Complexity-Performance Trade-offs in Robust Access Point Clustering for Edge Computing

Nour-El-Houda Yellas, Selma Boumerdassi, Alberto Ceselli, Stefano Secci

Complexity-Performance Trade-offs in Robust Access Point Clustering for Edge Computing

Nour-El-Houda Yellas, Selma Boumerdassi, Alberto Ceselli*, Stefano Secci
Cnam, 292 rue St. Martin, 75003, Paris, France. Email: firstname.lastname@cnam.fr
*Università degli Studi di Milano, Dip. Informatica, Milan, Italy. Email: alberto.ceselli@unimi.it

Abstract—**Edge computing penetration in mobile access networks is the next barrier to break in communication networks. The virtualization of radio access functions currently under study is expected to trigger the deployment of edge cloud facilities in telecom operator points-of-presence and central offices, to serve the virtualization of both application servers and network functions. The problem of clustering network access points for their assignment to edge cloud facilities has been addressed in the literature. Nonetheless, the inclusion of key-performance indicators such as robustness against traffic variations in the optimization process can increase its complexity excessively while hindering the achievable performance. Leveraging on previous work in this area, in this paper we explore how to reduce time and spatial complexity while introducing additional a robust access point assignment target by using a spatial clustering preprocessing in the optimization problem, grouping together access points based on their spatio-temporal traffic profile. By extensive simulation against real traffic traces and network maps, we show under which conditions we can outperform existing methods at the state of the art. The obtained results show that our approach helps reducing time and space complexity for small to medium instances, indicating the geographical scale at which these operations could be run in a near-real-time manner.**

I. Introduction

Multi-access Edge Computing (MEC) infrastructures move the cloud computing data-center edge down into the access networks to better meet current and forthcoming requirements of pervasive computing applications. Besides the original mobile application-driven trigger in their design, MEC infrastructures are nowadays recognized as a 5G key enabler, thanks to the capability to converge at MEC facilities both application servers and virtualized network functions. Indeed, in 5G, Network Functions Virtualization (NFV) becomes a compulsory technology to run carrier-grade functions related to the 5G core control-plane and data-plane [1], as well as to radio access network (RAN) subsystems [2].

The set of constraints can therefore increase and can go beyond basic low latency, high bandwidth and real time access to users location and up-to-date radio network information requirements [3]. Colocating application servers, cellular core data and/or control functions, vRAN (virtualized RAN) subsystems at the same MEC host facility make the assignment of base stations (BS) access points to MEC hosts an orchestration task with direct impact on network operations costs and overall end-to-end infrastructure reliability [4].

The location of MEC hosts is currently envisioned by telecom operators to happen at so-called Central Offices (CO) and/or Points-of-Presence (PoP). The distribution of MEC hosts horizontally across different access network segments and vertically at different layers of the backhauling network is needed to meet the access latency and reliability requirements. Typically, MEC hosts are meant to be therefore situated between BSs and the core network [5]. Strictly speaking [6], a 'MEC host' (cloudlet or MEC facility) refers to the hardware servers belonging to the virtualization infrastructure; it can be generic or NFVI (NFV Infrastructure) based, and in this case the MEC host can be deployed as a VNF (Virtual Network Function), possibly supporting network slicing [7]. The 'MEC platform' is responsible for managing MEC applications.

Under the above mentioned convergence of multiple service and network functions at MEC hosts, and the related high diversity of constraints to take into account to meet the diverse set of requirements, MEC orchestration algorithm scalability and result robustness are key concerns to address. We address the scalability-robustness challenge by extending a problem formulation and related algorithm in [8]. More precisely, we propose the integration of spatial clustering as a precomputation step to the algorithm in [8] to reduce the number of variables of constraints, while integrating in the spatial clustering optimization an objective that aims at making the access point to MEC host assignment more robust against traffic variations within a cluster. We numerically show for which MEC network sizes the problem becomes tractable.

This paper is organized as follows. In Section II we give an overview about existing works. Our contribution is presented in Section III. Simulations results are analyzed in Section IV. We draw conclusions in Section V.

II. Background

In the following, we provide the necessary background on network virtualization, network analytics and optimization.

A. Network virtualization

MEC is one of the 5G key enabler technologies whose main goal is to reduce access latency and optimize bandwidth to provide real time performance. Combining it with NFV technology can be of a great benefit for mobile network operators since the management operations can be held by the NFV architecture, more precisely by the NFV MANO (Management and Orchestration) subsystem [6].

Several works exist in the area of MEC-NFV environment, and either MANO algorithms or solutions taking profit from the presence of a MEC-NFV environment. For instance, [9] addresses the relationship between MEC and other technologies that are considered as 5G enablers such as NFV and SDN: authors propose an architectural framework where an SDN controller is responsible of management operations in a MEC-NFV environment, hence being able to reconfigure the network stack to take into consideration orchestration decisions such as on the assignment of BSs to MEC hosts. Other works focus on VNF placement in a MEC environment [10], [11], balancing the placement across multiple MEC locations. A clustering scheme for network service chaining is proposed in [12] in order to minimize end-to-end service latency in MEC. More details about the MEC architecture and different orchestration and deployment scenarios are presented in [13].

From a radio-access perspective, the architectures have evolved toward the virtualization and disaggregation of its control-plane and data-plane functions to improve interference coordination and resource efficiency. This evolution started with the Centralized RAN or Cloud-RAN, in 2010, where the innovation consisted in disaggregating BS facilities into two main units: Radio functions assured by RRHs that are deployed on cell sites, and Base Band computation functions provided by BBUs (BaseBand Units). BBUs are then centralized at so-called BBU pools, hence taking profit from the centralization for resource allocation and scaling [14]. More recently, the CRAN evolution has been integrated in 5G systems, where a more dense deployment of BSs is needed for a more flexible infrastructure, leading to a generalized virtualized or software-defined RAN (vRAN or SD-RAN) environment. In vRAN, the equivalent of the BBU function is split into two units, the Centralized Unit (CU) and the Distributed Unit (DU), in order to facilitate the virtualization and radio scheduling tasks [15], while the radio part is called Radio Unit (RU). Splitting radio processing functions is known as functional split [14] and it enables to choose the functions that turn on cell sites and those that will be offloaded to CUs, with different splitting options, and that possibly in a dynamic (runtime) and flexible (different options decisions for different segments and times) fashion.

Many works investigate on how to combine vRAN and MEC technologies [16]. In [17] authors implement a MEC platform on the vRAN front-haul and evaluate the QoS for end users for two different locations of MEC hosts. Another work consists of proposing a MEC vRAN joint design problem where authors introduce an optimization framework that aims at the same time to find the best functional split of BSs and MEC service placement, taking into account flows routing [2]. The integration of vRAN with SD-x system lead to the Open RAN (ORAN) initiative, which has the goal to desegregate software and hardware and to create open interfaces between them for more flexibility. A first ORAN software suite was lately released [18]. In [19] authors discuss RAN evolution where they present ORAN reference architecture.

The standpoint we adopt is the one of an operator running a MEC infrastructure the operator leverages on, for converging MEC applications and virtualized network functions. Hence BSs are assigned to MEC hosts facilities in a dynamic way by means of MANO operations, and leveraging on a programmable network stack between BSs and MEC infrastructure, hence going largely beyond the legacy situation where BSs are statically assigned to COs and PoPs. In our work, we therefore do not need to delve into the details related to, for instance, functional splitting and the actual coexistence of NFV and MEC systems; on the other hand, our model has to take into consideration the traffic fluctuations deriving from the BSs to MEC hosts assignments and related MEC switching.

B. MEC infrastructure planning

In [20], a study was conducted on how a MEC infrastructure should be planned, that is, where MEC facilities should be placed, and that as a function of different MEC resource placement policies. A take-away result of that work that we take into consideration in our work is that for a large metropolitan area network as the one of Paris, France, the number of MEC facilities ranges from 5 to 20. The workload was the one equivalent to plan for as much as one virtual machine per mobile user, which can be considered as an upper bound, and that for a network of approximately 180 thousands users with 606 BSs. Authors used real data volume information from Orange France mobile network. We evaluate in our work MEC infrastructure setting with 10, 20, 30 and 50 MEC facilities so as to cover different sizes, while keeping in mind that for a scope as the Parisian one 20 MEC facilities can be considered as a reference upper-bound size.

C. Data-driven MEC orchestration

A wide range area of works using data for network and services management in MEC exists in literature. The applications different in the time scale at which mobile data-analytics needs to be done. Stream/online data-analytics is needed in mobile computation offloading frameworks, where for tasks offloading online decisions need to be made. For instance, a feedback prediction model of average resource usage (RAM and CPU) and offloading time is proposed in [21]. In [22] authors tackle the offloading decision for MEC applications where the performance of the solution is evaluated using real world dataset. [23] aims at offloading intensive computing tasks for energy saving by optimizing resource allocation, and [24] presents solutions for computation offloading in edge servers for internet of connected vehicles.

The BS to cloud/network facility assignment problem is a frequent subject in the literature [25]. Because of MEC hosts limited resources, resource orchestration is a very important task for optimizing its utilization [26]. Thus, operations that consist of re-assigning BSs to other MEC hosts need to be deployed. Similarly, clustering techniques are often used in RAN optimization problems; in [27] [28] authors propose a clustering scheme for BSs, where BSs of each cluster share the same data processing units that are centralized in data-centers to optimize costs and energy consumption in vRAN. However, few works using clustering to manage the MEC network infrastructure exist. Authors in [29] aim at predicting mobile traffic generated by a cluster of BSs to anticipate MEC resource orchestration using real world dataset. Different mobile service types are taken into consideration.

In [30] authors propose a geo-clustering method of BSs while taking into account the spatial distribution of mobile traffic. The main goal is to define MEC clusters as a set of BSs and users served by the same MEC host, so that at the end the whole area will be partitioned into MEC clusters, in order to offload the core network by maximizing intra MEC hosts communications. Similarly, in [8] where authors apply a temporal clustering model proposed in [31] on traffic demands of a real world dataset and integrate it into an orchestration model. The temporal clustering consists of grouping together similar mobile network profiles using the traffic volume generated by BSs at a time slot, this allows to retrieve a reduced number of profiles. Here, similarity is based on traffic volume and traffic distribution. On the other side, the orchestration model consists of assigning BSs to a set of MEC hosts for each time slot. The objective is to find an assignment and switching plan where a BS belongs to exactly one MEC host and time slots of the same profile have the same plan. The resolution approach in [8] consists of iteratively solving a version of a linear program involving only a (small) subset of its variables, finding variables outside this subset having negative reduced cost (pricing), and enlarging the subset with these variables; pricing is in turn an optimization problem: when no more negative reduced cost variables can be found by pricing, optimality for the full problem is reached.

We go beyond [8] by introducing robust spatial clustering, aiming at reducing both space and time complexity while enhancing the solution robustness against traffic variations within the cluster. Our contribution can be resumed as:

- a robust clustering model for spatio-temporal grouping of BS, formulating it as an integer linear program,
- that accordingly finds an optimal pattern for assigning clusters of BSs to MEC hosts.
- by application to real world data collected at two different regions in France.

III. Problem Formulation

In this section we elaborate our contribution. We start with a concise description of the addressed optimization problem, then we detail the pre-processing step where we propose a spatial clustering model with a robust assignment objective, and finally we present the orchestration model when applied to the set of clusters issued from the clustering step.

A. Problem statement

We describe our optimization framework as an orchestration problem that aims at assigning a group of BSs belonging to a given geographical area to a set of MEC hosts deployed at the edge network. The assignment operations come at a cost defined by the access latency for users connected to these BSs. On the other hand, unlike traditional Cloud datacenters, MEC hosts have limited capacities, thereby reallocating resources occasionally is requested to cope with traffic variation. Given the lower traffic granularity at MEC hosts, resource reallocation entails a cost for operators because it could generate service-level-agreement violations and hence a VM workload variation across MEC hosts to get back to nominal conditions.

To reduce the spatial and temporal complexity of the orchestration process, we propose to group together BSs into clusters based on their spatio-temporal behavior so that the likelihood of traffic variation within the cluster is minimized. These requirements lead us to the adaptation of the orchestration model in [8] using the clusters in place of BSs, using a robust assignment in the clustering process. To minimize the likelihood of cluster traffic variation, we opt for minimizing the variance of BS traffic volume within the cluster.

B. Spatial clustering model

In our spatial clustering model, we search to group BSs so that for each time slot, the difference between their traffic demands is minimized. In order to have a linear and expressive robust clustering objective, we express the traffic variance minimization by minimizing the gap between the maximum and minimum BS demand within the clusters.

For the instrumentation of the spatial cluster, we do as follows. We fix the number of clusters as corresponding to the number of MEC hosts. Given the collected traffic demands, we calculate the representative week by averaging demands of the same period of the week; we then aggregate the traffic demands of successive time periods aiming at reducing the number of intervals of time. In total we get a set of time slots that compose our training set.

For the spatial clustering optimization we aim at grouping together BSs that have for each time slot t similar traffic demands so that the likelihood to have traffic fluctuation is reduced or can at least be relatively easy predictable - our tests to evaluate this assumption revealed to be extremely positive with the available dataset, which confirms that BS traffic profiles within a not too large time-slot do follow a similar temporal behavior over time [31].

The mathematical formulation is as follows:

$$\min \sum_{c \in C} (M_c - m_c) \tag{1}$$

$$\text{s.t.} \quad \max_{t \in T} \sum_{i \in A} d_i^t x_i^c \leq Cap_{MEC} \qquad \forall c \in C \tag{2}$$

$$\sum_{c \in C} x_i^c = 1 \qquad \forall i \in A \tag{3}$$

$$\sum_{i \in A} d_i^t x_i^c \leq M_c \qquad \forall c \in C, t \in T \tag{4}$$

$$\sum_{i \in A} d_i^t x_i^c \geq m_c \qquad \forall c \in C, t \in T \tag{5}$$

$$\mathrm{x}_i^c \in \{0,1\} \qquad \forall i \in A, c \in C \tag{6}$$

Where T refers to the set of all time slots, C is the set of clusters and A is the set of all BSs. d_i^t represents the traffic demand generated by the BS i at the time slot t. In our dataset, we have the traffic demands recorded for each BSs separately for each 10 minutes during a given period of time. To solve our problem we use the binary variable x_i^c, it is equal to 1 if the BS i belongs to cluster c, 0 otherwise. We also need to calculate the two real variables M_c and m_c where the former represents the demand traffic of a BS i representing the maximum for a time slot and belonging to cluster c, and the latter represents the demand traffic of a BS i representing the minimum for a time slot and belonging to cluster c, and finally Cap_{MEC} is the capacity of each MEC host. The objective function in (1) aims at minimizing the difference, for all the clusters, between the maximum and minimum traffic demands yield by BSs belonging to the same cluster. Constraint (2) ensures that the maximum traffic demands that can be handled by each cluster must not exceed MEC hosts capacity. In (3) we guarantee that a BS belongs to exactly one cluster. (4) and (5) ensure the M_c and m_c computation, i.e., the maximum and the minimum traffic demand generated by a BS that belongs to cluster c at time slot t, respectively. (6) is an integrality constraint.

C. Orchestration optimization model

Our proposal consists of resolving the orchestration problem where we apply the same orchestration decision on BSs belonging to the same cluster. For this purpose we extended the orchestration model from [8] to fit with our spatial clustering model. The model is represented by equations from (7) to (13). In Table I we define all notations used in the model.

A	Set of all base stations (BSs).
K	Set of all MEC hosts.
T	Ordered set of time slots.
T'	$T' \subset T$ subset of T excluding the first time slot in T.
T''	$T'' \subset T$ subset of T excluding the last time slot in T.
C	Set of all clusters.
x^t_{ck}	Real variable, upper than 0 and less or equal to 1 if cluster c is assigned to MEC host k at time slot t, 0 otherwise.
y^t_{cjk}	Real variable, upper than 0 and less or equal to 1 if traffic demand of cluster c must be switched from MEC host j to MEC host k at time slot t, 0 otherwise.
M_c	Variable computing the maximum BS demand within cluster c.
m_c	Variable computing the minimum BS demand within cluster c.
d^t_c	Traffic demand of cluster c at time slot t.
ljk	Distance between the two MEC hosts j and k.
m_{ik}	Distance between the BS i and the MEC host k.

TABLE I: MEC orchestration model notations.

The objective (7) aims at finding the assignment and switching plans for each cluster to a set of MEC hosts and each time slot while minimizing the network (switching) and users (assignment) costs. In (8) we ensure that the overall traffic demand assigned to a MEC host must not exceed its capacity. (9), (12) and (13) give the possibility to assign a cluster to one or more MEC hosts for each time slot. (10) (resp. (11)) reflects the coherence of the assignment and switching plans. It guarantees that if c is assigned to MEC host k at t, then c is switched from (resp. to be switched to) one or many MEC hosts $j \in K$ including $j = k$.

$$\min \sum_{t \in T} \sum_{c \in C} \sum_{\substack{(j,k) \in \\ K \times K}} d^t_c l_{jk} y^t_{cjk} + \sum_{t \in T} \sum_{c \in C} \sum_{k \in K} d^t_c m_{ck} x^t_{ck} \tag{7}$$

$$\text{s.t.} \sum_{c \in C} d^t_c x^t_{ck} \leq Cap_{MEC} \qquad \forall k \in K, \forall t \in T \tag{8}$$

$$\sum_{k \in K} x^t_{ck} = 1 \qquad \forall c \in C, \forall t \in T \tag{9}$$

$$x^t_{ck} = \sum_{j \in K} y^t_{cjk} \qquad \forall c \in C, \forall k \in K, \forall t \in T' \tag{10}$$

$$x^t_{ck} = \sum_{j \in K} y^{t+1}_{cjk} \qquad \forall c \in C, \forall k \in K, \forall t \in T'' \tag{11}$$

$$x^t_{ck} \in [0,1] \qquad \forall c \in C, \forall k \in K, \forall t \in T \tag{12}$$

$$y^t_{cjk} \in [0,1] \qquad \forall c \in C, \forall j,k \in K, \forall t \in T \tag{13}$$

IV. Results

We describe the dataset and evaluation numerical results.

A. Data

We used a dataset with traffic demands collected at the core and access network of the French mobile operator ”Orange”, at a national scale and for a period of three months. The collection process takes into account both 3G and 4G connections and describes traffic demands generated by several services and aggregated at the antenna level. Volume data is collected every 10 minutes, which is the time slot duration.

B. Numerical results

We assess the results through CDF (Cumulative distribution Function) of spatial and time complexity metrics, i.e., maximum memory usage (GB) and execution time (s), as well as the optimality gap (%) and the assignment and switching costs. We use up to 50 different locations for MEC hosts, hence using different numbers of MEC host facilities to generate the training set, for the cities of Paris and Lyon. MEC locations are generated using a centroid based clustering, i.e., K-means clustering where the centroids of BS clusters represent MEC hosts locations. For each simulation we randomly generate the parameter representing the number of time the k-means algorithm is executed, then the best results are returned based on inertia. We evaluate the following four algorithms to solve the orchestration problem:

- ‘MECA’: solving the reference orchestration model without spatial clustering, i.e., (7)-(13) with $C \equiv A$;
- ‘MECA-CS’: solving the reference orchestration model with spatial clustering, i.e., (1)-(13);
- ‘MECA-CG’: solving the reference orchestration model without spatial clustering and using the dynamic variable generation approach proposed in [8];
- ‘MECA-CG-CS’: as MECA-CG but with spatial clustering precomputation.

In the following we present simulation results generated by both Paris and Lyon datasets, using different MEC infrastructure sizes: 20 and 50 MEC facilities, and 10, 20 and 30 MEC facilities respectively.

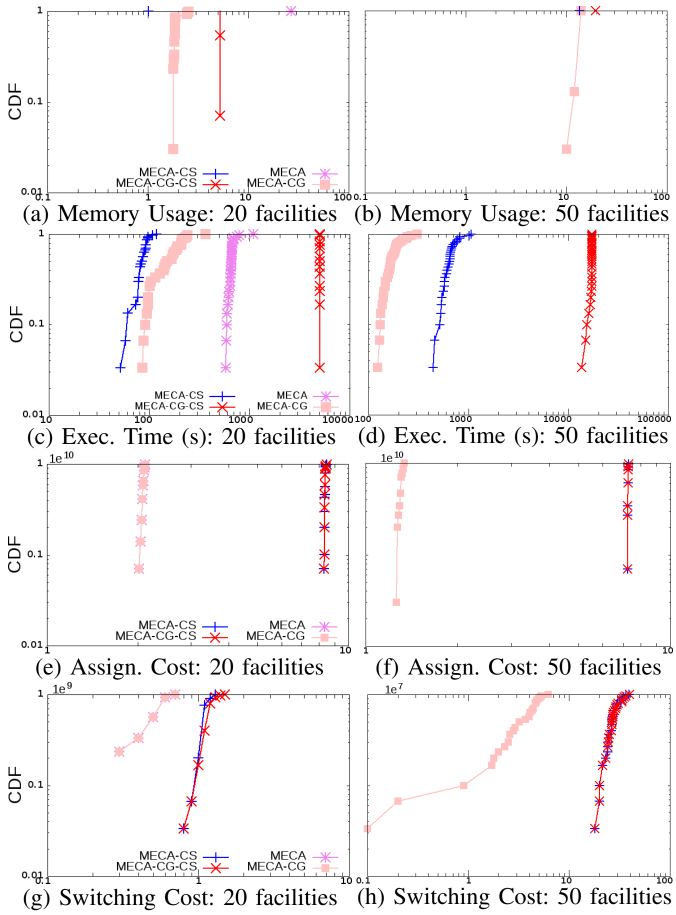
Due to its high memory consumption, we could not execute the MECA case for the two highest MEC infrastructure sizes, i.e., 50 and 30, for Paris and Lyon, respectively.

MECA-CG-CS was the least memory consuming case on single computations. Contrary to expectations, it increased on average by 4.9 GB (1600%) for Paris dataset using 20 MEC hosts and by 1.3 GB (540%), 2.7 GB (700%) and 4.9 GB (980%) for Lyon dataset using respectively 10, 20 and 30 MEC hosts when post-processing the intermediate solutions to retrieve the variable vectors. The maximum execution time limit is set to 17000 s for all instances, seldom reached.

C. Paris dataset

The 0% optimality gap was reached for all the approaches and for all cardinalities, except for MECA when using 50 MEC hosts. As aforementioned, it stopped before getting any results.

In Figure 1 we present the distribution of the maximum memory usage in GigaBytes (GB), the execution time in seconds (s) and the assignment and switching costs for Paris dataset for two different sizes for the MEC infrastructure, i.e, 20 and 50 facilities.



(a) Memory Usage: 20 facilities (b) Memory Usage: 50 facilities

(c) Exec. Time (s): 20 facilities (d) Exec. Time (s): 50 facilities

(e) Assign. Cost: 20 facilities (f) Assign. Cost: 50 facilities

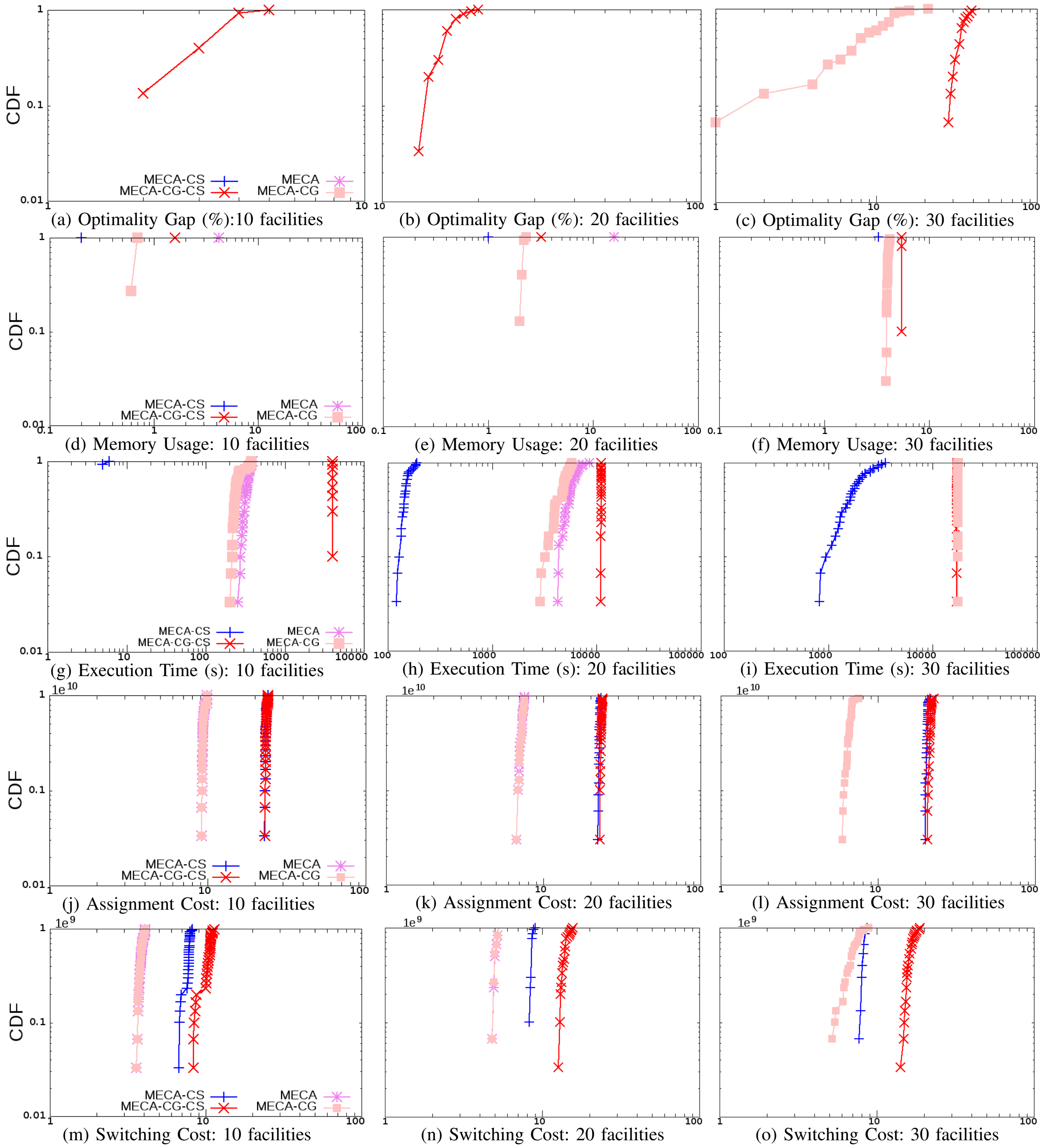
(g) Switching Cost: 20 facilities (h) Switching Cost: 50 facilities

Fig. 1: Paris dataset.

The distribution of the maximum memory used by each of the approaches is depicted in Figures 1a and 1b using Paris dataset and respectively 20 and 50 MEC hosts. We notice that: when using 20 MEC hosts, MECA-CS has a constant maximum memory usage through all the proposed MEC hosts locations and it represents the lowest value (1 GB) compared to all the other approaches, followed by MECA-CG and MECA-CG-CS (2.5 and 5.25 GB as maximum values respectively). Meanwhile, MECA is the most memory consuming and it has also a constant consumption through all the proposed locations (26 GB). For 50 MEC hosts, MECA-CS and MECA-CG have a close maximum memory usage on average, the former has a constant consumption equal to 13.6 GB while the latter consumption varies between 10 GB and 14 GB. However, MECA-CG-CS has the highest values that reach 19.4 GB. MECA has the highest memory consumption and it stopped before reaching the final solution because of lack of memory. Increasing MEC hosts number from 20 to 50 has increased the maximum amount of memory used by each of the proposed approaches. Figure 1c (resp. Figure 1d) represents the distribution of execution time values required by each approach when using 20 MEC hosts (resp. 50). We note that: MECA-CS is the fastest approach when using 20 MEC hosts followed by MECA-CG with execution time values that go from 50 s to 120 s and from 86 s to 360 s respectively. On the other side, when increasing the infrastructure size to 50 MEC hosts MECA-CG becomes the fastest one reaching 300 s, followed by MECA-CS where the execution time is between 440 s and 1000 s. MECA needs a higher execution time that reaches 600 s at least and 1000 s at most for 20 MEC hosts. However, MECA-CG-CS represents the highest execution time and requires around 5000 s for 20 MEC hosts and reaches the execution time limit with 50 MEC hosts. It is worth mentioning that it was clearly stated in [20] that 20 MEC hosts is sufficient to satisfy strict requirements in terms of latency and bandwidth.

The distribution of the assignment costs is presented in Figures 1e and 1f for both 20 and 50 facilities, we can notice that: the approaches without the spatial clustering yield a lower assignment cost compared to the ones using it. Let us underline that the spatial clustering adds a constraint to the orchestration problem that produces the same assignment plan to all BSs that belong to the same cluster. When the 0% optimality gap is reached, MECA-CS and MECA-CG-CS (MECA and MECA-CG respectively) have roughly the same assignment cost: a little difference can be noticed due to numeric precision used by the two methods.

In Figures 1g and 1h, we present the distribution of the switching costs. We notice that: there is a slight cost difference between the two approaches (with vs without spatial clustering precomputing). As explained, the approaches spatial clustering-based produce additional costs due to proposing the same switching plan to BSs that belong to the same cluster. Achieving the 0% optimality gap produces the same switching costs for MECA-CS and MECA-CG-CS (and for MECA and MECA-CG respectively), and increasing the number of MEC hosts has reduced both the switching and the assignment costs for all the approaches.



(a) Optimality Gap (%):10 facilities

(b) Optimality Gap (%): 20 facilities

(c) Optimality Gap (%): 30 facilities

(d) Memory Usage: 10 facilities

(e) Memory Usage: 20 facilities

(f) Memory Usage: 30 facilities

(g) Execution Time (s): 10 facilities

(h) Execution Time (s): 20 facilities

(i) Execution Time (s): 30 facilities

(j) Assignment Cost: 10 facilities

(k) Assignment Cost: 20 facilities

(l) Assignment Cost: 30 facilities

(m) Switching Cost: 10 facilities

(n) Switching Cost: 20 facilities

(o) Switching Cost: 30 facilities

Fig. 2: Lyon dataset.

D. Lyon dataset

Figure 2 depicts the numerical results when using 3 different sizes for our infrastructure (10, 20 and 30 facilities) and traffic demands from Lyon dataset. We report the results of the 5 aforementioned metrics.

Figures 2a, 2b and 2c present the distribution of the optimality gap values using Lyon dataset and the three different sizes for the MEC infrastructure. We note that: 0% optimality gap was reached by MECA-CS, MECA and MECA-CG after a finite execution time when using 10 and 20 MEC hosts, contrary to MECA-CG-CS that reached optimality gaps between 2% and 5% and between 10% and 17% respectively, depending on the MEC hosts locations. This can be explained by the use of large scales, i.e., aggregated demands of all BSs that belong to the same cluster to get the cluster traffic demand. For 30 MEC hosts, we reached the 0% optimality gap only with MECA-CS in a finite time. For MECA-CG, values are between 0% and 20%. Meanwhile, MECA-CG-CS has the highest values that varies between 28% and 44%.

Figures 2d (resp 2e and 2f) depicts the distribution of the maximum amount of memory used by the processes run by the proposed approaches in GigaBytes (GB) when using 10 MEC hosts (resp. 20 and 30 MEC hosts). We note that: for all cardinalities, the best case always corresponds to MECA-CS with a constant memory consumption for all the 30 MEC hosts locations, i.e. 0.15 GB for 10, 1 GB for 20 and 3.29 GB for 30 MEC hosts, followed by MECA-CG with maximum consumption of 0.72 GB for 10 MEC hosts, more than 2 GB for 20 and 4.19 GB for 30. MECA has the highest memory consumption peak for both cases 10 and 20 where it reaches respectively 4 GB and 15.7 GB. However, when using 30 MEC hosts simulations has stopped before reaching any solution due to its high memory consumption. Meanwhile, MECA-CG-CS has an intermediate consumption between MECA and MECA-CG for all cardinalities, i.e. 1.5 GB, 3.2 GB and 5.4 GB.

In Figures 2g, 2h and 2i we present the CDF histograms of the required execution times by the proposed approaches and for the 30 MEC hosts different locations for each of the three infrastructure sizes. We note that: comparing the two approaches MECA-CS and MECA highlights the contribution of the spatial clustering: MECA-CS is the fastest approach with an execution time less than 7 s, between 120 s and 190 s and less than one hour for the three cardinalities. Nevertheless, MECA reaches 390 s and 2 hours of execution time for the first two infrastructure sizes. For 30 MEC hosts the algorithm did not get any results. For 10 and 20 MEC hosts, MECA-CG requires between 200 s and 380 s and 1 hour and a half whereas MECA-CG-CS gives the worst case with an execution time exceeding 1 hour and 3 hours respectively. Hence, both MECA-CG and MECA-CG-CS reached the execution time limit which is 17000 s when using 30 MEC hosts.

We present in Figures 2j, 2k and 2l the distribution of the assignment costs values yield by the proposed approaches. We note that: the assignment costs yield by approaches using spatial clustering model are higher than costs generated by the approaches spatial clustering-free. For 10 and 20 MEC hosts cases a 0% optimality gap was reached by both MECA and MECA-CG, so the two assignment costs are equal. We can also notice that the assignment cost decreases when we broaden the MEC infrastructure size.

We present in Figures 2m, 2n and 2o the distribution of the switching costs values yield by the proposed approaches when using Lyon dataset. We note that: for 10 and 20 MEC hosts (2m and 2n), the switching costs are lower when not using the spatial clustering as explained before. MECA and MECA-CG have the same and lowest switching cost, followed by MECA-CS (less than double) and finally MECA-CG-CS. On the other hand, when using 30 MEC hosts (2o) we notice that MECA-CS and MECA-CG have the same switching cost for some MEC hosts locations. MECA-CS and MECA-CG-CS have different switching costs because this latter could not reach the 0% optimality gap. The switching cost increases when increasing the MEC infrastructure size from 10 to 20 facilities. However, increasing it from 20 to 30 has decreased the switching cost for MECA-CS and increased it for MECA-CG.

V. Conclusion

In this work we focused on the optimization of algorithms that deal with base-station access-point to MEC hosts assignment orchestration decisions by taking into account an assignment objective robust against traffic fluctuations. For this purpose, we proposed a spatial clustering model which consists of grouping together base-station access points into clusters that reveal the same spatio-temporal traffic through time. Afterwards, a data-driven solution for MEC orchestration was added to the model. The results from extensive simulation on a real world dataset show that our approach outperforms existing algorithms while helping reduce time and space complexity especially for small to medium instances, i.e., 10, 20 and 30 MEC hosts for Lyon city and 20 MEC hosts for Paris city. As aforementioned, a previous work has evidently demonstrated that using around 20 MEC hosts for the region of Paris would therefore be more than sufficient for realistic massive MEC service deployment, even with strict constraints on latency and maximum link utilization.

Despite the fact that the spatial clustering model entails an additional cost due to the constraint that imposes the same assignment and switching plan for base-station access points belonging to the same cluster, numerical results have shown that our framework can be carried out in a near-real-time manner. Future works may further push time-execution requirements barrier for real-time MEC orchestration, integrating real-time traffic prediction.

Acknowledgment

This work was supported by the ANR CANCAN Project. We would like to thank Marco Premoli from Univ. degli Studi di Milano and Cezari Ziemlicki from Orange for their support.

References

[1] W. d. S. Coelho, A. Benhamiche, N. Perrot, S. Secci, ”Network Function Mapping: From 3G Entities to 5G Service-Based Functions Decomposition”, *IEEE Comm. Standards Magazine* 4(3):46-52, Sept. 2020.

[2] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez and D. J. Leith, ”Joint Optimization of Edge Computing Architectures and Radio Access Networks”, *IEEE J. on Selected Areas in Communications* 36(11):2433-2443, Nov. 2018.

[3] ”Multi-access Edge Computing (MEC); Radio Network Information API”, RGS/MEC-0012v211RnisApi, V2.1.1 (2019-02).

[4] F. Giust et al. ”MEC deployments in 4G and evolution towards 5G”. ETSI White paper, 2018, vol. 24, no 2018, p. 1-24.

[5] S. Kekki et al. ”MEC in 5G networks”. ETSI white paper, 2018, vol. 28, p. 1-28.

[6] ”Deployment of Mobile Edge Computing in an NFV environment”, DGR/MEC-0017MECinNFV, V1.1.1 (2018-02).

[7] ”Multi-access Edge Computing (MEC); Support for network slicing”, DGR/MEC-0024NWslicing, V2.1.1 (2019-02).

[8] A. Ceselli et al. ”Prescriptive Analytics for MEC Orchestration”, *Proc of IFIP Networking 2018.*

[9] A. Filali et al. ”Multi-Access Edge Computing: A Survey”, *IEEE Access* 8:197017-197046, 2020.

[10] L. Yala, P. A. Frangoudis, A. Ksentini, ”Latency and Availability Driven VNF Placement in a MEC-NFV Environment”, in *Proc. of IEEE GLOBECOM 2018.*

[11] R. Cziva, C. Anagnostopoulos, D. P. Pezaros, ”Dynamic, Latency-Optimal vNF Placement at the Network Edge”, in *Proc. of IEEE INFOCOM 2018.*

[12] Y. Nam, S. Song, J. Chung, ”Clustered NFV Service Chaining Optimization in Mobile Edge Clouds”, *IEEE Comm. Letters* 21(2):350-353, Feb. 2017.

[13] T. Taleb et al. ”On Multi-Access Edge Computing: A Survey of the Emerging 5G Network Edge Cloud Architecture and Orchestration”, *IEEE Communications Surveys and Tutorials* 19(3):1657-1681, 2017.

[14] H. Yu et al. ”DU/CU Placement for C-RAN over optical metro-aggregation networks”, in *Proc. of ONDM 2019.*

[15] GSTR-TN5G, ITU-T. Transport network support of IMT-2020/5G. 2018.

[16] A. Reznik et al. ”Cloud RAN and MEC: A perfect pairing”. ETSI White paper, 2018, no 23, p. 1-24.

[17] N. Makris et al. ”Employing MEC in the Cloud-RAN: An Experimental Analysis”, in *Proc. of the 2018 on Technologies for the Wireless Edge Workshop.*

[18] W. Diego, ”Evolution Toward the Next Generation Radio Access Network”, in *Proc. of IFIP Networking 2020.*

[19] S. K. Singh, R. Singh, B. Kumbhani, ”The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities”, in *Proc. of IEEE WCNCW 2020.*

[20] A. Ceselli, M. Premoli, S. Secci, ”Mobile Edge Cloud Network Design Optimization”, *IEEE/ACM Trans. on Networking* 25(3):1818-1831, 2017.

[21] M. Zheng et al. ”A Feedback Prediction Model for Resource Usage and Offloading Time in Edge Computing”, in *Proc. of 2018 Int. Conference on Cloud Computing*. Springer.

[22] I. Alghamdi, C. Anagnostopoulos, D. P. Pezaros, ”Time-Optimized Task Offloading Decision Making in Mobile Edge Computing”, in *Proc. of Wireless Days 2019.*

[23] J. Li et al. ”Deep reinforcement learning based computation offloading and resource allocation for MEC”, in *Proc of IEEE WCNC 2018.*

[24] X. Xu et al. ”Adaptive Computation Offloading With Edge for 5G-Envisioned Internet of Connected Vehicles”, *IEEE Transactions on Intelligent Transportation Systems*, early access, 2020.

[25] S. Namba, T. Warabino, S. Kaneko, ”BBU-RRH switching schemes for centralized RAN,” in *Proc. of 2012 Int. Conference on Communications and Networking in China*.

[26] B. Wu et al. ”A Game-Theoretical Approach for Energy-Efficient Resource Allocation in MEC Network”, in *Proc. of IEEE ICC 2019.*

[27] L. Chen et al. ”Deep mobile traffic forecast and complementary base station clustering for C-RAN optimization”. *Journal of Network and Computer Applications* 121:59-69, 2018.

[28] L. Chen et al., ”Complementary base station clustering for cost-effective and energy-efficient cloud-RAN,” in *Proc of 2017 IEEE SmartWorld.*

[29] S. Ntalampiras, M. Fiore, ”Forecasting Mobile Service Demands for Anticipatory MEC”, in *Proc. of IEEE WoWMoM 2018.*

[30] M. Bouet, V. Conan, ”Mobile Edge Computing Resources Optimization: A Geo-Clustering Approach”, *IEEE Transactions on Network and Service Management* 15(2):787-796, June 2018.

[31] A. Furno et al. ”Mobile Demand Profiling for Cellular Cognitive Networking”, *IEEE Trans. on Mobile Computing* 16(3):772-786, March 2017.

[32] IBM ILOG CPLEX 12.6 User Manual. IBM corp., 2013. Accessed on: 01-15-2020.