A Dynamic Clustering Algorithm for Multi-Point Transmissions in Mission-Critical Communications

Alaa Daher[1,2], Marceau Coupechoux[1], Philippe Godlewski[1], Pierre Ngouat[3] and Pierre Minot[2]

[1]LTCI, Telecom Paris, Institut Polytechnique de Paris, France
{alaa.daher, marceau.coupechoux, philippe.godlewski}@telecom-paris.fr

[2]ETELM, Les Ulis, France; {alaa.daher,pierre.minot}@etelm.fr

[3]PNG-Technologies, Torcy, France; pierre.ngouat@png-technologies.com

***Abstract*—Reliable group video call is one of the main services offered by future Mission-Critical Communications (MCC). To support its requirements, coordinated multi-point transmission in multi-cell environments is an attractive feature for MCC over Multimedia Broadcast Multicast Services owing to its potential for coverage improvement and multicast transmission. In such a scheme, full cooperation among all cells of an area achieves the highest cooperative gain, but has stringent impact on system capacity. A trade-off in the cluster's size of serving cells thus arises between high Signal to Interference plus Noise Ratio (SINR) and network capacity. In this paper, we formulate an optimization problem to maintain an acceptable system blocking probability, while maximizing the average SINR of the multicast group users. For every multicast group to be served, a dynamic cluster of cells is selected based on the minimization of a submodular function that takes into account the traffic in every cell through some weights and the average SINR achieved by the group users. Traffic weights are then optimized using a modified Nelder-Mead simplex method with the objective of tracking a blocking probability threshold. The proposed clustering scheme is compared to full cooperation and to Single-Cell Point-To-Multipoint (SC-PTM) schemes. Results show that dynamic clustering offers the best trade-off between coverage and capacity for MCC.**

***Index Terms*—Mission-critical communications, multi-point transmissions, MBMS, dynamic clustering, optimization.**

I. Introduction

Mission-Critical Communications (MCC) are all communications related to the safety and the security of the civil society[1]. They may involve public safety services, like police forces, firemen, rescue and ambulance services, or employees of companies managing critical infrastructures, like energy and transportation suppliers [1]. MCC are conveyed by Professional Mobile Radio (PMR) networks, owing to their specific requirements such as reliability, coverage and network availability [1]–[3]. One of the most important and indispensable services offered by mission-critical networks is the group communication, which is a one-to-many or a many-to-many communication [4]. A group communication occurs for example within a team of rescuers, in which the communication of one is simultaneously received by all other team members. Group communications provide an efficient management of the rescue teams, and allow sending commands and sharing information with all contributors in a disaster area. These groups can be predefined or formed on-demand, may have geographic areas to cover, and may be organized based on types of skills or activities to be performed. Group communication is based on Multimedia Broadcast/Multicast Service (MBMS) in 3rd Generation Partnership Project (3GPP) standards. Due to its ability to mitigate Inter-Cell Interference (ICI) and improving cell-edge coverage, Coordinated MultiPoint (CoMP) transmission schemes appear as natural solutions for MCC for improving their reliability. In this paper, we propose a dynamic CoMP clustering algorithm, which addresses two unique features of MCC, which are the high reliability requirement and the possibility to perform group communications.

When several Base Stations (BSs) cooperate to serve a group of users, CoMP transmission helps mitigating ICI, improving system throughput and cell edge performance. The New Radio 5th Generation (5G) standard facilitates this functionality by allowing a split of the radio protocol stack of the next generation Node-B (gNB) between a central unit able to coordinate several distributed units dedicated to lower layers [5]. If all BSs of an area cooperate, we have a Multicast/Broadcast Single Frequency Network (MBSFN) transmission. All the BSs in an MBSFN area are synchronized and transmit the same signal over the same radio resources to all users of a group, so that the Signal to Interference plus Noise Ratio (SINR) of the users is maximized. However, radio resources can be wasted due to the transmission of data in cells where there is no group user. Thus, MCC benefits in this case of the best reliability in terms of coverage, but some group calls may be blocked due to the lack of resource. Another approach consists in transmitting the signal only in cells that cover the group users. If several group users are located in a given cell, the Base Station (BS) uses multicast transmission to serve them. Several BSs serving the same group may however interfere as resources are chosen independently. This technique is called Single-Cell Point-To-Multipoint (SC-PTM) by 3GPP [6] and has the advantage of maximizing the capacity of the network at the price of a degraded SINR for the group users, especially those at cell edge owing to the higher ICI compared to MBSFN. More MCC group calls can be served at the price of a degraded coverage reliability [7].

In general, there is a coverage-capacity trade-off between the full MBSFN cooperation and SC-PTM and it may be advantageous to consider *clusters* of cells cooperating using

[1] The work of M. Coupechoux has been performed at LINCS laboratory (www.lincs.fr).

CoMP. In this paper, we formulate a problem to capture this coverage-capacity trade-off in mission-critical networks using multi-point transmissions and we propose a dynamic clustering algorithm to solve this problem.

A. Related Work

Dynamic clustering in cooperative transmissions has been well investigated in the literature, but mostly for unicast best effort traffic with the goal of maximizing functions of the user data rates under static traffic models. The different approaches can be classified into network-centric and user-centric clustering. In the former, a set of BSs forms a cluster, and all the users attached to them are served by the cluster. In the latter, each user may have its own cluster of coordinated BSs and thus clusters for different transmissions may overlap.

Greedy algorithms based approaches have been widely used in network-centric clustering algorithms, e.g. [8]–[12]. They are based on the idea of sequentially adding the best BSs to the cluster up to a certain maximum cluster size. A greedy algorithm based on sum-rate maximization is developed in [8], one of the first reference on BS clustering. Yoon et al. focus in [9] on cell-edge users performance and propose a greedy clustering algorithm to maximize the cooperation gain, along with an interference weight calculation algorithm to reduce complexity. Improving the spectral efficiency is the main goal of [10], which relies on dynamic but non-overlapping clusters formed based on a greedy approach. In [11], a dynamic BS clustering algorithm has been proposed based on maximizing the weighted sum rate, using a greedy iterative algorithm. This clustering is combined with a scheduling algorithm and evaluated with dynamic traffic models in [12]. However, greedy search method based clustering algorithms may not provide the optimal cluster for group communications, since the cluster must be designed based on all group users performance. We note that, for computational complexity reasons, only non-overlapping clusters are considered in these studies, which may be sub-optimal for some users.

In an another set of papers, dynamic user-centric clustering schemes have been investigated. In [13], the authors designed a user-centric clustering scheme which aimed at maximizing the average throughput of the network, subject to the limitations on the synchronization and backhaul capacity constraints, as well as limited number of served users per cell. However, this scheme does not address dynamic group communications and focuses on unicast communication average rate maximization. Garcia et al. presented in [14] a criterion to balance the cost of resource utilization and the radio quality induced by BS cooperation; this criterion penalizes in a simple way higher cluster sizes without being related to traffic characteristics. In [15], a user-centric joint clustering and scheduling for CoMP in heterogeneous networks has been modeled using game theoretic approaches, combined with graph-coloring algorithms. To account for the large number of cells in a dense network, the authors proposed a scalable algorithm and assumed a maximum cluster size. Bassoy et al. evaluated in [16] the trade-off in CoMP clustering in the aim of improving heterogeneous network load balancing. The proposed algorithm employs a user-centric clustering approach assuming a limited cluster size to maximize CoMP gains in a first stage; then, a new clustering algorithm is used to distribute traffic gradually from macro loaded cells into relatively less-loaded small cells. Such scheme increases drastically model complexity when applied to MBMS traffic. In [17], the authors study the trade-off between cooperative gain and cost in a CoMP system; a user-centric cluster size minimization problem subject to rate constraints is formulated, and a sub-gradient method is employed to solve a relaxed version of the problem, using a Lagrangian approach. This paper considers a fixed number of users and thus does not take into account traffic dynamics.

Although user-centric approaches proposed in the literature offer more flexibility compared to network-centric schemes, they are not adapted to group calls in MCC. The first reason is that existing clustering algorithms mainly focus on the physical data rate maximization, while MCC require improved reliability and coverage. The second reason is that dynamic traffic constraints are ignored, e.g. [8]–[14]. Besides, many proposed models, such that [8]–[11], [15], [16], are limited by a fixed or maximal cluster size, which may not be optimal for some groups, especially in case group members are distributed in many different cells. At last, some papers minimize the cluster size, which contradicts the reliability requirement of MCC, see e.g. [17].

In the literature related to MBMS, Rong et al. presented in [18] a comparison between different MBMS transmission schemes; they evaluated a dynamic MBSFN scheme which only utilizes a subset of the cells for adaptively broadcasting the signal, depending on the presence of users in the cells. An algorithm to optimize the scheduling and resource allocation for unicast User Equipments (UEs) and MBMS in a MBSFN environment has been proposed in [19]; the authors show the trade-off between improving user data rates through unicast, and improving spectrum efficiency through multicast. However, the trade-off involved by clustering is not assessed in both studies. To the best of our knowledge, dynamic clustering of cooperative cells has thus not been studied in the literature in the context of mission-critical communications with the objective of dynamically finding the right trade-off between coverage and capacity for group communications.

B. Contributions

In this paper, our main contribution is the design of a clustering algorithm for multi-point transmissions that is able to maximize group average SINR under the constraint of maximum target cell blocking probabilities for group communications in MCC. Our detailed contributions are the following:

- We formulate a problem to capture the coverage-capacity trade-off in mission-critical communications, where group calls are served by multiple cells using coordinated multi-point transmissions.
- We derive a Kaufman-Roberts-like formula for computing the blocking probability of group calls, in every cell of a cooperative multi-point transmission scheme.
- To solve the formulated problem, we propose a Dynamic Clustering Algorithm (`DCA`) that is decomposed into

an inner loop called Group Call Clustering Algorithm (GCCA) and an outer loop called Cell Weights Optimization Algorithm (CWOA).

- For every group, we formulate a submodular minimization problem, in which the average SINR in the group and weights related to every selected cell are taken into account and that results in an optimal cluster of serving BSs. The traffic weight of a cell represents a cost for using the radio resources in this cell. This problem is solved using GCCA inspired by the minimum-norm algorithm. We evaluate its performance compared to a Greedy based approach, and we show that it provides better results.
- As group calls arrive in the system, are served and leave the system in a dynamic fashion, every cell experiences a blocking probability that depends on its traffic weight. We formulate a minimization problem, in which traffic weights are optimized in order to maintain blocking probabilities close to a target value. As the objective function involves the resolution of a Markov process, we propose CWOA, which relies on derivative-free optimization and the Nelder-Mead simplex method to solve this problem.
- We provide simulation results showing how the proposed framework can dynamically adapt clusters to traffic conditions in order to maximize the cell coverage under the constraint of a target blocking probability. As the traffic intensity increases, our scheme smoothly moves from a full cooperation scheme as in MBSFN to single cell transmissions as in SC-PTM. We also show that for mission-critical communications, the minimum-norm clustering algorithm outperforms the greedy clustering, often adopted by the literature.

The rest of the paper is organized as follows: in Section II, we present our system model and we introduce the problem. Our dynamic clustering algorithm is presented in Section III. Section IV presents and discusses the simulation results and conclusions are summarized in Section V.

Notations: For a finite set $V = \{1, ..., n\}$ of n elements, we denote 2^V the set of all its subsets. Every subset S of V can be represented by a vector in $\{0, 1\}^n$ denoted $\mathbf{1}_S$, which has 1 at the positions of the elements of S and 0 elsewhere. Take now $x \in \mathbb{R}^n$, a vector of n real components. We denote x_i the component of x corresponding to the element $i \in V$. We denote also $x(S) = \sum_{i \in S} x_i = \mathbf{1}_S^T x$. A set function F is a function from 2^V to $\mathbb{R}$. Equivalently, we can write: $F : \{0, 1\}^V \mapsto \mathbb{R}$. We denote $\{x \geq \alpha\}$ the set $\{i \in V | x_i \geq \alpha\}$. For a collection of points in $\mathbb{R}^n$ $S = \{q_1, ..., q_m\}$, we denote $aff(S) = \{y \in \mathbb{R}^n | y = \sum_{i=1}^m \alpha_i q_i, \alpha_i \in \mathbb{R}\}$ the affine hull of S and $conv(S) = \{y \in \mathbb{R}^n | y = \sum_{i=1}^m \alpha_i q_i, \alpha_i \in \mathbb{R}_+\}$ the convex hull of S. Standard asymptotic notations are used: $O(n)$ for an upper bound and $\Theta(n)$ for a tight bound.

II. Model and Problem Formulation

In this section, we present our system model, and we introduce the clustering problem.

A. System Model

We consider the downlink of a cellular network with omni-directional BSs serving groups of UEs[2]. We focus on a specific subset $V = \{1, ..., n\}$ of n cells forming an MBSFN synchronization area and thus able to perform multi-point transmissions to serve group of users located in this area. When a subset $S \subseteq V$ of BSs serves a group $\mathcal{U}$ of users using multi-point transmissions, a time-synchronized common waveform is transmitted simultaneously from S using the same resources, to convey the common service data requested by the group of users. User Equipment (UE) receives copies of the signal with different delays, amplitudes and phases and treat the multi-cell transmissions in the same way as multipath components of a single-cell transmission without incurring any additional complexity. It can thus benefit from spatial diversity, increased useful signal power and reduced inter-cell interference.

The signal received from a BS $b \in S$ is part of the useful received signal, provided that the propagation delay does not exceed the cyclic prefix duration [20]. Hence, the SINR experienced by the UE u can be expressed as:

$$\gamma_u(S) = \frac{\sum_{b \in S} \xi_{ub} g_{ub} P_T}{\sum_{b \in S}(1 - \xi_{ub}) g_{ub} P_T + \sum_{b \notin S} P_T g_{ub} + N_{th}}, \quad (1)$$

where ξ_{ub} denotes the useful portion of the signal received by u from b (see [20] for the detailed calculation as a function of the distance between u and b and the length of the cyclic prefix); P_T is the BS transmit power; g_{ub} is the channel gain between u and b and N_{th} is the thermal noise power. More specifically, $g_{ub} = \kappa d_{ub}^{-\eta} 10^{X_{ub}/10}$, where κ and $\eta > 2$ are constants, d_{ub} is the distance between u and b and X_{ub} is a zero-mean Gaussian random variable with standard deviation σ in dB, which models the shadowing effect. We call *best server* of a UE, the cell providing the maximum received power to this user, i.e., $\text{argmax}_b \, g_{ub} P_T$. Channel state information may be required at a central unit for an effective CoMP, which may represent an overhead because of pilot transmissions. However, in a TDD system, where channel reciprocity applies, the overhead is independent on the number of BSs and downlink channel estimation is based on uplink pilots [21]. There is thus no extra overhead on the downlink. For a multicast group $\mathcal{U}$ of N users served by cells in S, we define the average SINR of the group as:

$$\bar{\gamma}_{\mathcal{U}}(S) = \frac{1}{N} \sum_{u \in \mathcal{U}} \gamma_u(S). \quad (2)$$

If $S = V$, we say that we have a full MBSFN transmission or full cooperation of the synchronization area, this is the best choice of S for a group in terms of average SINR. If S is made of the set of best servers without cooperation, we have a SC-PTM transmission. Figure 1 shows an example of such a network, where BS locations have been drawn according to a Poisson process. The set V is shown in white, while gray cells are outside the MBSFN synchronization area. A group of users is shown that is served by a subset of the cells in this area.

[2] In this paper, we indifferently speak about *cell* and *BS*, and about *users* and *UEs*.

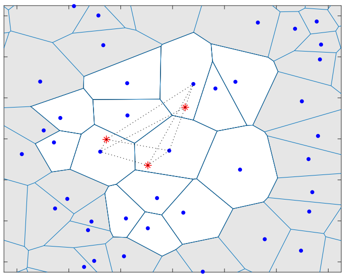


Fig. 1: The MBSFN synchronization area is made of the white cells (set V). Red stars represent UEs of a group. This group is served by a cluster of cells (set S) of the synchronization area.

B. Traffic Model and Preliminary Results

We consider a dynamic traffic model, in which group of users arrive in the synchronization area, use a resource for a group communication for a certain duration and leave the system. The whole process from the start to the end of the communication is called a *group call*. Group calls are usually video calls in mission-critical communications and for reliability reasons, MBSFN adopts a single robust Modulation and Coding Scheme (MCS) for transmission [22]. We can thus adopt a call blocking system, in which every call requires a single elementary resource. We thus assume random Poisson arrivals with rate λ [s^{-1}] and random exponential service duration with rate μ [s^{-1}]. There are R resources available in every BS. Note that CoMP transmissions imply an extra traffic in the backhaul between BSs or in the fronthaul in a Cloud Radio Access Network (CRAN). In case of backhaul capacity limitation, R should be interpreted as the *minimum* between the number of available resources in the radio interface and in the backhaul/fronthaul. Let $\rho = \lambda/\mu$ be the traffic load. This model has no restriction in terms of group size or geographical distribution of group members. In particular, individual calls can be seen as calls for groups of one or two members, whether the other group member is outside the MBSFN synchronization area or not.

When a group arrives in the system, it is served by a subset S of BSs with probability p_S and one resource is consumed in every BS in S. The way S is chosen will be detailed later on. To fix ideas, S can be the set of best servers, i.e., the BSs providing the highest power to every user in the group; or we can choose $S = V$ if we want to maximize the average SINR of the group. In general, S is chosen according to some policy balancing the radio quality and the usage of resources. The cardinality of 2^V is $P = 2^n$. To every element of S of 2^V, we associate a unique index $s \in \{1, ..., P\}$. For $b \in \{1, ..., n\}$, let $\mathcal{P}_V(b) = \{s \in 2^V | b \in S\}$ be the elements of 2^V that contain b, i.e., the set of clusters that include BS b. The traffic model induces a continuous-time Markov chain with state space:

$$\begin{aligned} \mathcal{E} \quad = \quad & \{(n_1, ..., n_s, ..., n_P) \in \mathbb{N}^P, \text{s.t.} \\ & \sum_{s \in \mathcal{P}_V(b)} n_s \leq R, b \in \{1, ..., n\}, s \in \{1, ..., P\}\} \end{aligned} \quad (3)$$

where n_s is the number of group calls served by S. The constraint states that in any cell b, no more than R resources can be allocated. The transition rate between x and $y \in \mathcal{E}$ is given by:

$$q(x, y) = \begin{cases} p_s \lambda & \text{if } y = x + e_s \\ x_s \mu & \text{if } y = x - e_s \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where $e_s = (0, ..., 1, ..., 0)$ with 1 at the s-th coordinate. Define $q(x, x) = -\sum_{y \neq x} q(x, y)$. Let $Q = (q(x, y))_{x,y \in \mathcal{E}}$ be the infinitesimal generator of the Markov process and $\pi(x)$ the stationary probability of being in state x, we have $\pi Q = 0$. Let $\mathcal{B}_s = \{(n_1, ..., n_P) | \exists b \in s \text{ s.t. } \sum_{t \in \mathcal{P}_V(b)} n_t = R\}$: these are the states that are blocking for calls requiring the activation of the BSs in cluster s because at least one station in s has already R resources occupied.

Proposition 1. *The stationary probabilities of the Markov process $(\mathcal{E}, Q)$ are given by:*

$$\pi(x) = \pi(0) \frac{(p_1 \rho)^{n_1}}{n_1!} ... \frac{(p_P \rho)^{n_P}}{n_P!}, \quad \forall x \in \mathcal{E} \quad (5)$$

where $\pi(0)$ is obtained by normalization.

Proof. Setting $R = \infty$, the random process is equivalent to P independent birth-and-death processes. It is thus reversible and its stationary probabilities are the product of the stationary probabilities of the independent processes. When $R < \infty$, the process is a truncation of a reversible process. Hence the result. □

Corollary 1. *The blocking probability for group calls served by the BSs in s is:*

$$\Pi(s) = \sum_{x \in \mathcal{B}_s} \pi(x). \quad (6)$$

The overall blocking probability is:

$$\Pi = \sum_{s=1}^{P} p_s \Pi(s). \quad (7)$$

We now focus on every individual cell $b \in V$ and compute the probability that this cell has R resources occupied. For $i = 1, ..., n$, let $v_i = [v_{i1} ... v_{iP}]^T$ be the vector such that $v_{is} = 1$ if $i \in s$ and 0 otherwise. Let $\zeta(m_1, ..., m_n)$, $0 \leq m_i \leq R$, $\forall i$, be the function defined as follows:

$$\zeta(m_1, ..., m_n) = \sum_{x \in \mathcal{E} | \forall i,\, x^T v_i = m_i} \frac{(p_1 \rho)^{n_1}}{n_1!} ... \frac{(p_P \rho)^{n_P}}{n_P!}. \quad (8)$$

This is a quantity that is proportional to the joint probability that cell 1 serves m_1 calls, cell 2 serves m_2 call,..., cell n serves m_n calls.

Lemma 1. *We have for any $m_i \neq 0$ in $\{m_1, ..., m_n\}$:*

$$\zeta(m_1, ..., m_n) = \frac{1}{m_i} \sum_{j=1}^{P} p_j \rho v_{ij} \zeta(m_1 - v_{1j}, ..., m_n - v_{nj}) \quad (9)$$

with $\zeta(0, ..., 0) = 1$ and $\zeta(m_1, ..., m_n) = 0$ whenever $\exists i$ s.t. $m_i < 0$.

Proof. See Appendix A. □

Proposition 2. *For a BS b, the probability that b has R resources occupied is given by:*

$$\tilde{\Pi}(b) = \frac{\sum_{\forall k \neq b,\ m_k=0,\ldots,R} \zeta(m_1, \ldots, m_b = R, \ldots, m_n)}{\sum_{\forall k,\ m_k=0,\ldots,R} \zeta(m_1, \ldots, m_n)}, \qquad (10)$$

where at the numerator, R is the value taken by the variable m_b.

With a slight abuse of vocabulary, we call this probability, *the blocking probability in station b*. It is clear that the policy for choosing S plays an important role in terms of traffic: the higher the probability for a BS b to be chosen, the higher the traffic load in this cell and the higher the probability $\tilde{\Pi}(b)$ to have no resource available for a new call.

Despite its accuracy in modeling group call performance, the blocking probability of a BS b provided in Proposition 2 has an exponential complexity, which is detrimental in large networks. Therefore, we can approximate this probability by using the Erlang-B law as follows:

$$\tilde{\Pi}(b) \approx E_B(b, R) = \frac{\frac{\rho_b^R}{R!}}{\sum_{r=0}^{R} \frac{\rho_b^r}{r!}}, \qquad (11)$$

where $E_B(b, R)$ is the classical Erlang-B law and $\rho_b = \rho \sum_{s=1}^{P} p_s v_{bs}$ is the traffic load in cell b. Although this sum has still 2^n terms, ρ_b can be in practice measured by BS b so that the load is obtained at very low cost. Moreover, the blocking probability given in (11) can be evaluated in $O(R)$, provided that ρ_b is evaluated in $O(1)$, using the following recursive classical formula:

$$E_B(b, R) = \frac{\rho_b E_B(b, R-1)}{R + \rho_b E_B(b, R-1)} \qquad (12)$$

and $E_B(b, 0) = 1$. This equation is an approximation of (10) in the sense that it ignores the correlation between loads in different cells induced by group calls.

C. Problem Formulation

The clustering policy to select the set S for a given group is based on a minimization problem. Our aim is to strike a balance between the radio quality of the group, measured by its average SINR and the amount of resources used in the network to serve this group. For this, we define the weight ω_b of a BS b, a free parameter that represents the *cost* of using resources in station b in a multi-point transmission. Our policy consists in choosing for the group $\mathcal{U}$ a set S^* that solves the following set function minimization problem:

$$\min_{S \in \mathcal{P}_V} \Psi_{\mathcal{U}}(S) \triangleq \omega(S) - \bar{\gamma}_{\mathcal{U}}(S) \qquad (13)$$

where $\Psi_{\mathcal{U}} : \mathcal{P}_V \mapsto \mathbb{R}$ is a set function balancing the cost and the radio quality, S is the set of serving BSs, $\omega(S) = \sum_{b \in S} \omega_b$ is the sum of the weights of the stations in S, and $\bar{\gamma}_{\mathcal{U}}(S)$ is the mean SINR of UEs $u \in \mathcal{U}$ when the group is served by S. By convention, we set $\Psi_{\mathcal{U}}(\emptyset) = 0$.

The objective function is made of two terms. The first term captures a cost of using cells in S in terms of traffic. Highly loaded cells have indeed large weights and are thus less likely to be chosen. Lightly loaded cells have small weights and are more likely to be included in the cluster. The second term captures the fact that we want to increase the SINR of the group and thus the communication reliability. Note that we have not considered user data rate as an objective function as it is usually done in the literature because group calls have a fixed data rate and reliability is the main requirement in MCC.

Clusters are formed dynamically for every group call according to the above clustering policy. For a fixed vector of weights $\omega \in \mathbb{R}^n$, the traffic demand together with the clustering policy induces a blocking probability $\tilde{\Pi}(b; \omega)$ in every cell b that we make now explicitly dependent on the weights vector. If $\tilde{\Pi}(b; \omega)$ is high, it is natural to increase the weight ω_b because the quality of service in this cell is poor. On the contrary, if $\tilde{\Pi}(b; \omega)$ is very small, we can decrease its weight so that this cell is chosen with higher probability and allows to increase the SINR of the users. We thus define target blocking probabilities $\bar{\Pi}(b)$ that every cell should attain. We now formulate a second minimization problem:

$$\min_{\omega \in \mathbb{R}^n} G(\omega) \triangleq \sum_{b=1}^{n} \|\tilde{\Pi}(b; \omega) - \bar{\Pi}(b)\|^2 \qquad (14)$$

where $G : \mathbb{R}^n \mapsto \mathbb{R}^+$ is a real valued function representing the quadratic error with respect to the targets, $\tilde{\Pi}(b; \omega)$ is the blocking probability of cell b when the clustering policy is applied with weights ω and $\bar{\Pi}(b)$ is the target blocking probability for cell b. Therefore, our problem is resolved by finding ω^*, a minimizer of G:

$$G(\omega^*) = \min_{\omega \in \mathbb{R}^n} G(\omega). \qquad (15)$$

III. A Dynamic Clustering Algorithm

In this section, we show how the above problems are solved.

A. Dynamic Clustering Algorithm

The Dynamic Clustering Algorithm DCA is illustrated in Fig. 2 and its pseudo-code is given in Algorithm 1. It is executed in a central entity for the whole synchronization area and selects a set of serving cells for every new group arrival. The algorithm can be run either in the Central Unit (CU) of a CRAN or in a Software Defined Network controller in the core network. In both cases, the central entity controls the transmissions of all BSs. The algorithm proceeds by periods of duration T (T is sufficiently long in order that the Markov process of group call arrivals and departures has reached the stationary regime). During a period, the weights are fixed, the minimization problem (13) is solved at every group arrival by GCCA, and traffic statistics are gathered. At the end of every period, the minimization problem (14) is solved by CWOA and weights are updated.

To be more specific, the algorithm is called at every group arrival (steps 5-11) and every T seconds (steps 12-18). When a group call arrives at a certain time instant τ (step 5), the set of serving cells is determined (step 7) by solving the optimization problem (13) with GCCA (DCA inner loop). Time is updated (step 6) and we record the number of calls during the period with the variable M (step 6). The number of group calls served by the same set of cells S is incremented (step 8). The group

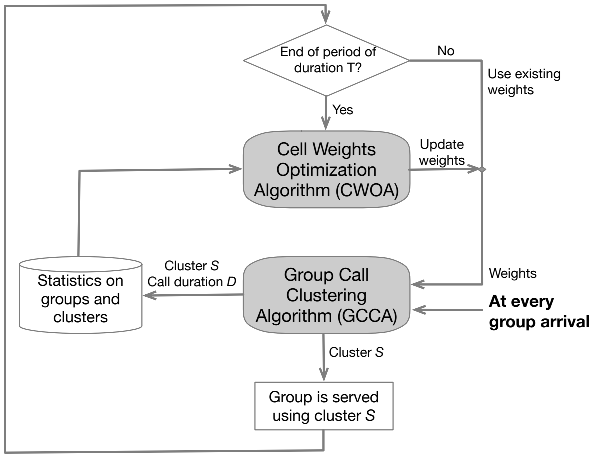

End of period of duration T?
No
Use existing weights
Yes
Cell Weights Optimization Algorithm (CWOA)
Update weights
Statistics on groups and clusters
Cluster S
Call duration D
Group Call Clustering Algorithm (GCCA)
Weights
At every group arrival
Cluster S
Group is served using cluster S

Fig. 2: Dynamic Clustering Algorithm (`DCA`). `GCCA` is called at every group arrival. `CWOA` is called every T.

of users is served with S (step 9). The duration of the call is recorded with variable D (step 10). At the end of a period (step 12), the probabilities of every subset are computed (step 14) and the load ρ is estimated (step 15). We see here that p_S and ρ are not inputs of our solution but are learned by the algorithm itself. New weights are computed by solving the optimization problem (14) (step 16) with `CWOA` (`DCA` outer loop). Variables M, D, and $n_{S'}$ are reinitialized before the next period starts (step 17). As we see, algorithms `GCCA` and `CWOA` are embedded in `DCA`. They are now described in Sections III-B and III-C, respectively.

Algorithm 1: `DCA` (executed in a central entity for the entire synchronization area)

```
1:  Input: $T$, $\bar{\Pi}(b) \forall b$.
2:  Output: For every group of users, a set $S$ of
    cooperative serving cells.
3:  Init: $t \leftarrow 0$; $t' \leftarrow 0$; $\omega \leftarrow 0$; $\forall S' \subseteq V,\ n_{S'} \leftarrow 0$;
    $M \leftarrow 0$; $D \leftarrow 0$
4:  while true do
5:    if a new group $\mathcal{U}$ arrives at $\tau$ then
6:      $t \leftarrow \tau$; $M \leftarrow M + 1$
7:      $S \leftarrow$ GCCA($\mathcal{U}$, $\omega$), see Algorithm 3
8:      $n_S \leftarrow n_S + 1$
9:      Serve group $\mathcal{U}$ with $S$.
10:     $D \leftarrow D+$ duration of this call
11:   end if
12:   if $|t - t'| \geq T$ then
13:     $t' \leftarrow t$
14:     $\forall S' \subseteq V,\ p_{S'} \leftarrow \frac{1}{M} n_{S'}$
15:     $\lambda \leftarrow \frac{M}{|t-t'|}$; $\mu \leftarrow \frac{M}{D}$; $\rho \leftarrow \frac{\lambda}{\mu}$
16:     $\omega \leftarrow$ CWOA($\rho$, $p_{S'} \forall S' \subseteq V$, $\bar{\Pi}(b) \forall b$), see
        Algorithm 5
17:     $M \leftarrow 0$; $D \leftarrow 0$; $\forall S' \subseteq V,\ n_{S'} \leftarrow 0$
18:   end if
19: end while
```

B. Group Call Clustering Algorithm

In this section, we focus on the first minimization problem (13). We first detail `GCCA` and then describe the Greedy Clustering algorithm for the sake of comparison.

1) `GCCA`: We show that, for a given group of users $\mathcal{U}$ and a given set of weights $\omega \in \mathbb{R}^n$, problem (13) is a submodular minimization problem. Then, we use the submodular properties and the minimum-norm algorithm to minimize the $\Psi_{\mathcal{U}}$ function.

Definition 1 (Submodular Function [23])**.** *A set function $F : 2^V \mapsto \mathbb{R}$ is submodular if and only if, for all subsets $A, B \subseteq V$ and $e \in V$ such that $A \subseteq B$ and $e \notin B$, we have: $F(A \cup e) - F(A) \geq F(B \cup e) - F(B)$.*

Moreover, a function F is *supermodular* if $-F$ is submodular and a function F that is both submodular and supermodular is called *modular*. A function F is modular if and only if there exists $\omega \in \mathbb{R}^n$ such that $F(A) = \sum_{e \in A} \omega_e$ for all $A \subseteq V$. Besides, the linear combination of submodular functions is submodular, and submodularity is preserved under taking non-negative linear combinations [23].

Proposition 3. *For a given group $\mathcal{U}$ and a given weight vector $\omega \in \mathbb{R}^n$, the objective function $\Psi_{\mathcal{U}}$ defined in* (13) *is submodular.*

Proof. See Appendix B. □

Two important sets play a role in submodular function minimization.

Definition 2 (Submodular and Base Polyhedron [23])**.** *Let F be a submodular function such that $F(\emptyset) = 0$. The submodular polyhedron $\mathcal{P}_F$ and the base polyhedron $\mathcal{B}_F$ are defined as:*

$$\begin{aligned} \mathcal{P}_F &= \{\omega \in \mathbb{R}^n : \omega(A) \leq F(A) \textit{ for all } A \subseteq V\} \\ \mathcal{B}_F &= \{\omega \in \mathcal{P}_F : \omega(V) = F(V)\} \end{aligned}$$

The connection between submodular function minimization problem and the base polytope is given in the following lemma deduced from the seminal work of Fujishige in [24].

Lemma 2 ([23])**.** *For a given $x \in \mathbb{R}^n$, Algorithm 2 solves the linear minimization problem $\min_{p \in \mathcal{B}_F} x^T p$ over $\mathcal{B}_F$. Moreover, all extreme points of $\mathcal{B}_F$ can be obtained using this algorithm using all possible ordering of x coordinates.*

Lemma 3 ([23], [24])**.** *Let F be a submodular function such that $F(\emptyset) = 0$. Let s^* be the point of $\mathcal{B}_F$ with minimum-norm, i.e., $s^* = \arg\min_{s \in \mathcal{B}_F} \frac{1}{2} ||s||_2^2$. The minimal minimizer of F is $\{j \in V | s_j^* < 0\}$.*

We thus see that minimizing a submodular function is equivalent to find the minimum-norm point of its base polyhedron. Wolfe has described in [25] an iterative procedure to find minimum norm points in polytopes. Although the base polytope has exponentially many constraints, a simple linear optimization method can minimize any linear function over it. Therefore, Fujishige has suggested in [26] to use Wolfes procedure on the base polytope coupled with Algorithm 2 as a natural approach to submodular function minimization.

Algorithm 2: Linear optimization over the base polyhedron.

```
Input: $x \in \mathbb{R}^n$
Output: $q = \arg\min_{p \in \mathcal{B}_F} x^T p$
Sort the coordinates of $x$ in increasing order :
    $$x_{j_1} \leq x_{j_2} \leq \ldots \leq x_{j_n},$$
    where $\{j_1, \ldots, j_n\}$ is a permutation of the elements of $V = \{1 \ldots n\}$.
Define for $q = \{q_{j_k}\}_{k=1,\ldots,n}$ such that :
    $$q_{j_k} = F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\}),$$
    with $F(\{j_1, \ldots, j_{k-1}\}) = F(\emptyset) = 0$ for $k = 1$.
return $q$
```

Algorithm 3 shows the procedure, inspired by [27], to solve (13). We provide in the following a brief explanation of the different steps of the algorithm.

Following [25], let's call a *corral*, a set of points whose affine minimizer is also in its convex hull. Along the algorithm, a set S is populated with vertices of $\mathcal{B}_F$. If this set of vertices is sufficient to find the minimum norm point of $\mathcal{B}_F$, the algorithm stops. Otherwise, a new vertex is added to S and some vertices are removed so that S becomes a corral. The algorithm thus proceeds by major cycles (steps 4-19) and every major cycles possibly involves several minor cycles (steps 8-17). At the beginning of a major cycle (step 5), we have a corral S together with its affine minimizer $x \in \text{conv}(S)$. If this x verifies the optimality condition, the algorithm stops (step 6). The instruction $S^* \leftarrow \{i | x_i < 0\}$ provides the set of BS indices that form the cluster. Note that the optimality check is a simple condition resulting from the convexity of $\mathcal{B}_F$. Otherwise, a new point from the vertices of $\mathcal{B}_F$ (step 5) is added to S (step 7) and minor cycles start. A vertex of $\mathcal{B}_F$ is obtained by means of Algorithm 2, as explained in Lemma 2. In the minor cycle, the new affine minimizer y of S is computed (step 9). This operation is a standard quadratic minimization problem with linear constraints, see Appendix C. If by chance $y \in \text{conv}(S)$ (step 10), then S is a corral and a new major cycle starts with S and $x \leftarrow y$. Otherwise (steps 12-16), x is moved in the direction of y (step 14) up to the boundary of $\text{conv}(S)$ (steps 13 and 14). As we are on the boundary of $\text{conv}(S)$, at least one point of S is not necessary any more to describe x. Such points are removed from S (step 15). When writing $x = \sum_i \lambda_i q_i$, only the q_i corresponding to $\lambda_i > 0$ are kept in S. Minor cycles continue until a corral is found (step 11). In this case, x and y can coincide (step 18) and a new major cycle starts with this corral and x. At the initialization of the algorithm, it is required to start with a vertex of $\mathcal{B}_F$. We proceed as follows. According to Theorem 2.1 of [28], every minimization problem of the form $\min_{x \in \mathcal{B}_F} p^T x$ provides a vertex of $\mathcal{B}_F$. As this minimization problem is solved using Algorithm 2, every permutation of $\{1, 2, \ldots, V\}$ generates a vertex of $\mathcal{B}_F$. So for the initialisation of the algorithm, we can choose the identity permutation. As a result, defining for $k = 1, \ldots, V$: $q_k = F(\{1, \ldots, k\}) - F(\{1, \ldots, k-1\})$ with $F(\{1, \ldots, k-1\}) = F(\emptyset) = 0$ when $k = 1$, provides a vertex of $\mathcal{B}_F$.

Algorithm 3: GCCA

```
1:  Input: A group $\mathcal{U}$. A weight vector $\omega \in \mathbb{R}^n$.
2:  Output: A set $S^*$ of serving cells solving (13).
3:  Init: $F \leftarrow \Psi_{\mathcal{U}}$. Let $q$ be a vertex of $\mathcal{B}_F$ (see Algorithm 2 and Lemma 2); $x \leftarrow q$; $S \leftarrow \{q\}$; let $\lambda_i$ s.t. $x = \sum_i \lambda_i q_i$; $\lambda_1 \leftarrow 1$
4:  while true (major cycle) do
5:      $q \leftarrow \arg\min_{p \in \mathcal{B}_F} x^T p$, (see Algorithm 2 and Lemma 2)
6:      if $||x||^2 = x^T q$ then Return $S^* \leftarrow \{i | x_i < 0\}$ end if
7:      $S \leftarrow S \cup \{q\}$
8:      while true (minor cycle) do
9:          $y \leftarrow \arg\min_{z \in \text{aff}(S)} ||z||$; let $\alpha_i$ s.t. $y = \sum_i \alpha_i q_i$
10:         if $\alpha_i \geq 0$ for all $i$ % $y$ is in conv($S$) then
11:             break.
12:         else
13:             $\theta \leftarrow \min_{i:\alpha_i < \lambda_i} \lambda_i / (\lambda_i - \alpha_i)$
14:             $x \leftarrow \theta y + (1 - \theta) x$; $\lambda \leftarrow \theta \alpha + (1 - \theta) \lambda$
15:             $S \leftarrow \{q_i : \lambda_i > 0\}$
16:         end if
17:     end while
18:     $x \leftarrow y$; $\lambda \leftarrow \alpha$
19: end while
```

Lemma 4. *Every iteration of Algorithm 3 has a $O(n^2)$ complexity.*

Proof. Algorithm 2 complexity is dominated by the sorting of the coordinates of x and is thus in $O(n \log n)$. The minimization over the affine hull in step 9 has a complexity in $O(n^2)$. All other steps are in $O(n)$. □

It is however not known how many iterations are required for the termination. Our simulation results show that very few iterations are in practice required in our case.

2) Greedy Clustering: The greedy algorithm has been widely used in the literature on clustering for multi-point transmissions, see e.g. [8]–[12]. For comparison with GCCA, we thus adapt the greedy approach to our context in order to find an approximate minimizer of the function $\Psi_{\mathcal{U}}$ for a given group of users $\mathcal{U}$ and a set of weights $\omega \in \mathbb{R}^n$. The Greedy Clustering, summarized in Algorithm 4, consists in initializing the cluster of BSs as the set of best server cells of group users (step 3). Then, at every iteration and if possible, we add to the cluster the BS that decreases at most the objective function (13) (steps 5-12). If there is no such BS (step 13) or if we have reached the whole set of cells (step 16), we return the current cluster (steps 17 or 14). Note that Algorithm 4 requires $n - 1$ iterations in the worst case and has thus a complexity in $O(n)$.

C. Cell Weights Optimization Algorithm

In this section, we consider the problem of finding a minimum of the real-valued function G (see (15)). The difficulty lies in the fact that a closed-form of G is not available

Algorithm 4: Greedy Clustering.

```
**Input:** A group $\mathcal{U}$. A weight vector $\omega \in \mathbb{R}^n$.
**Output:** A set $S$ of serving cells solving (13) and the value $F$ of the objective function.
**Init:** Let $S$ the set of best server BSs of $\mathcal{U}$ members; $\bar{S} \leftarrow V \backslash S$; $F \leftarrow \Psi_{\mathcal{U}}(S)$.
while $\bar{S} \neq \emptyset$ do
    for $j \in \bar{S}$ do
        $\Psi_{\mathcal{U},j} \leftarrow \Psi_{\mathcal{U}}(S \cup \{j\})$
    end for
    if $\Psi_{\mathcal{U}}(S) > \min_{j \in \bar{S}}(\Psi_{\mathcal{U},j})$ then
        $b \leftarrow \arg\min_{j \in \bar{S}}(\Psi_{\mathcal{U},j})$
        $S \leftarrow S \cup \{b\}$
        $\bar{S} \leftarrow \bar{S} \backslash \{b\}$
        $F \leftarrow \min_{j \in \bar{S}}(\Psi_{\mathcal{U},j})$
    else
        return $S$, $F$
    end if
end while
return $S$, $F$
```

as its evaluation requires to solve a Markov process. An interesting class of methods for solving derivative-free (or black box) functions are the direct search methods, which update iteratively an initial assumption of a solution using a few function evaluations along linearly independent directions. The most popular is the Nelder-Mead simplex method [29]. We thus propose CWOA, an adaptation of this method for the cell weight optimization, that includes a new random oriented restart procedure, see Algorithm 5[3]. The algorithm is an heuristic that operates on an n-dimensional simplex of $n+1$ vertices, and aims to improve the "worst" vertex (i.e., with highest objective function value), through certain transformations of the simplex. These transformations are done with respect to a centroid vector (step 20), which is nothing else than the average vector of the simplex. We thus start with $n+1$ random vectors ω (step 3) that are sorted according to the objective function (step 6). To speed up the convergence, it is preferable to randomly draw initial weights with the same order of magnitude as average group SINRs (this can be easily obtained by a pre-processing step). With period I (step 7, using mod, the modulo operator), we check if the standard deviation of the blocking probability around their target values is small and if it is the case, we return the best weight vector (step 8-10). In step 8, stddev provides the standard deviation of a set of values. Otherwise, we perform a random oriented restart, i.e., we randomly generate new weights for the cells, which are far from their target (steps 11-18). In step 11, rand$(1, n)$ outputs a vector of n values uniformly drawn between 0 and 1. In step 12, the notation $(\omega_i)_b$ indicates the b-th coordinate of vector ω_i.

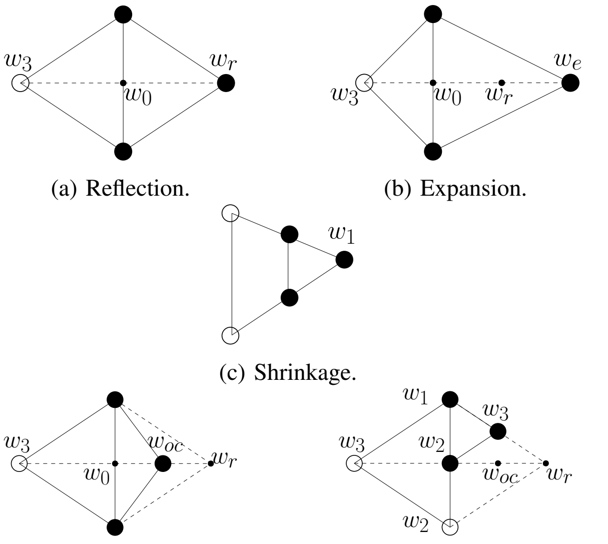
The possible transformations of the simplex are illustrated in Fig. 3. In a Reflection (steps 21-23), the worst vector (ω_{n+1}) is reflected with respect to the centroid and the reflected point replaces the worst vector if the objective function is decreased at this point. Expansion (steps 24-26) and contraction (steps 28-30, 35-37) are two other possible transformations. The simplex is shrunk (steps 32 and 39), towards the best vertex, whenever it is not improved by the preceding transformations, and a new iteration is then started. A restatement of the original Nelder-Mead (NM) algorithm [29] has been presented in [30], and is the basis of Algorithm 5.

Algorithm 5: CWOA

```
1:  **Input:** A set of parameters $-1 < \delta_{ic} < 0 < \delta_{oc} < \delta_r < \delta_e$ and $0 < \delta_s < 1$, $\epsilon_1 > 0$, $\epsilon_2 > 0$, $I \in \mathbb{N} \backslash \{0\}$, $\rho$, $p_{S'} \forall S' \subseteq V$, $\bar{\Pi}(b) \forall b \in V$.
2:  **Output:** A weight vector $\omega^* \in \mathbb{R}^n$ solving (14).
3:  **Init:** $f \leftarrow G$. Generate $n+1$ vertices $\omega \in \mathbb{R}^n$. $It \leftarrow 0$.
4:  while true do
5:      $It \leftarrow It + 1$
6:      Sort the vertices according to the objective function values:
        $f(\omega_1) \leq \ldots \leq f(\omega_{n+1})$
7:      if $f(\omega_{n+1}) - f(\omega_1) > \epsilon_1$ or mod$(It, I) = 0$ then
8:          if stddev$(\tilde{\Pi}(b;\omega_1), b \in V) < \epsilon_2$ then
9:              Return $\omega^* \leftarrow \omega_1$
10:         end if
11:         $b \leftarrow \arg\min_{j \in V} \tilde{\Pi}(j;\omega_1)$; $\theta \leftarrow$rand$(1, n+1)$
12:         $\forall i \in \{1, \ldots, n+1\}$, $(\omega_i)_b \leftarrow (\omega_i)_b \theta_i$
13:         $\theta \leftarrow$rand$(1, n+1)$
14:         for $b \in V$ do
15:             if $\tilde{\Pi}(b;\omega_1) > \bar{\Pi}(b)$ then
16:                 $\forall i \in \{1, \ldots, n+1\}$, $(\omega_i)_b \leftarrow (\omega_i)_b / \theta_i$
17:             end if
18:         end for
19:     end if
20:     % Compute the centroid $(\omega_0)$ of all vertices except $\omega_{n+1}$: $\omega_0 \leftarrow \frac{1}{n} \sum_{i=1}^{n} \omega_i$
21:     % **Reflection:** Compute reflected point: $\omega_r \leftarrow \omega_0 + \delta_r(\omega_0 - \omega_{n+1})$
22:     if $f(\omega_1) \leq f(\omega_r) < f(\omega_n)$ then
23:         $\omega_{n+1} \leftarrow \omega_r$
24:     else if $f(\omega_r) < f(\omega_1)$ then
25:         % **Expansion:** Compute the expanded point: $\omega_e \leftarrow \omega_0 + \delta_e(\omega_r - \omega_0)$
26:         if $f(\omega_e) < f(\omega_r)$ then $\omega_{n+1} \leftarrow \omega_e$ else $\omega_{n+1} \leftarrow \omega_r$ end if
27:     else if $f(\omega_n) \leq f(\omega_r) < f(\omega_{n+1})$ then
28:         % **Outside contraction:** Compute the contracted point:
            $\omega_{oc} \leftarrow \omega_0 + \delta_{oc}(\omega_r - \omega_0)$
29:         if $f(\omega_{oc}) \leq f(\omega_r)$ then
30:             $\omega_{n+1} \leftarrow \omega_{oc}$
31:         else
32:             **Shrink:** $\omega_i \leftarrow \omega_1 + \delta_s(\omega_i - \omega_1); \forall i \in \{2, \ldots, n+1\}$
33:         end if
34:     else
35:         % **Inside contraction:** Compute the contracted point:
            $\omega_{ic} \leftarrow \omega_0 - \delta_{ic}(\omega_{n+1} - \omega_0)$
36:         if $f(\omega_{ic}) \leq f(\omega_{n+1})$ then
37:             $\omega_{n+1} \leftarrow \omega_{ic}$
38:         else
39:             **Shrink:** $\omega_i \leftarrow \omega_1 + \delta_s(\omega_i - \omega_1); \forall i \in \{2, \ldots, n+1\}$
40:         end if
41:     end if
42: end while
```

However, despite its generally good performance, the NM algorithm can stagnate, fail to converge or converge to a non-optimal vertex, even for simple and convex objective functions. Then, numerous variants of this algorithm have been developed and analyzed, aiming to improve its performance and converge to a stationary vertex under mild conditions on the objective function. A possible improvement of the original algorithm is to impose restarts of the algorithm during the optimization run at certain iterations, by applying a fresh simplex. A restart is global when it is completely independent of previous results. It is local when a new simplex is initialized using the best known solution and a projection of the most recent simplex. However, such schemes, e.g. the globalized bounded NM [31], increase significantly the number of iterations of the algorithm, and don't provide good solutions to our problem, as observed in our simulations.

[3] In this algorithm, the symbol % is used for a commented line (not executed by the algorithm).



(a) Reflection.

(b) Expansion.

(c) Shrinkage.

(d) Contraction from reflected point.

(e) Shrinkage using reflected point.

Fig. 3: Simplex operations in $\mathbb{R}^2$ (white dots indicate the worst vertices before operations; solid dots indicate new simplex).

Therefore, we propose an oriented restart of the NM algorithm adapted to our model (steps 8-19). The idea is to reinitialize the simplex by modifying the weights of the cells having blocking probabilities far from their target. At every restart, the weight of cells having the lowest blocking probability are decreased, while the weight of cells having blocking probabilities higher than their targets are increased. A random perturbation is applied to every modification. The proposed restart scheme is applied periodically at the beginning of the NM algorithm major cycle. The period is controlled by the integer parameter I in Algorithm 5.

D. Complexity Analysis

The general complexity of the Nelder-Mead algorithm has not been assessed in the literature [32]. However, the complexity of a single iteration of Nelder-Mead algorithm has been assessed in [32] and depends on the complexity of evaluating the objective function. In our case, we have the following results for CWOA and then for the overall complexity of DCA.

Lemma 5. *Each iteration of CWOA has a complexity in $\Theta(n + R)$ (when shrink is not needed) or $\Theta(n^2 + R)$ (shrink is used) when the Erlang-B approximation (12) is adopted for the evaluation of G. Each iteration has an exponential complexity when (9) is adopted for the evaluation of G.*

Lemma 6. *Let λ be the number of group arrivals per second. Let N_1 and N_2 be the numbers of iterations of GCCA and CWOA, respectively at every algorithm call. Then, the overall complexity of DCA is in $O(\lambda T N_1 n^2 + N_2 n^2 + N_2 R)$ for every period of duration T.*

IV. Simulation Results

In this section, we show that GCCA and CWOA converge in few iterations in a realistic scenario. We have indeed no theoretical guaranty on these numbers. We show that DCA meets the blocking probabilities targets and is adaptive with respect to the traffic load. At last, we compare our solution to existing schemes, i.e., Greedy, globalized bounded Nelder-Mead, SC-PTM and full MBSFN cooperation.

A. Simulations Settings

We evaluate the performance of the proposed scheme in a MBSFN synchronization area of 14 BSs. All cells outside this area are interfering (see Fig. 1). For any computed cluster S and any randomly generated group $\mathcal{U}$, the SINR of every group member and the average SINR of the group are numerically evaluated using (1) and (2). Note that weights are dynamically adjusted by CWOA along the simulation. In the simulations, for simplicity reasons, we ignore the correlation between blocking probabilities in different cells and adopt the Erlang-B approximation. In order to assess the traffic dynamics in our evaluations, we assume the following users and group distributions[4]: 25% of arriving UEs are served by BS1 (an overloaded cell), 3% of them are dropped in BS7 (an underloaded cell), while other UEs are uniformly distributed in the other cells of the synchronization area; 50% of arriving groups are centralized, i.e., all the users are distributed in the closest area (of radius 1.5 km) of a group leader UE. In the remaining groups, the UEs are randomly distributed in the target network. These assumptions induce heterogeneous cell loads and heterogeneous group users distributions. The other system simulation parameters are given in Tab. I. They are typical for mission-critical communications.

Parameter	Value
Carrier frequency	700 MHz
Channel Bandwidth (W)	5 MHz
Groups arrival rate (λ)	1/55 s^{-1}
Mean service duration ($1/\mu$)	180 s
Available resources per BS (R)	5 resources
Number of users per group (N)	10 UEs/group
Clustering algorithm period (T)	30 minutes
Target blocking probability $\bar{\Pi}(b)$	2%
NM parameters $\{\delta_r, \delta_e, \delta_{oc}, \delta_{ic}, \delta_s\}$	$\{1, 2, 1/2, -1/2, 1/2\}$

TABLE I: Simulation parameters.

B. Group Call Multi-point Transmission

We focus here on the results provided by GCCA, Algorithm 3. Fig. 4 shows the evolution of function $\Psi_{\mathcal{U}}$ along algorithm's iterations for the proposed scheme, evaluated with the weight vector provided by CWOA at the end of the simulation. The values of the objective function for SC-PTM and full MBSFN cooperation are provided for comparison. Results show that these schemes are outperformed by GCCA and that the submodular minimization problem (13) is solved in very few iterations. In Fig. 5, we compare for a specific group the Greedy Clustering with GCCA and show the objective function along the iterations of the two algorithms. GCCA stabilizes in very few iterations and outperforms Greedy. Unfortunately, we

[4] These figures are taken from a typical use case for MCC, experienced by ETELM.

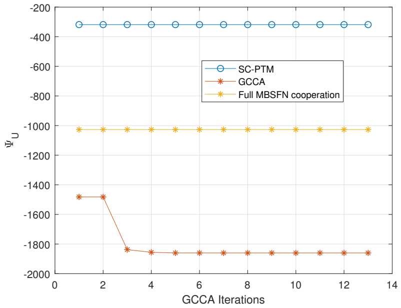


Fig. 4: Evolution of $\Psi_{\mathcal{U}}$ along the iterations of GCCA.

don't have any guaranty on the number of iterations. In the seminal paper of Fujishige et al. [33], it is shown by numerical experiments that the number of iterations increases with the number of vertices. It is thus expected that the number of iterations would increase for larger MBSFN synchronization areas. In the same paper, we note that very few iterations are required to reduce drastically the duality gap, which is in line with our experiments. We observe that greedy, a widely used algorithm for multi-point transmission clustering, may not provide the optimal solution.

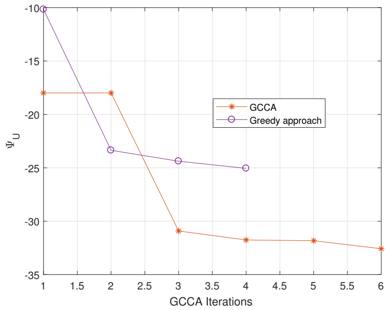
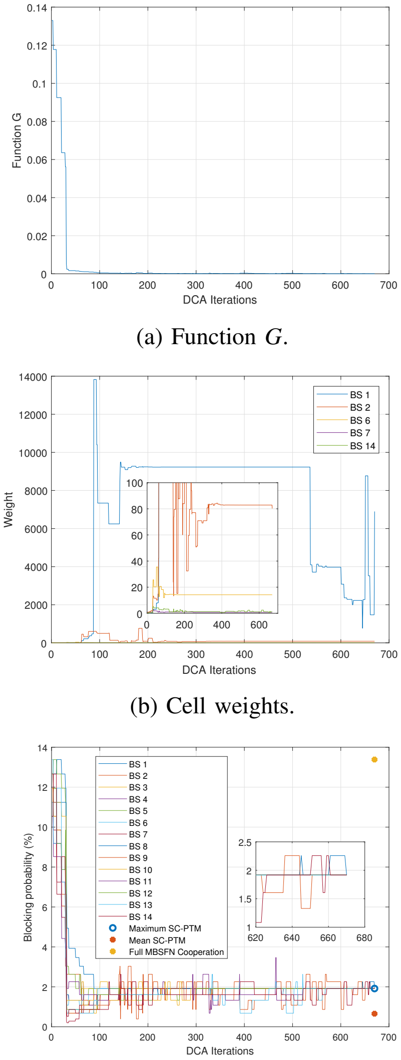


Fig. 5: Comparison of GCCA with Greedy Clustering.



(a) Function G.

(b) Cell weights.

(c) Cells blocking probabilities.

Fig. 6: Evolution of the objective function G, cell weights and blocking probabilities along the iterations of DCA.

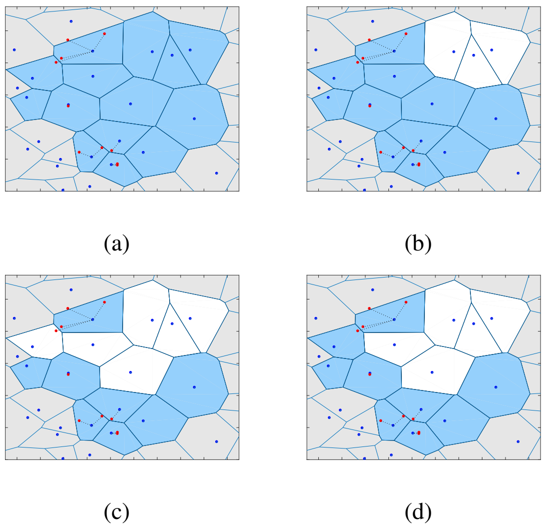
C. Objective Function and Blocking Probabilities

Fig. 6 shows the evolution of the objective function G (a), of cell weights (b) and of blocking probabilities (c) for each cell along the iterations of DCA. If the blocking probabilities are above the targets, this means that the network is unable to sustain the traffic load, i.e. that capacity is not sufficient. In Fig. 6a, we see that the function G decreases until it converges in few tens of iterations to a value close to 0 when blocking probabilities of all cells are close to corresponding targets. Along the algorithm, the cell weights and blocking probabilities fluctuate, especially at the iterations corresponding to random oriented restarts, until the probabilities reach their fixed targets (2%), see Fig. 6b. Note that the weight of BS1 is very high compared to other weights because this cell is overloaded. In Fig. 6c, the first iterations of DCA reveals that the blocking probabilities are very high, close to the ones achieved with full MBSFN cooperation (13.4%). This means that cluster sizes are large with initial weights. Gradually, the cluster sizes decrease, so that blocking probabilities reach their target value. This means that DCA meets the blocking constraint. With SC-PTM, the blocking probability is 3% and 0.01% in the overloaded and underloaded cells respectively, and falls below 0.8% in all other cells. Thus, SC-PTM provides the best performance in terms of blocking, however at the cost of lower coverage as we will see. Fig. 7 shows for a given group the cluster evolution provided by DCA along some iterations.

In Fig. 8, we test the globalized bounded NM [31] in place of CWOA (implementing random oriented restart) on the same simulated scenario and show the evolution of G with this scheme. We see that for our problem, the function G and the cell weights are still fluctuating after 2000 iterations with the globalized bounded NM, whereas with our random oriented restart, few tens of iterations were sufficient to obtain convergence (see Fig. 6a).

D. SINR Improvements

In order to compare the performance of different transmission schemes in terms of coverage, we show in Fig. 9

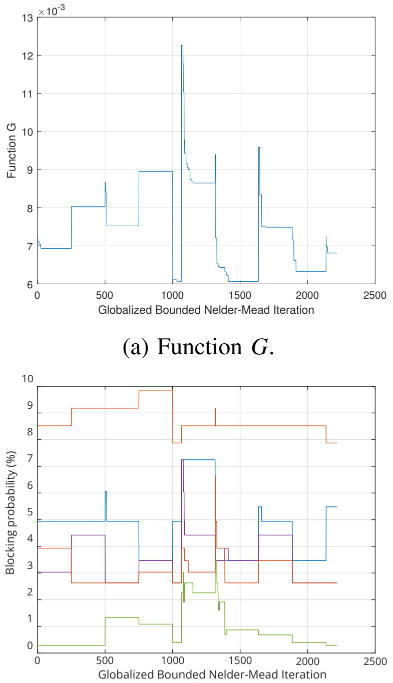

(a)

(b)

(c)

(d)

Fig. 7: Evolution of the cluster of a given group along some iterations of DCA. The cluster serving the group is formed by the blue cells, while the white cells are interfering. The links show the best server of each UE.



(a) Function G.

(b) Cells blocking probabilities.

Fig. 8: Evolution of the objective function G and blocking probabilities for some BSs along the iterations of globalized bounded NM algorithm.

the distribution of the mean group SINR with SC-PTM, full MBSFN cooperation and DCA. Coverage is indeed closely related to the SINR distribution since it can be defined as the probability that for a group the SINR is above a certain threshold. The SINR depends of course on clusters size. Thus, full MBSFN cooperation scheme provides the best SINR results, since all BSs inside the synchronization area contribute to the transmission. On the other hand, in SC-PTM, there is no cooperation between BSs and only the UEs best server cells contribute independently to the transmission, therefore, the SINR values are the lowest in such a scheme. Since in dynamic clustering the cluster size is intermediate between those schemes, it leads to moderate SINR gains. As a conclusion, in the proposed scheme, the target blocking probabilities allows to trade capacity against coverage.

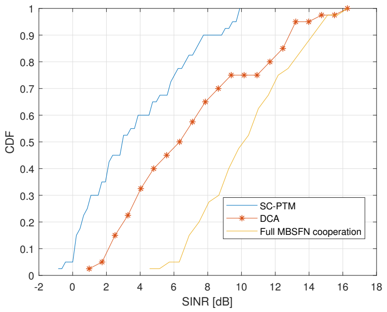


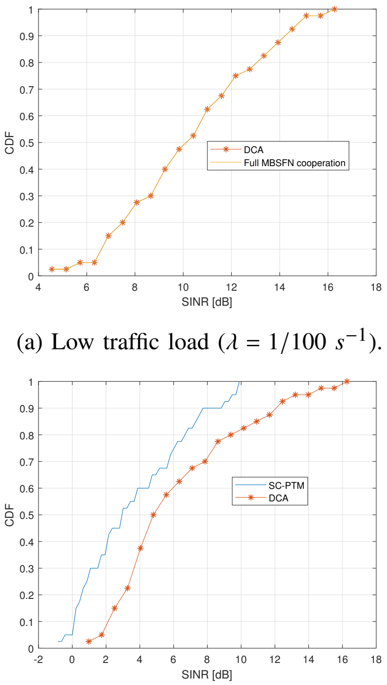
Fig. 9: Mean group SINR distribution with SC-PTM, full MBSFN cooperation and DCA.

E. Impact of Traffic Intensity

We now show how in extreme cases of traffic our scheme tends to full MBSFN cooperation or SC-PTM. In Fig. 10a a very low traffic is considered, so that all cells can cooperate to serve groups without violating the blocking probability constraint. In Fig. 10b, a high traffic is assumed. Cells are selected as in SC-PTM. The SINR is however improved because we assume that serving cells cooperate to serve the groups whereas SC-PTM serving BSs interfere each others. This shows our approach self-adapts to traffic conditions and includes MBSFN and SC-PTM as special cases. Note that SC-PTM and full MBSFN are two simple approaches, so that the advantages of DCA are obtained at the cost of an increased complexity. However, when compared to full MBSFN, DCA can sustain a higher input traffic load given a target blocking probability. When compared to SC-PTM, DCA guarantees to maximize the SINR if there are available resources in other cells and thus increases SINR given the same target blocking probability.

V. Conclusion

In this paper, we formulate a problem to capture the coverage-capacity trade-off arising in mission-critical networks serving group of users by means of cooperative multipoint transmissions. A Dynamic Clustering Algorithm (DCA) is proposed to solve this problem that is made of an inner loop GCCA and an outer loop CWOA. We consider a dynamic traffic in which group calls arrive, are served by a cluster of cells and leave the system. For every group, GCCA minimizes a submodular function that captures the trade-off between the average group SINR and cell weights representing a cost for the radio resource utilization. At a larger time scale, cell costs are optimized by CWOA in order that the blocking probability in every cell is close to a target. We rely here on derivative-free optimization and more specifically on the Nelder-Mead



(a) Low traffic load ($\lambda = 1/100\ s^{-1}$).

(b) High traffic load ($\lambda = 1/25\ s^{-1}$).

Fig. 10: Mean group SINR distribution with low (a) and high (b) traffic intensities.

simplex method that we extend by including a random oriented restart. The comparison of the proposed scheme with SC-PTM and full MBSFN cooperation schemes show that DCA is able to adapt to traffic variations by maximizing the coverage under the constraint of a blocking probability.

Appendix

A. Proof of Lemma 1

The proof is similar to the derivation of the Kaufman-Roberts formula [34]. We assume that there is at least one $m_i \neq 0$. Let $\boldsymbol{m} = (m_1, ..., m_n)$, $x = (n_1, ..., n_P)$, and $\mathcal{E}(\boldsymbol{m}) = \{x \in \mathcal{E} | \forall j,\ x^T v_j = m_j\}$. Now:

$$\begin{aligned}
\zeta(\boldsymbol{m}) &= \sum_{x\in\mathcal{E}(\boldsymbol{m})} \frac{(p_1\rho)^{n_1}}{n_1!} \cdots \frac{(p_P\rho)^{n_P}}{n_P!} \\
&= \sum_{x\in\mathcal{E}(\boldsymbol{m})} \frac{x^T v_i}{m_i} \frac{(p_1\rho)^{n_1}}{n_1!} \cdots \frac{(p_P\rho)^{n_P}}{n_P!} \\
&= \frac{1}{m_i} \sum_{x\in\mathcal{E}(\boldsymbol{m})} \sum_{j=1}^{P} n_j v_{ij} \frac{(p_1\rho)^{n_1}}{n_1!} \cdots \frac{(p_P\rho)^{n_P}}{n_P!}
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{m_i} \sum_{x\in\mathcal{E}(\boldsymbol{m})} \sum_{j=1}^{P} p_j \rho v_{ij} \frac{(p_1\rho)^{n_1}}{n_1!} \cdots \frac{(p_j\rho)^{n_j-1}}{(n_j-1)!} \cdots \frac{(p_P\rho)^{n_P}}{n_P!} \\
&= \frac{1}{m_i} \sum_{j=1}^{P} p_j \rho v_{ij} \sum_{x\in\mathcal{E}(\boldsymbol{m})} \frac{(p_1\rho)^{n_1}}{n_1!} \cdots \frac{(p_j\rho)^{n_j-1}}{(n_j-1)!} \cdots \frac{(p_P\rho)^{n_P}}{n_P!} \\
&= \frac{1}{m_i} \sum_{j=1}^{P} p_j \rho v_{ij} \zeta(m_1 - v_{1j}, ..., m_n - v_{nj}).
\end{aligned}$$

B. Proof of Proposition 3

For a user $u \in \mathcal{U}$, we first show that γ_u is supermodular. Take $A \subseteq B \subseteq V$, $e \in V \backslash B$ and denote: $\gamma_u(A) = \frac{a_1}{a_2}$, $\gamma_u(B) = \frac{b_1}{b_2}$ and $c \geq 0$ the useful power received by u from e, so that $\gamma_u(A \cup e) = \frac{a_1+c}{a_2-c}$ and $\gamma_u(B \cup e) = \frac{b_1+c}{b_2-c}$. From the inclusions of sets and the conservation of the system energy, we have $a_1 \leq b_1$, $a_2 \geq b_2$, $a_1 + a_2 = b_1 + b_2$, $a_2 - c \geq 0$ and $b_2 - c \geq 0$. From the two last inequalities, we deduce that $a_2 + b_2 \geq 2c$. Now the inequality $\gamma_u(A \cup e) - \gamma_u(A) \leq \gamma_u(B \cup e) - \gamma_u(B)$ is equivalent to $a_2 + b_2 \geq c$, which is always verified whatever $A \subseteq B \subseteq V$, $e \in V \backslash B$ and $u \in \mathcal{U}$. We have indeed: $\gamma_u(A \cup e) - \gamma_u(A) = \frac{c(a_1+a_2)}{a_2(a_2-c)}$ and $\gamma_u(B \cup e) - \gamma_u(B) = \frac{c(b_1+b_2)}{b_2(b_2-c)}$, so that the inequality is successively equivalent to:

$$\begin{aligned}
b_2(a_1 + a_2)(b_2 - c) &\leq a_2(b_1 + b_2)(a_2 - c) \\
b_2^2 a_1 + b_2^2 a_2 - b_2 a_1 c - b_2 a_2 c &\leq a_2^2 b_1 + a_2^2 b_2 - a_2 b_1 c - a_2 b_2 c \\
b_2^2(a_1 + a_2) - a_2^2(b_1 + b_2) &\leq c(b_2(a_1 + a_2) - a_2(b_1 + b_2)) \\
a_2 + b_2 &\geq c
\end{aligned}$$

By non-negative linear combination, we deduce that $\bar{\gamma}_{\mathcal{U}}$ is also supermodular and $-\bar{\gamma}_{\mathcal{U}}$ is submodular. The function $\Psi_{\mathcal{U}}$ is the sum of a submodular function and a modular function, so it is submodular.

C. Affine Minimizer

The problem to be solved in step 9 of Algorithm 3: $y = \arg\min_{z\in \text{aff}(S)} ||z||$. Let $S = \{q_1, ..., q_m\}$, $\alpha = [\alpha_1 \ldots \alpha_m]^T$, $Q = [q_1\ q_2\ \ldots\ q_m]$, and $A = 2Q^T Q$. Define the affine hull of S as follows: $\text{aff}(S) \triangleq \{\sum_{i=1}^m \alpha_i q_i | q_i \in S \text{ and } \sum_{i=1}^m \alpha_i = 1\}$. We remark that: $||\sum_{i=1}^m \alpha_i q_i||^2 = ||Q\alpha||^2 = (Q\alpha)^T(Q\alpha) = \alpha^T Q^T Q\alpha$. So that the problem in step 9 of the algorithm becomes: $\min \frac{1}{2}\alpha^T A\alpha$ s.t. $\mathbf{1}^T\alpha = 1$ and $\alpha \in \mathbb{R}^n$. There is a unique solution to this problem if and only if the matrix $\begin{bmatrix} 0 & \mathbf{1}^T \\ \mathbf{1} & Q^T Q \end{bmatrix}$ is non singular, i.e., if the points of Q are affinely independent. Using results of [35] on the quadratic minimization problem with linear constraint, the solution is given by: $\alpha^* = \frac{A^{-1}\mathbf{1}}{\mathbf{1}^T A^{-1}\mathbf{1}}$, so that $y = Q\alpha^*$ and $||y||^2 = \frac{1}{2}\frac{1}{\mathbf{1}^T A^{-1}\mathbf{1}}$. Note that for any $v := Q\alpha$ such that $\mathbf{1}^T\alpha = 1$ (i.e., in the affine hull of S), we have $v^T y = \alpha^T Q^T Q\alpha^* = \frac{1}{2}\frac{\alpha^T AA^{-1}\mathbf{1}}{\mathbf{1}^T A^{-1}\mathbf{1}} = ||y||^2$.

References

[1] TCCA, “Critical Communications and Mobile Network Operators,” The Critical Communications Association (TCCA), Tech. Rep., May 2018.

[2] Y. Lair and G. Mayer, “Mission Critical Services in 3GPP,” https://www.3gpp.org/news-events/1875-mc_services, Jun. 2017.

[3] GSMA, “Network 2020: Mission Critical Communications,” GSM Association, Tech. Rep., Mar. 2017.

[4] 3GPP, “Mission Critical Services Common Requirements,” 3rd Generation Partnership Project, TS 22.280, Jul. 2018.

[5] 3GPP, “NR; Base Station Radio Transmission and Reception,” 3rd Generation Partnership Project, TS 38.104, Sep. 2018.

[6] 3GPP, “Support of Single-Cell Point-To-Multipoint transmission in LTE,” 3rd Generation Partnership Project, RP RP.151110, Jun. 2015.

[7] A. Daher, M. Coupechoux, P. Godlewski, P. Ngouat, and P. Minot, “SC-PTM or MBSFN for Mission Critical Communications?” in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, 2017.

[8] A. Papadogiannis, D. Gesbert, and E. Hardouin, “A Dynamic Clustering Approach in Wireless Networks with Multi-Cell Cooperative Processing,” in *2008 IEEE International Conference on Communications*, 2008, pp. 4033–4037.

[9] M. Yoon, M.-S. Kim, and C. Lee, “A Dynamic Cell Clustering Algorithm for Maximization of Coordination Gain in Uplink Coordinated System,” *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 1752–1760, 2016.

[10] Y. Du and G. de Veciana, “”Wireless networks without edges”: Dynamic radio resource clustering and user scheduling,” in *2014 IEEE Conference on Computer Communications (INFOCOM)*, 2014, pp. 1321–1329.

[11] P. Baracca, F. Boccardi, and N. Benvenuto, “A dynamic clustering algorithm for downlink CoMP systems with multiple antenna UEs,” *EURASIP Journal on Wireless Communications and Networking*, pp. 1–14, 2014.

[12] S. Scholz, “Combining dynamic clustering and scheduling for coordinated multi-point transmission in LTE,” in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2017, pp. 1–7.

[13] D. Liu, S. Han, C. Yang, and Q. Zhang, “Semi-dynamic User-Specific Clustering for Downlink Cloud Radio Access Network,” *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 2063–2077, 2016.

[14] V. Garcia, Y. Zhou, and J. Shi, “Coordinated Multipoint Transmission in Dense Cellular Networks With User-Centric Adaptive Clustering,” *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4297–4308, 2014.

[15] L. Liu, Y. Zhou, V. Garcia, L. Tian, and J. Shi, “Load Aware Joint CoMP Clustering and Inter-Cell Resource Scheduling in Heterogeneous Ultra Dense Cellular Networks,” *IEEE Transactions on Vehicular Technology*, vol. 67, pp. 2741–2755, 2018.

[16] S. Bassoy, M. Jaber, M. A. Imran, and P. Xiao, “Load Aware Self-Organising User-Centric Dynamic CoMP Clustering for 5G Networks,” *IEEE Access*, vol. 4, pp. 2895–2906, 2016.

[17] Z. Zhang, N. Wang, J. Zhang, X. Mu, and K. M. Wong, “Cooperation Resource Efficient User-Centric Clustering for QoS Provisioning in Uplink CoMP,” in *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2017, pp. 1–5.

[18] L. Rong, S. E. Elayoubi, and O. B. Haddada, “Performance evaluation of cellular networks offering TV services,” *IEEE Transactions on Vehicular Technology*, vol. 60, no. 2, pp. 644–655, 2011.

[19] J. Chen, M. Chiang, J. Erman, G. Li, K. Ramakrishnan, and R. K. Sinha, “Fair and optimal resource allocation for LTE multicast (eMBMS): Group partitioning and dynamics,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 1266–1274.

[20] L. Rong, O. B. Haddada, and S. E. Elayoubi, “Analytical analysis of the coverage of a MBSFN OFDMA network,” *GLOBECOM - IEEE Global Telecommunications Conference*, pp. 2388–2392, 2008.

[21] G. Caire, S. A. Ramprashad, and H. C. Papadopoulos, “Rethinking Network MIMO: Cost of CSIT, Performance Analysis, and Architecture Comparisons,” in *2010 Information Theory and Applications Workshop (ITA)*. IEEE, 2010, pp. 1–10.

[22] 3GPP, “Physical Layer Procedures,” 3rd Generation Partnership Project, TS 36.213, Sep. 2018.

[23] F. Bach, “Learning with Submodular Functions: A Convex Optimization Perspective,” 2011.

[24] S. Fujishige, “Lexicographically Optimal Base of a Polymatroid with Respect to a Weight Vector,” *Mathematics of Operations Research*, vol. 5, no. 2, pp. 186–196, 1980.

[25] P. Wolfe, “Finding the nearest point in A polytope,” *Mathematical Programming*, vol. 11, no. 1, pp. 128–149, 1976.

[26] S. Fujishige, “Submodular systems and related topics,” in *Mathematical Programming at Oberwolfach II*, 1984, pp. 113–131.

[27] D. Chakrabarty, P. Jain, and P. Kothari, “Provable Submodular Minimization using Wolfe’s Algorithm,” in *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, vol. 2014, no. 1, 2014, pp. 1–9.

[28] S. T. McCormick, “Submodular Function Minimization,” *Handbooks in Operations Research and Management Science*, vol. 12, pp. 321–391, 2005.

[29] J. A. Nelder and R. Mead, “A Simplex Method for Function Minimization,” *The Computer Journal*, vol. 7, no. 4, pp. 308–313, 1965.

[30] C. T. Kelley, “Iterative Methods for Optimization,” *Siam*, p. 188, 1999.

[31] M. A. Luersen and R. Le Riche, “Globalized Nelder-Mead method for engineering optimization,” *Computers and Structures*, vol. 82, no. 23-26, pp. 2251–2260, 2004.

[32] S. Singer, “Complexity analysis of Nelder-Mead search iterations,” *Conference on Applied Mathematics and Computation*, pp. 185–196, 1999.

[33] S. Fujishige, T. Hayashi, and S. Isotani, “The Minimum-Norm-Point Algorithm Applied to Submodular Function Minimization and Linear Programming,” *Kyoto University, Research Institute for Mathematical Sciences [RIMS]*, pp. 1–19, 2006.

[34] J. W. Roberts, “A service system with heterogeneous user requirements,” in *Performance of Data Communications systems and their applications*, vol. 29, no. 10, 1981, pp. 423–431.

[35] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

Alaa Daher is R&D Engineer at ETELM. He received the Masters degree from Pierre and Marie Curie University and Telecom Paris, France, in 2013, and his Ph.D. from Telecom Paris, France, in 2019. His main interests include quality of service assessment and multi-criteria optimization in 5G networks, next generation mission-critical communications and self organizing cellular networks.

Marceau Coupechoux is Professor at Telecom Paris and École Polytechnique. He obtained Masters’ degrees from Telecom Paris (1999) and University of Stuttgart (2000), his Ph.D. from Institut Eurecom (2004), and his Habilitation from Sorbonne University (2015). He was Visiting Scientist at IISc Bangalore during 2011-2012. He is working on wireless networks optimization and resource allocation.

Philippe Godlewski received the Engineers degree from Telecom Paris (formerly ENST) in 1973 and the PhD in 1976 from University Paris 6. He is Professor at Telecom Paris in the Computer and Network Science department. His fields of interest include Cellular Networks, Air Interface Protocols, Multiple Access Techniques, Error Correcting Codes, Communication and Information Theory.

Pierre Ngouat received his PhD from Suplec/University of Paris XI (1991) and is a former student of ESSEC Business School. He has been among the pioneers of GSM R&D and PMR in Alcatel and then in ARANE/ETELM SAS. From 2000 onwards, he has developed the ALTRAN R&D Telecom skill center. Since 2006, Pierre is engaged with PNG TECHNOLOGIES in research, promotion and development of innovative products and processes for the mobile networks sector.

Pierre Minot is President for ETELM, a manufacturer of mission-critical communications infrastructure technology (TETRA, LTE) based in France. He founded ETELM in 1981 after having spent over 15 years in several companies in the Telecom industry. He is recognized by his peers as a visionary expert and a respected speaker in the radio communication sector. Pierre has an Engineers degree in Electronics and Telecommunications.