Robust Access Point Clustering in Edge Computing Resource Optimization

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

Robust Access Point Clustering in Edge Computing Resource Optimization

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

***Abstract*—Multi-access Edge Computing (MEC) technology has emerged to overcome traditional cloud computing limitations, challenged by the new 5G services with heavy and heterogeneous requirements on both latency and bandwidth. In this work, we tackle the problem of clustering access points in MEC environments, introducing a set of clustering models to be deployed at the pre-provisioning phase. We go through extensive simulations on real-world traffic demands to evaluate the performance of the proposed solutions. In addition, we show how MEC hosts capacity violation can be decreased when integrating access points clustering into the orchestration model, by investigating on solution accuracy when applied on held-out users traffic demands. The obtained results show that our approach outperforms two state-of-the-art algorithms, reducing both memory usage and execution time, by 46% and 50%, respectively, in comparison to a baseline algorithm. It surpasses the two methods in gaining control over MEC hosts capacity usage for different maximum achieved occupancy levels on MEC hosts.**

***Index Terms*—Multi-access Edge Computing, RAN virtualization, Access Point Clustering, MEC Resource Allocation.**

I. Introduction

THE Multi-access Edge Computing paradigm was initially developed for running services close to the end devices, in order to lower latency and improve user experience. Despite the fact that MEC infrastructure hosts can be densely distributed at the edge, resource limitation and robustness against traffic fluctuations are important challenges to handle by network service providers. New technologies such as Network Functions Virtualization (NFV) and Software-Defined Networking (SDN) were proposed in the last decade; their consideration into network architecture design is to push the technology barriers toward virtualized infrastructures at Radio Access Network (RAN) subsystems as well [1], including both the centralization of the control and the virtualization and softwarization of all involved network functions.

Radio function virtualization leads to additional flexibility in a segment historically more rigid than core networks, due to the lower importance of routing in these environments. This flexibility can help to meet the growing and unpredictable demands of mobile users, and also allows the use of standard hardware to reduce costs for Mobile Network Operators (MNOs) and delay capital expenditures. In addition, MEC technology allows to cope with users demand variation, since network reconfiguration becomes an easier operation to perform [2].

Indeed, while MEC infrastructures are recognized as a 5G key enabler, the reverse is also true: 5G can be considered a key enabler for MEC infrastructures, thanks to NFV technology [3]. The deployment of virtualization facilities in the access network, for 5G functions and RAN functions, can therefore favor the deployment of MEC infrastructure elements. The so-favored deployment of application servers near end users can increase user bitrates and reduce end-to-end latency [4].

Several MEC hosts deployment scenarios are considered in practice; at the macro Base Station (BS) site, at the core network or even between them at the so-called Central Offices (CO) and/or Points-of-Presence (PoP), depending on the service requirements [5].

In order to have a complete control of service deployment at the MEC infrastructure, the ETSI standards call for the development of orchestration service elements, with the aim of efficiently managing the available resources on MEC hosts. Hence, the automation of the aforementioned task is deemed as one of the important challenges to address. In other words, the inclusion of both the access and core network functions into MEC hosts requires to handle the task of assigning base-station Access Points (APs) to MEC facilities, while taking into consideration the computing capacity needed to handle AP traffic demands.

Scalability and robustness are major challenges that arise when dealing with MEC resource orchestration problems [6]. In this work, we shed a light on the scalability-robustness challenge by extending a MEC orchestration framework from [7]. We propose a portfolio of AP clustering algorithms, to integrate as a preprocessing phase to the actual orchestration problem, scheduling AP-to-MEC facility assignments over time. Our clustering algorithms are meant to show the flexibility we can benefit from MEC orchestration, while evaluating the impact in terms of robustness when considering heterogeneous demands profiles.

The main contributions of this work are as follows:

- We formulate a collection of AP clustering algorithms that we integrate into a MEC orchestration baseline algorithm [7], with the aim of reducing both execution time and memory usage, while integrating robustness criteria in the orchestration problem;
- We train the proposed frameworks using a real-world dataset, composed of demands collected at two different regions in France;
- We apply the resulting assignment plans on held-out data,

A preliminary version of this work is presented in [34].

NEH. Yellas, S. Boumerdassi, B. Maaz, S. Secci are with Cnam, Paris, France. Email: first-name.last-name@cnam.fr

A. Ceselli is with Università degli Studi di Milano, Italy. Email: alberto.ceselli@unimi.it

in order to assess how well the proposed assignment plans can adapt to access point changing demands by evaluating the violation of servers capacity;
- We finally compare our algorithms to two different approaches from the state-of-the-art. Numerical results show that our algorithms outperform these existing solutions in terms of robustness and computing performance.

The paper is organized as follows. In Section II, we give an overview about existing works. We define the AP-to-MEC assignment problem and the proposed formulations of the clustering models in Section III. We evaluate the performance of our algorithms in Section IV. We draw conclusions in Section V.

II. State of the art

In the following, we provide the necessary background on virtualization in edge computing infrastructures, network analytics and optimization.

A. Access Network Virtualization

MEC is one of the 5G key enabler technologies; its combination with NFV can be of a great benefit for mobile network operators, since the management operations can be held by the NFV architecture, more precisely via its Management and Orchestration (MANO) subsystem [8].

Several works exist in the area of MEC-NFV MANO, proposing algorithms or architectures taking advantage from the presence of MEC-NFV systems. For instance, [9] addresses the relationship between MEC and other technologies that are considered as 5G enablers such as NFV and SDN: the authors propose an architectural framework where an SDN controller is responsible for management operations in a MEC-NFV environment, hence being able to reconfigure the network stack to take into consideration orchestration decisions such as the assignment of APs 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].

In [14], a study was conducted on how a MEC infrastructure should be planned, that is, where MEC facilities should be placed, as a function of different MEC resource placement policies. A take-away result 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 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. The authors used real data volume information from a french mobile network.

From a radio-access perspective, 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/Cloud RAN (C-RAN), in 2010, where the innovation consisted in disaggregating AP facilities into two main units: Radio functions assured by Remote Radio Heads (RRHs) that are deployed on cell sites, and Base Band computation functions provided by BaseBand Units (BBUs). BBUs are then centralized at so-called BBU pools, hence taking advantage from the centralization for resource allocation and scaling [15]. More recently, the C-RAN evolution has been integrated in 5G systems, where a more dense deployment of base stations 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 can be split into two units, the Centralized Unit (CU) and the Distributed Unit (DU), in order to facilitate the virtualization and radio scheduling tasks [16], while the radio part is called Radio Unit (RU). Splitting radio processing functions is known as 'functional split' [15] and it enables to choose the functions that turn on cell sites and those that will be offloaded to CUs, with different splitting options [17], 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 [18]. In [19] the authors implement a MEC platform on the vRAN front-haul, and evaluate the QoS for end users for two different locations of MEC hosts. Authors in [1] propose a MEC vRAN joint design problem, introducing an optimization framework that aims to simultaneously find the best functional split of BSs and MEC service placement, taking into account flow routing. The integration of vRAN with SD-x system lead to the Open RAN (O-RAN) initiative, which has the goal to disaggregate software and hardware and to create open interfaces for more flexibility. Many O-RAN software releases exist today already [20]. In [21], the authors discuss the RAN evolution including the detailed description of the O-RAN reference architecture.

The standpoint we adopt in this article is the one of an operator running a MEC infrastructure the operator leverages on, for converging MEC applications and virtualized network functions. Hence APs are assigned to MEC hosts facilities in a dynamic way by means of MANO operations, leveraging on a programmable network stack between APs and MEC infrastructure, hence going largely beyond the legacy situation where APs 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 AP to MEC host assignments and related MEC switching operations.

B. Clustering in MEC Orchestration

Given the natural limitations of MEC facilities or hosts in terms of computing resources, resource orchestration is an important task to optimize its utilization, particularly important when considering the environmental footprint of edge computing [22]. Thus, operations that consist of re-assigning APs to other MEC hosts need to be deployed; this is needed to ensure that a number of desirable key performance indicators (KPIs) are met, as for instance maximizing resource utilization, increasing resiliency against network and computing

impairments, and increasing robustness against load variations in time and space.

Often, data analytics techniques are used in MEC design frameworks, so as to take into account dynamic load and communication channel states. For instance, stream or online data-analytics is needed in mobile computation offloading frameworks, where tasks offloading online decisions need to be made. In this direction, a feedback prediction model of average resource usage (RAM and CPU) and offloading time is proposed in [23]. In [24], the authors tackle the offloading decision for MEC applications where the performance is evaluated using real-world dataset. Reference [25] aims at offloading intensive computing tasks for energy saving by optimizing resource allocation, and [26] presents solutions for computation offloading in edge servers for internet of connected vehicles. Near-real-time or offloading data analytics is indeed often considered when addressing MEC design problems.

Leveraging on network data analysis is a common requirement for clustering techniques in C-RAN and MEC environments. In [27] [28] the authors propose a clustering scheme for APs, where APs of each cluster share the same data processing units that are centralized in datacenters to optimize costs and energy consumption in vRAN. Another example is [29], where the authors aim at predicting mobile traffic generated by a cluster of APs to anticipate MEC resource orchestration using real-world dataset. In [30] the authors propose an AP geo-clustering technique, while taking into account the spatial distribution of mobile traffic. The main goal is to define MEC clusters as a set of AP and users served by the same MEC host, so that the whole area is partitioned into MEC clusters, in order to offload the core network by maximizing intra MEC hosts communications. Similarly, in [7] where the authors apply the temporal clustering model proposed in [31] on 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 APs at a time slot: this allows to retrieve a reduced number of profiles, with a similarity assessment based on traffic volume and traffic distribution.

Recently, the authors in [32] propose an access point clustering scheme extending K-means to use 3D Hyperbolic distance, using access points locations and traffic demands; the algorithm groups together APs with complementary demands behaviors, that is, not grouping together APs with similar demand behavior to avoid idle states during off-peak hours. A similar approach is presented in [33], where the goal is to reduce the number of reconfiguration handovers, i.e., change of BBUs for base stations, also called later in this manuscript as switching operation.

Among all the previous works, either spatial diversity or load changes over time are considered in the clustering and AP-to-cloud facility assignment modeling. Taking both the spatial and temporal dimensions is however often not explicitly modeled. The resolution approach in [7] does model both time and space dimensions when determining an assignment plan by means of an optimization model; the approach consists in applying decomposition techniques, to obtain an extended formulation which is then optimized by a branch-and-price algorithm.

The authors in [34] extend the model in [7] anticipating the spatial clustering by means of preprocessing, so that both memory usage and execution time can be reduced using variable aggregation, hence granting important computing resource gains. In this paper, we extend the framework in [34] to include a more variate set of clustering fitness functions, comparing them in terms of reliability. Reliability is hereafter meant as the capacity of not exceeding the allocated computing capacity under varying traffic loads. We demonstrate how, thanks to the preliminary spatial clustering, we can integrate different orchestration flavors to make the MEC orchestration decision framework more robust against traffic fluctuations, while taking into consideration secondary performance indicators.

III. Orchestration Model

In our resource orchestration model we focus on assigning a set of APs belonging to a MNO infrastructure, to the available MEC facilities.

A	Set of all access points.
C	Set of all clusters of access points.
K	Set of all MEC hosts.
$T = \{1, \ldots, \tau\}$	Ordered set of τ time slots.
$T' = \{2, \ldots, \tau\}$	subset of T excluding the first time slot.
$T'' = \{1, \ldots, \tau - 1\}$	subset of T excluding the last time slot.
d_i^t	Parameter, represents the traffic demands of AP i at time slot t.
d_c^t	Parameter, represents the demands of the cluster c at time slot t.
${d'}_i^t$	Parameter, represents the variance of demands, of the same period of each week for AP i (for example, if the number of samples of the dataset corresponds to w weeks, ${d'}_i^t$ is then, the variance of all demands of AP i at the same time period t over all the w weeks).
$\bar{d}_i$	Parameter, represents the average of traffic demands of access point i through all the time slots.
c_{ij}	Parameter, identifier of the pairwise clustering criterion;
Q	Parameter, represents the capacity of each MEC host. All MEC hosts have the same capacity.
δ_{ij}	Parameter, represents the (complete link) distance between the two AP clusters i and j;
$\tilde{\delta}$	Parameter, represents the maximum distance between each couple of AP that belongs to the same cluster.
l_{nk}	Parameter, distance between the two MEC hosts n and k.
m_{ck}	Parameter, distance between the APs in cluster c and the MEC host k.
x_{ck}^t	Real non-negative variable, representing the fraction of APs belonging to cluster c that are assigned to MEC host k at time slot t.
y_{cnk}^t	Real non-negative variable, representing the fraction of APs belonging to cluster c that are switched from MEC host n to MEC host k at time slot t.
z_{ij}	Binary variable, takes value of 1 if APs cluster i is paired to APs cluster j to form a cluster, 0 otherwise.

TABLE I: Notations.

We decompose the MEC orchestration problem into two independent but connected phases: a pre-processing phase that

consists of grouping together APs into clusters based on some criteria, and a second phase that holds the assignment of the resulting clusters to the MEC hosts with available capacity. The assignment task takes into consideration the distance between access point to MEC hosts to which it is assigned to, thus representing the user costs, also called the assignment cost in the remainder of the paper. We propose to trigger assignment operations for each time period, where the duration of a time period can go from few seconds up to few hours. Since MEC hosts capacity is limited, and due to users traffic variations, VM resource migration can be required; these operations yield a deployment cost of the network, referred to as switching cost.

In the following, we introduce the complete orchestration framework in details. First, we present the AP-to-MEC assignment problem as a Mixed Integer Linear Program (MILP). Second, we introduce the preproccessing phase with the clustering approaches.
Table I lists the notations used.

The proposed framework is based on a training process that consists of using historical data i.e., access point traffic demands, to identify the parameters of a model. We define two different models, one model that groups together APs based on a given criterion at the clustering phase, and a second one that assigns the resulting clusters to the available MEC hosts at the orchestration phase.

A. AP-to-MEC assignment

The goal of the orchestration framework is to assign a group of APs belonging to a given geographic area, to a set of MEC hosts. We consider the possibility of having a cluster composed of only one AP, in this case, the model represents a single AP assignment problem, which refers to the baseline algorithm.
Let us consider a user equipment (UE) connected to an AP; the assignment operation of its traffic to a given MEC host yields a cost defined by the access latency. We assume that hosting demands of a given AP on a MEC host consists of allocating one Virtual Machine (VM) for each UE. On the other hand, and unlike traditional datacenters, MEC hosts have limited capacity, thus switching AP demands from a MEC host to another is sometimes requested in order to cope with traffic variation. Given the lower traffic granularity at MEC hosts, switching operations 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.

We adapt the model in [7], that we present by equations from (1) to (7).

$$\min \sum_{t\in T}\sum_{c\in C}\sum_{\substack{(n,k)\in\\ K\times K}} d_c^t l_{nk} y_{cnk}^t + \sum_{t\in T}\sum_{c\in C}\sum_{k\in K} d_c^t m_{ck} x_{ck}^t \quad (1)$$

$$\text{s.t.} \sum_{c\in C} d_c^t x_{ck}^t \le Q \qquad \forall k\in K, \forall t\in T \quad (2)$$

$$\sum_{k\in K} x_{ck}^t = 1 \qquad \forall c\in C, \forall t\in T \quad (3)$$

$$x_{ck}^t = \sum_{n\in K} y_{cnk}^t \qquad \forall c\in C, \forall k\in K, \forall t\in T' \quad (4)$$

$$x_{ck}^t = \sum_{n\in K} y_{ckn}^{t+1} \qquad \forall c\in C, \forall k\in K, \forall t\in T'' \quad (5)$$

$$x_{ck}^t \in [0,1] \qquad \forall c\in C, \forall k\in K, \forall t\in T \quad (6)$$

$$y_{cnk}^t \in [0,1] \qquad \forall c\in C, \forall n,k\in K, \forall t\in T \quad (7)$$

The objective formulated in (1) aims to find an assignment plan for each cluster of APs to the set of MEC hosts for each period of time, where each cluster can be composed of one or multiple APs. We aim to minimize both assignment and deployment costs.

In (2) we ensure that the overall demands assigned to a MEC host must not exceed its capacity. Constraints (3), (6) and (7) give the possibility to assign a cluster of APs to one or more MEC hosts for each time slot. In fact, in this case, the AP demands can be split and assigned to different MEC hosts. If we have the nearest MEC host with a very small available capacity, the proposed solution allows us to assign the remaining demands to other MEC hosts.

Constraints (4) and (5) ensure the balancing of demand flows for each cluster, each MEC host and each time slot. More precisely, the right-hand side of (4) represents the overall fraction of demands of APs belonging to cluster c *incoming* to MEC host k at time t, possibly being switched from other MEC hosts; this needs to be consistent with the value of the corresponding x_{ck}^t variable. Similarly, the right-hand-side of each (5) represents the overall fraction of demands of APs belonging to cluster c *outgoing* from MEC host k at time t, possibly being switched to other MEC hosts. Since the left-hand-sides are identical, (4) and (5) impose incoming and outgoing demand fractions to be equal.

B. Multi-objective AP clustering

The idea of performing access points spatial clustering as a preprocessing to the optimization problem is to, from the one hand, take more robust decisions with respect to traffic variations by grouping together APs that satisfy a given performance target and, from the other hand, decrease both execution time and memory space of (1)-(7) thanks to variable aggregation and constraints reduction.

We propose an extension of the spatial clustering model proposed in [34]. In order to ease reaching optimal configurations we set an iterative pairwise access point clustering instead of grouping an indeterminate number of APs together.

The motivation is to use simpler combinatorial models by merging two APs as an AP pair, based on a different set of criteria. Depending on the scales of the problem (i.e., number of APs), this pairwise clustering can be iterated so as to group,

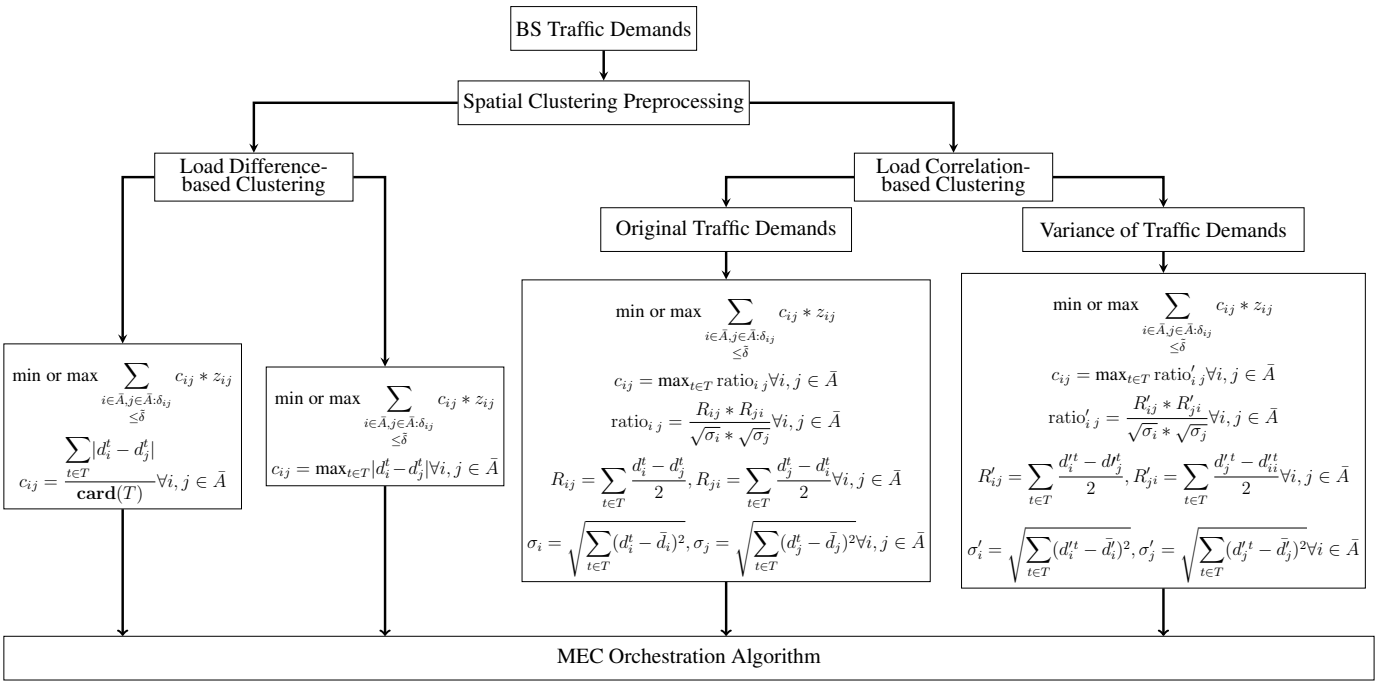


Fig. 1: MEC Orchestration Framework Options.

at a second stage, two pairs of APs within a cluster; as so on so forth, if needed, further hierarchical clustering can happen. We then apply the orchestration decision on the set of resulting clusters.

The clustering criterion is meant to allow granting a robustness flavor to the orchestration model, namely in terms of robustness against load variation in time. The goal is to reduce MEC host capacity violations.

The generic clustering mathematical formulation is as follows:

$$\text{min or max} \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \tag{8}$$

$$\text{s.t.} \sum_{j\in\bar{A}} z_{ij} + z_{ji} = 1 \quad \forall i \in \bar{A} \tag{9}$$

$$z_{ij} = 0 \quad \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \tag{10}$$

$$z_{ij} \in \{0,1\} \quad \forall i,j \in \bar{A}$$

The set $\bar{A}$ contains one element for each AP cluster. The objective function in (8) aims at minimizing (maximizing, respectively) the clustering cost value expressed using the criterion parameter c_{ij} which defines the degree of similarity or difference according to which elements (access points or clusters of access points) are grouped in the same cluster. More precisely, it is given by the sum of costs for those pairs of AP clusters which are joined.

Constraints (9) ensure that each cluster $i \in \bar{A}$ is paired to exactly one other cluster (either i is paired to some j, or j exists, which is paired to i). Constraints (10) ensure that such a pairing is made only among AP clusters whose distance does not exceed a given threshold. Note that having z_{ij} =1 implies the creation of a cluster that merges APs (resp. groups of APs) i and j together, in which case all other z variables involving i are set to 0, to technically keep consistency (i.e. all nodes with degree one) on the directed graph model we employ.

Initially, $\bar{A} = A$. That is, single APs are paired. Then, single elements of $\bar{A}$ are replaced by the pairwise clusters which are formed. After all replacements are made, new cluster criterion parameters and distances are computed for the elements of $\bar{A}$, and the clustering process is iterated. For each iteration, we compute the distance between each couple of APs that belong to different clusters. Only clusters with distances that are lower than the threshold are grouped together. We propose two classes of criteria as in the following.

1) Clustering-based on load differences

This class of criteria aims at grouping together access points depending on the demand differences without taking into consideration AP demand profiles. We propose four different criteria:

- MIN-MAX. To reduce the absolute value of the difference in demands for each couple of APs during all the time periods. The goal is to obtain clusters with similar demands for each period of time. In fact, this criterion represents an enhanced version of the single-one used in [34].
- MIN-SUM. To minimize the average of differences in demands for all the time periods, for each couple of APs belonging to the same cluster. The goal is to group together APs where the average of their demands is minimized through all time slots.
- MAX-MAX. To group each couple of access points where the difference between their demands for each time

slot is maximized. This criterion yields clusters of APs with highly different traffic demands.
- MAX-SUM. To maximize the average of demand differences of each couple of APs and for all the time periods in order to produce clusters of APs that behave differently through time.

2) Clustering-based on load correlation

This class of criteria uses different forms of correlation among AP load, taking into consideration the AP demand profiles. We propose four different criteria:

- MIN-CORR. To find negatively correlated couples of APs to group them together into the same cluster. To do so, we propose an expression that defines the relationship between demands of each couple of APs expressed as a ratio. Minimizing this ratio leads to a maximization of the difference in APs demand profiles.
- MAX-CORR. To search for couples of APs where the correlation between their demands is maximized through time in order to have positively correlated demands in the same cluster. In this way, the resulting clusters group together APs with highly similar demand profiles.
- MIN-CORR-VAR. To have couples of APs with negatively correlated variance of demands. In order to achieve this objective, we minimize the correlation of the variance of demands of APs i and j. An access point with high demand variance will be then merged with an AP having low demand variance.
- MAX-CORR-VAR. To maximize the correlation of demand variance between couples of APs belonging to the same cluster, with a view to have each couple of AP with demand variance that are positively correlated grouped into the same cluster. Thus, an AP having high variance in its demands through time will be merged with an AP of the same demand profile.

We embed in Figure 1 the mathematical expression for each criterion. The complete mathematical formulations are given in the Appendix.

IV. Experimental Results

In the following, we first describe the dataset, then we provide a numerical evaluation of our MEC orchestration approaches using different metrics while comparing them to two approaches from the state-of-the-art.

We solve the resource orchestration problem using the eight clustering variants in Section III-B and the following additional algorithms:

- 'MECA': solving the reference orchestration model without spatial clustering, i.e., (1) to (7), this refers to the baseline MEC assignment algorithm, which we refer to in the following as the benchmark;
- 'HYPERBOLIC': solving the reference orchestration model with the HYPERBOLIC-KMEANS spatial clustering at the state-of-the-art [32];

We also refer to 'MECA-CS^{+}' as solving the reference orchestration model with MIN-MAX spatial clustering in the pre-processing phase (enhancement of the MECA-CS algorithm in [34]);

A. Dataset and simulation setting

We used a real traffic dataset from a national mobile network made available in the frame of the French ANR CANCAN (Content and Context based Adaptation in Mobile Networks) project. The dataset gives us access points downlink volume information every 10 minutes for a period of three months in 2019, for Paris and Lyon metropolitan area networks. The collection process takes into consideration both 3G and 4G connections and records data on a per-user basis that are aggregated at the AP level. Paris dataset contains a larger amount of demands when compared to Lyon dataset. In fact, we have chosen 1908 base stations for the area of Paris and 332 for the area of Lyon [1] We split our datasets into two parts: we used the first two-thirds to train both the clustering and orchestration models, and the remaining third, i.e., held-out data, to evaluate the quality of the solution.

For the clustering algorithms described in Section III-B, we perform (i) for the Lyon dataset, only once the pairwise clustering, so clusters of two APs are formed, while (ii) for the Paris dataset, we perform two pairwise clustering iterations, to decrease the memory usage and execution time otherwise too high given the higher number of APs. Indeed, the number of hierarchical pairwise clustering iterations can be customized based on the trade-off between execution time and assignment cost, as discussed hereafter.

For the simulations, we generate MEC facilities locations using a variant of the K-means clustering method, called weighted K-means, such that each MEC host location is the centroid of a given group of APs; the weight is represented using APs demands dispersion. In this way, MEC hosts positions are generated depending on the access points demands profiles. We fix the number of MEC facilities to 20 servers for both Lyon and Paris datasets. Then, we randomly generate 10 different configurations for MEC hosts locations with the aim of producing different inputs to train our MEC orchestration algorithm.

We implemented our model in AMPL (A Mathematical Programming Language) [35] using CPLEX as the linear solver [36]. We run our algorithms on an Ubuntu Server 14.04 LTS virtual machine with 64 GB of RAM and 8 $\times$ 2.5 GHz CPU cores.

B. Numerical evaluation

We analyze the quality of the solution based on the following metrics:

- the memory usage and execution time: represented by the execution time (s) and maximum memory usage (GB);
- the assignment and switching costs;
- the total cost gap convergence in percentage, against the benchmark;
- the distribution of BS-to-MEC distances during the period of the training (refers to a representative week) in kilometers;
- the number of switching operations.

We use for this evaluation Lyon dataset.

In Table II, we summarize the average $\pm$ standard deviation of the aforementioned metrics results.

[1]The content of the dataset is private, additional details cannot be provided.

Models	Execution Time (s)		Memory (GB)[2]	Assign. Cost e^{+10}	Switch. Cost e^{+9}	Gap
	Orchestration	Clustering				
MECA-CS^{+}	753.84 ±160.50	0.902 ±0.034	7.99	7.79 ±0.169	3.74 ±0.53	1.31 ±0.02
MIN-SUM	838.44 ±208.75	1.727 ±0.049	7.99	7.65 ±0.139	3.69 ±0.58	1.29 ±0.03
MAX-MAX	1066.5 ±283.41	0.340 ±0.019	8.002	6.75 ±0.252	4.07 ±0.5	1.14 ±0.01
MAX-SUM	1003.9 ±191.74	0.621 ±0.028	8.002	6.60 ±0.249	4.05 ±0.52	1.12 ±0.08
MIN-CORR	969.65 ±243.9	0.688 ±0.034	8.002	6.57 ±0.239	4.08 ±0.56	1.12 ±0.01
MAX-CORR	576.63 ±124.80	1.288 ±0.059	7.99	7.42 ±0.147	3.63 ±0.61	1.24 ±0.03
MIN-CORR-VAR	900.00 ±160.11	0.552 ±0.018	8.002	6.63 ±0.23	4.00 ±0.51	1.13 ±0.01
MAX-CORR-VAR	915.83 ±235.25	0.716 ±0.022	7.99	7.59 ±0.18	3.59 ±0.48	1.27 ±0.03
HYPERBOLIC	376.97 ±79.73	787.9 ±46.65	7.90	6.82 ±0.25	4.11 ±0.51	1.16 ±0.013
MECA (benchmark)	2025.4 ±473.68	-	15.80	5.84 ±0.224	3.90 ±0.55	1

TABLE II: Average ± standard deviation of the evaluation metrics for the different algorithms.

Figures 2 to 6 depicts the distributions for each of the aforementioned metrics. We now draw our observations on these results as follows.

1) Execution time

Figure 2 reports the distribution of execution times experienced with each case. We can remark that the slowest algorithm is the benchmark (MECA) solution where the average execution time exceeds half an hour. In fact, applying the clustering algorithm in the preprocessing, as done for the other cases, helps reduce the execution time as proven in [34].

We can also notice that the MAX-CORR has the lowest execution time among the eight proposed algorithms, followed by MECA-CS^{+}. The highest execution times are yield by the algorithms based on load difference which are MAX-MAX and MAX-SUM, respectively.
The HYPERBOLIC clustering is globally the fastest one with an average of 376 s; due to the fact that the number of APs per cluster is not fixed, which generates clusters with a higher number of APs compared to the other models, hence reducing both the memory usage and execution time, as shown in [34]. However, this gain in the MEC-to-AP assignment phase comes at the expense of a much longer pre-processing phase time as shown in Table II. This time, moreover, increases with the dataset size: tests with the Paris dataset show that HYPERBOLIC clustering needs more than 24 hours to be trained versus few dozen of seconds for the other cases performing iterative pairwise clustering.

2) Maximum RAM usage

From Table II, all the clustering-based algorithms are comparable in this respect; we record a slight difference between the HYPERBOLIC and all the other cases. We can remark that the benchmark algorithm is the most memory-consuming (almost the double than the others).

We can also notice that the standard deviation is null (therefore omitted), i.e., changing the MEC hosts positions does not affect the maximum memory usage.

[2]The maximum memory usage metric was the same for each of the MEC hosts configurations, because we do not change the data size for each of the proposed configurations.

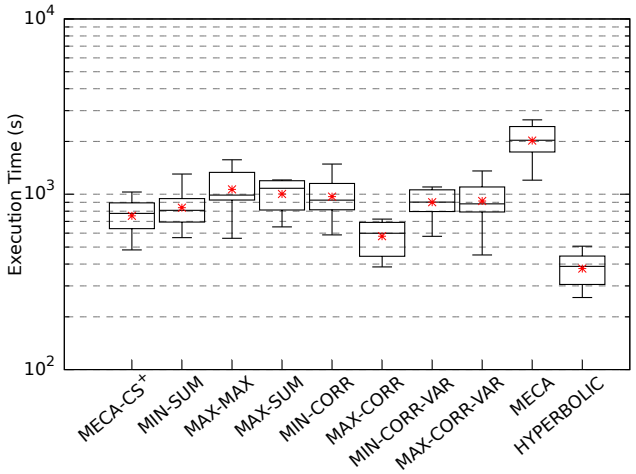
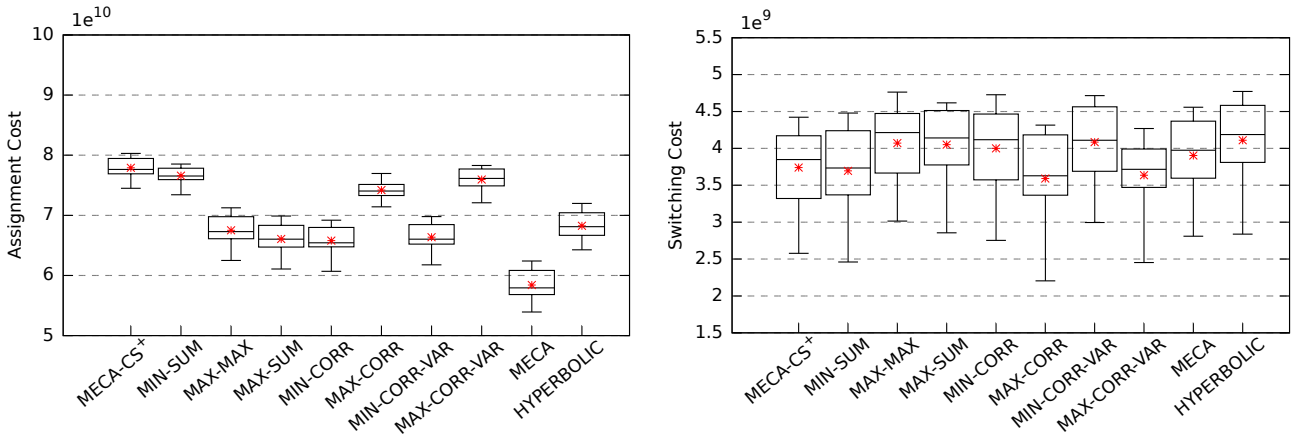


Fig. 2: Orchestration execution time.

3) Assignment and switching costs

Figure 3 plots the distribution of both the assignment and the switching costs. Figure 3a shows that the benchmark algorithm (MECA), which is clustering-preprocessing free, has the lowest assignment cost. Indeed, as shown in [34], solving the orchestration problem for a set of clusters of APs forces the algorithms to propose the same orchestration plan for the AP belonging to the same MEC host, where some APs will not be assigned to the closest MEC host. Regarding the other algorithms, solutions grouping together APs with different demands through time have the lowest average assignment costs, i.e., MIN-CORR, MIN-CORR-VAR and MAX-SUM, whereas, the worst cases refer to the algorithms based on both MIN-SUM and MIN-MAX, which group together similar AP traffic demands. This shows that assigning APs with complementary demands profiles to the same MEC hosts reduces the assignment costs.

Figure 3b presents the distribution of switching costs. MAX-CORR-VAR and MAX-CORR, which group together APs with maximized correlation, have the smallest switching costs with an average of $3.59 \times e^{+9} \pm 0.48 \times e^{9}$ and $3.63 \times e^{+9} \pm 0.61 \times e^{9}$, respectively. On the other hand, HYPERBOLIC clustering has the highest switching costs values with an average of $4.11 \times e^{9} \pm 0.51 \times e^{9}$. This shows that grouping APs with similar demand profiles allows having convenient assignment plans that last for longer periods of time compared to the other algorithms.



(a) Assignment costs.

(b) Swtiching costs.

Fig. 3: Costs distribution

4) Convergence gap

Figure 4 shows the distribution of the convergence gap obtained by each of the aforementioned algorithms. By convergence gap we indicate the relative difference between the best solution (expressed by the total cost) produced by each of the clustering-based algorithms and the benchmark, within a given execution time limit.

The orchestration algorithms based on models that maximize the difference in demands (resp. minimize the correlation coefficient of APs of the same cluster) have the lowest total costs. We can also notice that these algorithms have the lowest assignment costs but not necessarily the lowest switching cost. This confirms that assigning APs with complementary demand profiles to the same cluster produces lower assignment costs.

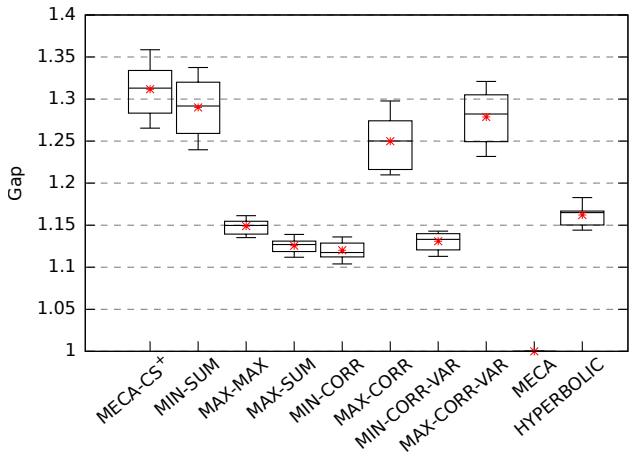


Fig. 4: Gap ratio against Benchmark.

5) BS-to-MEC distance

Figure 5 presents the distribution of access points to MEC facilities distances aggregated during a period of 1 week and for all the proposed MEC locations, for each of the algorithms. Benchmark gets the lowest distances, that is, the number of APs that are assigned to their closest MEC hosts is bigger compared to the other algorithms.

The other approaches are comparable, except HYPERBOLIC that gives slightly lower distances.

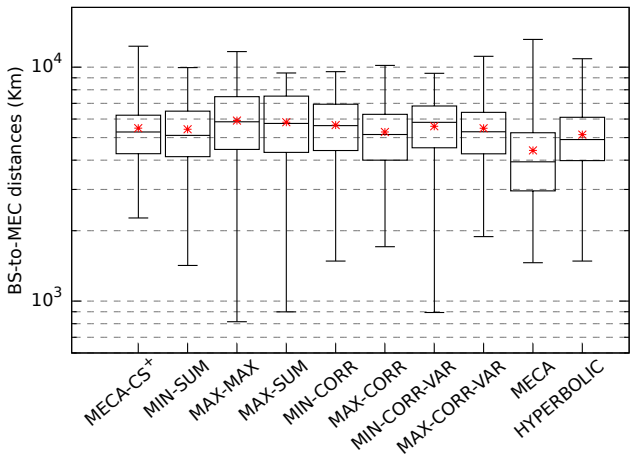


Fig. 5: Distribution of the AP-to-MEC hosts distances.

6) Switching operations

Figure 6 shows the number of switching operations on each MEC host during a period of 1 week. The benchmark gives the lowest number, followed by the algorithms that minimize the difference in traffic demands, i.e., MIN-MAX and MIN-SUM, while HYPERBOLIC yields the highest number of switching operations. In fact, this does not imply a lower switching cost, as can be seen in Figure 3b because it depends on the switched traffic demand of each of the base stations.

As a final remark on this part, it is worth stressing that assignment cost and execution time metrics are negatively correlated: the lower the execution times, the higher the assignment costs. For example, MAX-MAX requires the highest

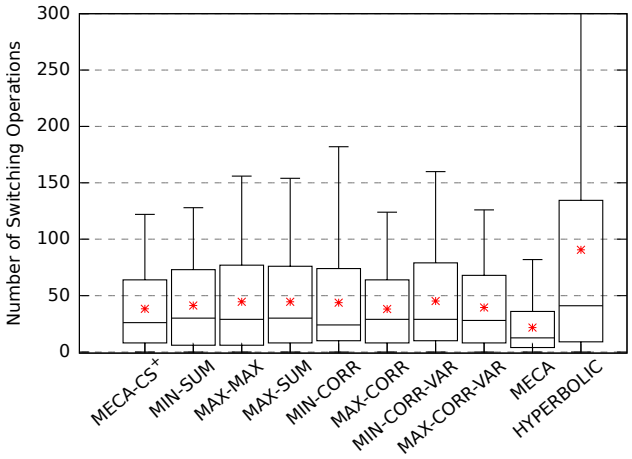


Fig. 6: Distribution of the number of switching operations.

execution time when compared to the proposed solutions, but it yields lower assignment costs than MAX-CORR ($6.75e^{+10}$ vs $7.42e^{+10}$), where this latter represents the fastest approach. We can also notice that solutions generating the lowest switching costs, require less running time.

C. Robustness Analysis

In this part, we evaluate the robustness of the proposed solutions over time, i.e., the impact of applying the resulting orchestration plans on held-out (test) data. For that purpose, we evaluate the violation of MEC host capacity produced by each solution when applied to both the Lyon and Paris datasets.

For this purpose, we use the set of parameters initially defined in [7] in order to estimate the degree of controlling MEC hosts capacities by each algorithm. We assess the performance for different maximum occupancy rates achieved by MEC hosts, while changing the capacity sizes. We choose high utilization levels in the interval between 71% to 95.5%. The analyzed parameters are as follows:

Capacity Overload Average

$$\sum_{t\in T}\sum_{k\in K}\max\left\{\sum_{c\in C} d_c^t * x_{ck}^t - Q, 0\right\} /(Q\times \mid T \mid \times \mid K \mid) \tag{11}$$

This index (called SUM-SUM in [7]) gives the average of the demands exceeding the MEC host capacity over all the time periods.

Violation Rate

$$\mid \{(t,k) : \sum_{c\in C} d_c^t * x_{ck}^t - Q \geq 0, \forall t \in T, \forall k \in K\} \mid /(\mid T \mid \times \mid K \mid) \tag{12}$$

This index (called SUPPORT in [7]) computes the percentage of number of violations that occurred over all periods of times.

Excess Demand Average

$$\frac{\sum_{c\in C, k\in K: d_c^t * x_{ck}^t - Q \geq 0} \sum_{c\in C} (d_c^t * x_{ck}^t - Q)/Q}{\mid \{(t,k) : \sum_{c\in C} d_c^t * x_{ck}^t - Q \geq 0, \forall t \in T, \forall k \in K\} \mid} \tag{13}$$

This index (called SUM-SUM-SUPPORT in [7]) is used to show the relationship between the amount of excess demands and the total number of violations.

Note that our take-away on this aspect is that, in an operational carrier grade environment, when the actual assignment operation yields a capacity violation, the service is not interrupted but runs instead in a degraded mode, given the actual computing scheduler management of peak overloads by means of resource sharing policies.

We represent the obtained results using the percentage gap with respect to the benchmark, computed as the ratio between the (i) difference between a given algorithm value and the benchmark value and (ii) the benchmark value.

For the sake of readability, we report only the results for the fastest algorithms from the previous analysis: MECA-CS$^+$, MIN-SUM, MAX-CORR, MIN-CORR-VAR and HYPERBOLIC.

1) Lyon dataset

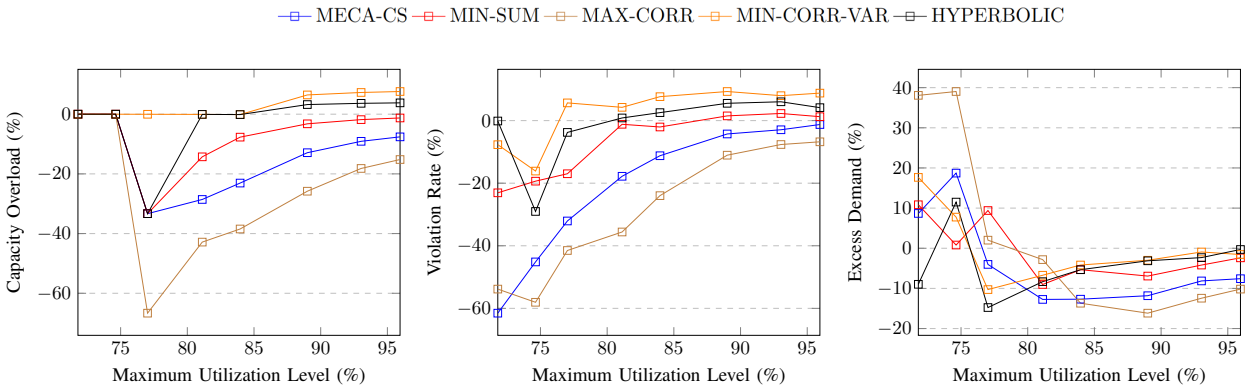
Figure 7 reports the robustness gaps with respect to the benchmark, as a function of the maximum MEC host utilization, for the five aforementioned approaches, for the Lyon dataset.

In terms of capacity overload (Figure 7a), we can observe that:

- When the maximum utilization is less than 75%, the gap is null: all the algorithms yield the same overload as the benchmark.
- For a maximum utilization level up to 84%, MIN-CORR-VAR yields the same overload as the benchmark and then increases it for higher utilization levels, whereas, all the other algorithms decrease the overload; the highest difference happens with an utilization level equal to 77% with a decrease of 68% for MAX-CORR algorithm and 34% for the two others (MECA-CS$^+$ and MIN-SUM).
- For a maximum utilization greater than 84%, cases merging complementary APs profiles (MIN-CORR-VAR and HYPERBOLIC) increase the capacity overload. This is reduced when using models that group together APs with similar demands (MECA-CS$^+$, MIN-SUM and MAX-CORR), i.e., these algorithms give an assignment with better robustness.
- MECA-CS$^+$, MIN-SUM and MAX-CORR lower the capacity overload when compared to the benchmark for each of the utilization levels greater than 75%. This shows that clustering AP demands in the preprocessing phase does not necessarily reduce the server capacity overload. In fact, the solution quality depends on the clustering criterion.

In terms of violation rate (Figure 7b), we can observe that:

- Except for HYPERBOLIC that yield the same number of violations for a maximum occupancy level equal to 71%, the other algorithms produce fewer violations when compared to the benchmark for a maximum utilization level less than 75%.
- MIN-CORR-VAR and HYPERBOLIC yield a higher number of capacity violations when the maximum occu-



(a) Capacity Overload (average). (b) Violation Rate (c) Excess Demand (average)

Fig. 7: Robustness results as a function of the maximum MEC host utilization - Lyon dataset.

pancy ratio is greater than 77% and 81%, respectively, when compared to the benchmark. On the other hand, MIN-SUM produces less violations when the maximum level of utilization is less than 87%.

- Both MECA-CS$^+$ and MAX-CORR have a better violation robustness, i.e., they produce less violations through all the maximum levels of occupancy when compared to the benchmark and to the other algorithms. This can be justified by the fact that these two approaches propose assignment patterns for groups of APs with less variations on their total demands through time.

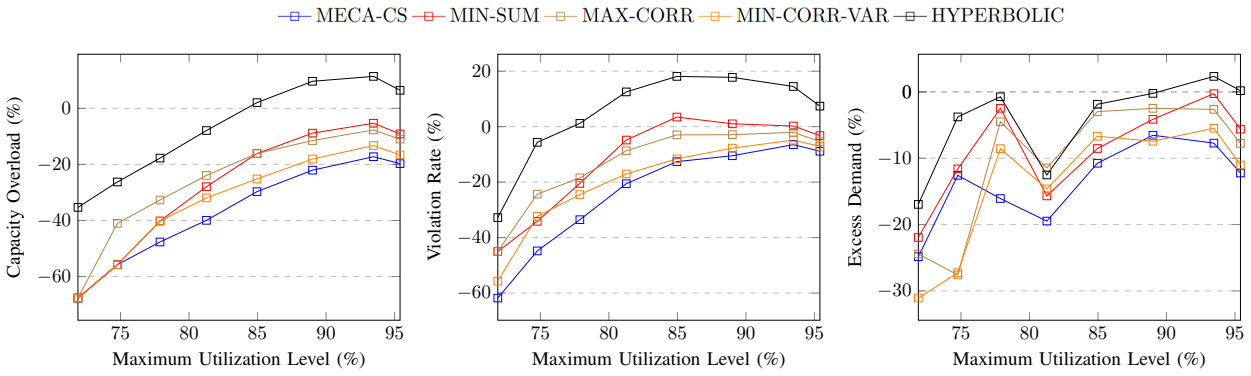
Since it is easier to fit AP demands separately into their closest MEC hosts when training the model, the likelihood of reaching full MEC hosts capacity increases too when compared to clusters-to-MEC assignment. However, hosting high outlier demands generated in the test set data will be more difficult in the latter case above. On the other hand, the assignment of traffic demands using clustering-based algorithms during the training phase, tend to be more balanced as assignment operations should be carried out for the demands of all APs belonging to the same cluster. This can justify the fact that clustering APs before their assignment to MEC hosts can help in reducing servers capacity violation when applied on held-out data.

In terms of capacity excess (Figure 7c), we can remark that: High values of the excess demands average refer to relatively low violation number compared to the amount of demand overload. Similarly, small values of the excess demands indicate that there is a relatively large number of capacity violations when compared to capacity overload.For a maximum capacity utilization between 71% and 75% all the models produce higher capacity excess on average when compared to the benchmark, except for HYPERBOLIC. For a maximum occupancy less than 76% and 79% when using MECA-CS$^+$, and MIN-SUM and MAX-CORRELATION, respectively, even though the capacity overload and the violation rate are both reduced compared to the benchmark, they produce higher average of excess of demands. This can be explained by the fact that these cases are generating relatively high excess demands, compared to the number of violations which makes the ratio bigger (in contrast to the benchmark, where the number of violations tend to be relatively low compared to capacity excess). For a level of occupancy greater than 79%, the excess is reduced when using clustering approaches, except for HYPERBOLIC that produces the same average of excess demands as the benchmark for the highest maximum utilization level. This can be explained by the fact that the clustering algorithms yield a proportionally higher number of violations in comparison to the demands excess (in contrary to the Benchmark that produces relatively high capacity overloads compared to the its number of violations).
In fact, applying the resulting assignment on held-out demands shows that training the orchestration model with Lyon dataset using clusters built based on the similarity of their traffic demands produces assignment and switching plans that are more suitable for traffic fluctuations. Thus, MECA-CS$^+$, MIN-SUM and MAX-CORR outperform both the algorithms from the state-of-the-art where they reduce the capacity excess of demands for any maximum occupancy level.

2) Paris dataset

Figures 8 depict the robustness metrics results for the Paris dataset. In terms of capacity overload (Figure 8a) we have the following observations:

- Except for HYPERBOLIC, all other algorithms reduce the capacity overload of the MEC hosts for each given utilization level.
- The highest reduction occurs at a utilization of 71%, with a decrease of 68% for the observed approaches. On the other side, the lowest decrease is recorded when the utilization level is equal to 93.5% with a reduction of 18% for MECA-CS$^+$, 15% for MIN-CORR-VAR, 8% for MAX-CORR and 5% for MIN-SUM.
- When the maximum utilization is less than 85%, HYPERBOLIC lowers the overload demands compared to the benchmark and increases it for higher utilization levels, where it achieves the max growth of 11% when this latter is equal to 93%.



(a) Capacity Overload (average). (b) Violation Rate (c) Excess Demand (average)

Fig. 8: Robustness results as a function of the maximum MEC host utilization - Paris dataset.

Looking at the violation rate (Figure 8b), we can assert that:

- MECA-CS$^+$, MAX-CORR and MIN-CORR-VAR reduce the number of violations for all the given utilization levels. As already explained, using clustering models at the preprocessing phase leads to finding the same assignment plan for each group of AP; fitting the total amount of their demands all together is then more difficult compared to single AP assignment problem. Thus, proposing solutions that ensure balanced availability on MEC hosts capacity is more likely to happen with cluster-based assignment. In addition, grouping access points based on their demands can help to assign groups of AP presenting less variance on their demands through time, which can justify the fact that some clustering-based algorithms outperform others.
- When the maximum utilization is less than 83%, MIN-SUM decreases the number of violations in comparison to the benchmark, and increases it for an utilization between 83% and 93%. As mentioned before, this latter produces less demands excess for such utilization levels. This can be justified by the fact that this approach generates high violations with low excess of demands.
- HYPERBOLIC decreases the number of violations by at most 30% when the servers maximum usage level is equal to 71%. On the other hand, for a capacity utilization level greater than 78%, the number of violations raises when compared to the Benchmark.
  When the occupancy level is greater than 93%, we get a decrease in the number of violations for all the cases when compared to the benchmark.

Finally, the results in Figure 8c show that:

- Even though both the capacity overload and the violation number difference percentage values are monotonously increasing, there is a variability in the average of excess demands. This can be explained by the fact that this ratio is related to the amount of excess demands per each violation. Having a high violation number with low demand overload yields a larger average of excess demands and vice versa.
- All the clustering algorithms decrease the average of excess demands for all maximum utilization levels, except for HYPERBOLIC when this latter is greater than 87%.

All in all, the results provided in Figure 8 show the contribution of iterated pairwise clustering when integrated to the orchestration algorithm and trained using traces from a highly heterogeneous metropolitan area network as the Paris one. When the robustness against traffic fluctuations is higher, the number of violations and the excess of capacity are reduced even for the highest MEC infrastructure utilization levels.

It is worth mentioning that, overall, MECA-CS$^+$ and MAX-CORR provide the most accurate results when applied to both Lyon and Paris datasets.

V. Conclusions and Perspective

In this paper we proposed a collection of access point clustering models aiming at grouping base station APs in a robust way with respect to their assignment to edge computing facilities. By leveraging on state-of-the-art orchestration algorithms for the assignment problem, we have evaluated the performance of the proposed approaches using real-world traffic demands, while comparing them to other two state-of-the-art approaches. Through extensive simulations and evaluation in terms of different performance metrics, we show under which conditions the algorithms we propose reveal to be the most efficient ones. We further compared the four fastest clustering algorithms with two state-of-the-art algorithms in terms of robustness against traffic fluctuations, and using two different city datasets. Among many important observations showing the general superiority of our approaches with respect to the state-of-the-art ones, a promising finding is that the robustness of the algorithms is higher with larger traffic source diversity. As the proposed clustering approaches can be applied on other assignment frameworks than the one used in this work, further works may concentrate on their usage for combined MEC and vRAN orchestration, including multiple decision points such as in functional splitting, also addressing different objectives such as based on additional quality-of-service criteria.

Acknowledgment

This work was funded by the ANR CANCAN (https://cancan.roc.cnam.fr; ANR-18-CE25-0011), the H2020

AI@EDGE (https://aiatedge.eu; grant nb. 101015922) and the AMI-5G ENE5AI projects. We would like to thank Cezari Ziemlicki from Orange for his support and Prosper Chemouil for his effort to review the manuscript.

REFERENCES

[1] A. Garcia-Saavedra et al. "Joint Optimization of Edge Computing Architectures and Radio Access Networks", *IEEE J. on Selected Areas in Communications* Nov. 2018.

[2] J. Zhang et al. "Joint Optimization of Virtual Function Migration and Rule Update in Software Defined NFV Networks," GLOBECOM 2017.

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

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

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

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

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

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

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

[10] L. Yala et al. "Latency and Availability Driven VNF Placement in a MEC-NFV Environment", in *Proc. of IEEE GLOBECOM 2018*.

[11] R. Cziva et al. "Dynamic, Latency-Optimal vNF Placement at the Network Edge", in *Proc. of IEEE INFOCOM 2018*.

[12] Y. Nam et al. "Clustered NFV Service Chaining Optimization in Mobile Edge Clouds", *IEEE Comm. Letters* 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* 2017.

[14] A. Ceselli et al. "Mobile Edge Cloud Network Design Optimization", *IEEE/ACM Trans. on Networking* 2017.

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

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

[17] DA SILVA COELHO et al. "On the impact of novel function mappings, sharing policies, and split settings in network slice design," in *CNSM 2020*.

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

[19] 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*.

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

[21] S. K. Singh et al. "The Evolution of Radio Access Network Towards Open-RAN: Challenges and Opportunities", in *Proc. of IEEE WCNCW 2020*.

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

[23] 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*.

[24] I. Alghamdi et al. "Time-Optimized Task Offloading Decision Making in Mobile Edge Computing", in *Proc. of Wireless Days 2019*.

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

[26] 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.

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

[31] A. Furno et al. "Mobile Demand Profiling for Cellular Cognitive Networking", *IEEE Trans. on Mobile Computing* March 2017.

[32] H. DJEDDAL et al. "Hyperbolic K-means for traffic-aware clustering in cloud and virtualized RANs," *Computer Communications 2021*.

[33] D. Naboulsi et al. "On User Mobility in Dynamic Cloud Radio Access Networks," *IEEE INFOCOM 2018*.

[34] N. -E. -H. Yellas et al. "Complexity-Performance Trade-offs in Robust Access Point Clustering for Edge Computing," *2021 17th International Conference on the Design of Reliable Communication Networks*, doi: 10.1109/DRCN51631.2021.9477332.

[35] IBM ILOG AMPL Version 12.2 User's Guide.

[36] IBM ILOG CPLEX 12.6 User Manual. IBM corp., 2013.

Nour-El-Houda YELLAS is currently a Ph.D. student at Cnam, Paris. She received her Master degree in Network and Engineering from Paris Saclay University (UVSQ) in 2019. Her Doctoral research concerns mainly the automation and optimization of 5G mobile networks orchestration in Multi-access Edge Computing (MEC) infrastructures using data analytics.

Selma Boumerdassi received the M.Sc. degree in computer engineering from ESI, Algeria, and the M.Sc. degree in computer science and the Ph.D. degree in computer science from UVSQ, France. She was an Assistant Professor with UVSQ. Her research interests include wireless and mobile networks, with a special focus on the impact and use of social networks, trajectory prediction, information dissemination, and lightweight security. She is currently an Associate Professor with CNAM and a Research Associate with INRIA, Paris.

Alberto Ceselli is currently an Associate Professor in computer science with the Department of Computer Science, University of Milan. His research interests include prescriptive data analytics, mathematical programming, computational integer programming, and design and experimental analysis of algorithms for combinatorial optimization problems.

Bilal Maaz received his M.SC degree in computer and telecommunication engineering from UL, Lebanon, and the M.SC. degree in networking and telecommunication from Ecole Centrale Paris and the Ph.D in Networking, Information and communication from UVSQ. Currently he is an Associate Professor at the Cnam Lebanon, and in addition to teaching, he worked for years as a network engineer, system and QoS management. His current research concerns resource allocation and scheduling in wireless networks.

Stefano Secci is professor of networking at Cnam, Paris, France. He received his M.Sc. degree in communications engineering from the Politecnico di Milano, Italy, in 2005, and a dual Ph.D. degree in computer science and networks from the Politecnico di Milano and Télécom Paris-Tech, France. He held postdoctoral positions at NTNU, Norway, and GMU, United States. He was an associate professor with Sorbonne University-UPMC, Paris, France, from 2010 to 2018. He has also covered positions at NTNU, George Mason University, Fastweb Italia, and Ecole Polytechnique de Montréal. His current research interests are in network automation and control protocols.

APPENDIX A
SPATIAL CLUSTERING MODELS MATHEMATICAL FORMULATIONS

MIN-MAX:

$$\begin{aligned}
&\min \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \\
c_{ij} &= \max_{t\in T}|d_i^t - d_j^t| && \forall i,j \in \bar{A} \\
\sum_{j\in\bar{A}} z_{ij} + z_{ji} &= 1 && \forall i \in \bar{A} \\
z_{ij} &= 0 && \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \\
z_{ij} &\in \{0,1\} && \forall i,j \in \bar{A}
\end{aligned}$$

MIN-SUM:

$$\begin{aligned}
&\min \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \\
c_{ij} &= \frac{\sum_{t\in T}|d_i^t - d_j^t|}{\mathbf{card}(T)} && \forall i,j \in \bar{A} \\
\sum_{j\in\bar{A}} z_{ij} + z_{ji} &= 1 && \forall i \in \bar{A} \\
z_{ij} &= 0 && \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \\
z_{ij} &\in \{0,1\} && \forall i,j \in \bar{A}
\end{aligned}$$

MAX-MAX:

$$\begin{aligned}
&\max \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \\
c_{ij} &= \max_{t\in T}|d_i^t - d_j^t| && \forall i,j \in \bar{A} \\
\sum_{j\in\bar{A}} z_{ij} + z_{ji} &= 1 && \forall i \in \bar{A} \\
z_{ij} &= 0 && \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \\
z_{ij} &\in \{0,1\} && \forall i,j \in \bar{A}
\end{aligned}$$

MAX-SUM:

$$\begin{aligned}
&\max \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \\
c_{ij} &= \frac{\sum_{t\in T}|d_i^t - d_j^t|}{\mathbf{card}(T)} && \forall i,j \in \bar{A} \\
\sum_{j\in\bar{A}} z_{ij} + z_{ji} &= 1 && \forall i \in \bar{A} \\
z_{ij} &= 0 && \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \\
z_{ij} &\in \{0,1\} && \forall i,j \in \bar{A}
\end{aligned}$$

MIN-CORR:

$$\begin{aligned}
&\min \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \\
c_{ij} &= \max_{t\in T} \mathrm{ratio}_{i\,j} && \forall i,j \in \bar{A} \\
\mathrm{ratio}_{i\,j} &= \frac{R_{ij} * R_{ji}}{\sqrt{\sigma_i} * \sqrt{\sigma_j}} && \forall i,j \in \bar{A} \\
R_{ij} &= \sum_{t\in T} \frac{d_i^t - d_j^t}{2}, R_{ji} = \sum_{t\in T} \frac{d_j^t - d_i^t}{2} && \forall i,j \in \bar{A} \\
\sigma_i &= \sqrt{\sum_{t\in T}(d_i^t - \bar{d}_i)^2}, \sigma_j = \sqrt{\sum_{t\in T}(d_j^t - \bar{d}_j)^2} && \forall i,j \in \bar{A} \\
\sum_{j\in\bar{A}} z_{ij} + z_{ji} &= 1 && \forall i \in \bar{A} \\
z_{ij} &= 0 && \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \\
z_{ij} &\in \{0,1\} && \forall i,j \in \bar{A}
\end{aligned}$$

MAX-CORR:

$$\begin{aligned}
&\max \sum_{i\in\bar{A},j\in\bar{A}} c_{ij} * z_{ij} \\
c_{ij} &= \max_{t\in T} \mathrm{ratio}_{i\,j} && \forall i,j \in \bar{A} \\
\mathrm{ratio}_{i\,j} &= \frac{R_{ij} * R_{ji}}{\sqrt{\sigma_i} * \sqrt{\sigma_j}} && \forall i,j \in \bar{A} \\
R_{ij} &= \sum_{t\in T} \frac{d_i^t - d_j^t}{2}, R_{ji} = \sum_{t\in T} \frac{d_j^t d_i^t}{2} && \forall i,j \in \bar{A} \\
\sigma_i &= \sqrt{\sum_{t\in T}(d_i^t - \bar{d}_i)^2}, \sigma_j = \sqrt{\sum_{t\in T}(d_j^t - \bar{d}_j)^2} && \forall i,j \in \bar{A} \\
\sum_{j\in\bar{A}} z_{ij} + z_{ji} &= 1 && \forall i \in \bar{A} \\
z_{ij} &= 0 && \forall i,j \in \bar{A} : \delta_{ij} > \tilde{\delta} \\
z_{ij} &\in \{0,1\} && \forall i,j \in \bar{A}
\end{aligned}$$

MIN-CORR-VAR:
same as MIN-CORR, where we use $d_i'^t$ instead of d_i^t.
MAX-CORR-VAR:
same as MAX-CORR, where we use $d_i'^t$ instead of d_i^t.