites within the same cluster and the smallest average distance of i to all sites in any other cluster, respectively. The obtained positive silhouette score in Fig. 5 means that—in average—the dataset sites are assigned to a close cluster (in terms of the similarity weight defined in (1)). Note that sparsely distributed suburban sites such as TA 3 in Fig. 2-(c) might present generally a negative silhouette score that degrades the overall performance.

IV. CONCLUSION

In this paper, we have presented a machine learning approach to automate tracking areas design in future 5G networks. It relies on a self-tuning spectral clustering algorithm capable of grouping gNBs without requiring the number of clusters as input. Alternatively, we feed STSC with a new kernel similarity matrix; taking into account inter-site distance, hand-over attempts and A3 events measurement reports count. The presented approach yields a significant reduction of tracking area updates and average paging requests per TA, and might be adopted by radio network planning engineers to periodically update TA design according to network evolution.

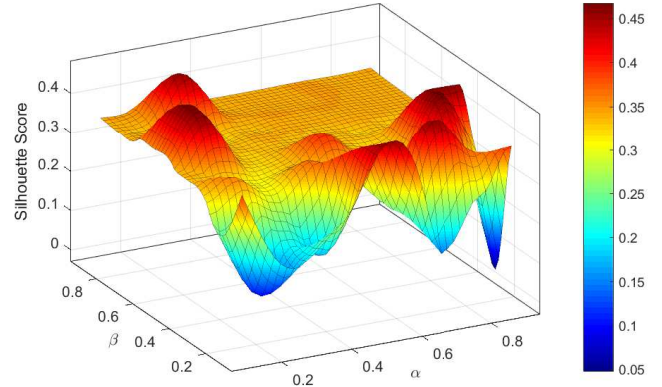


Figure 5. Silhouette score vs. parameters α and β .

ACKNOWLEDGEMENT

This work is supported by the telecom operator INWI, Casablanca, Morocco.

REFERENCES

- [1] 3GPP, “Evolved packet system (EPS) mobility management entity (MME) and serving GPRS support node (SGSN) related interfaces based on Diameter protocol,” TS 29.272 v15.4.0, Jun. 2018.
- [2] 3GPP, “Evolved universal terrestrial radio access network (E-UTRAN): Self-configuring and self-optimizing network (SON) use cases and solutions,” TS 36.902 v9.3.1, Mar. 2011.
- [3] S. M. Razavi and D. Yuan, “Performance improvement of LTE tracking area design: A re-optimization approach,” in *Proc. the 6th ACM International Workshop on Mobility Management and Wireless Access (MobiWac’08)*, pp. 77-84, 2008.
- [4] S. M. Razavi *et al.*, “Dynamic tracking area list configuration and performance evaluation in LTE,” in *Proc. IEEE Global Communications Conference (GLOBECOM Workshop’10)*, 2010.
- [5] T. Deng *et al.*, “Modeling and performance analysis of a tracking-area-list-based location management scheme in LTE networks,” in *IEEE Trans. on Veh. Tech.*, vol. 65, no. 8, Aug. 2016.
- [6] M. Toril, S. Luna-Ramírez and V. Wille, “Automatic replanning of tracking areas in cellular networks,” in *IEEE Trans. on Veh. Tech.*, vol. 62, no. 5, pp. 2005-2013, Jun. 2013.
- [7] L. Zelnik-Manor and P. Perona, “Self-Tuning Spectral Clustering,” in *Proc. the 17th International Conference on Neural Information Processing Systems (NIPS’04)*, pp. 1601-1608, 2004.
- [8] U. V. Luxburg, “A tutorial on spectral clustering,” in *Statistics and computing*, vol. 17, no. 4, pp. 395-416, Aug. 2007.
- [9] WOOL, “STSC,” 2018. [Online] Available: <https://github.com/wOOL/STSC>
- [10] HIPS Group, “Autograd,” 2018. [Online] Available: <https://github.com/HIPS/autograd>
- [11] J. Townsend, N. Koep and S. Weichwald, “Pymanopt: A Python toolbox for optimization on manifolds using automatic differentiation,” in *Journal of Machine Learning Research*, vol. 17, no. 137, pp. 1-5, Aug. 2016. [Online] Available: <https://pymanopt.github.io/>
- [12] F. Pedregosa *et al.*, “Scikit-learn: Machine learning in Python,” in *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, Oct. 2011.