Joint Positioning of Flying Base Stations and Association of Users: Evolutionary-based Approach

JAN PLACHY, (Student Member, IEEE), ZDENEK BECVAR, (Senior Member, IEEE), PAVEL MACH, (Member, IEEE), RADEK MARIK, and MICHAL VONDRA
Department of Telecommunication Engineering, FEE, Czech Technical University in Prague Technicka 2, 166 27 Prague, Czech Republic (e-mails: jan.plachy@fel.cvut.cz, zdenek.becvar@fel.cvut.cz, machp2@fel.cvut.cz, marikr@fel.cvut.cz, michal.vondra@fel.cvut.cz)

Corresponding author: Jan Plachy (e-mail: jan.plachy@fel.cvut.cz).

ABSTRACT Time-varying requirements of users on communication push mobile operators to increase density of base stations. However, the dense deployment of conventional static base stations (SBSs) is not always economical, for example, when periods of peak load are short and infrequent. In such cases, several flying base stations (FlyBSs) mounted on unmanned aerial vehicles can be seen as a convenient substitution for the dense deployment of SBSs. This paper focuses on a maximization of user satisfaction with provided data rates. To this end, we propose an algorithm that associates users with the most suitable SBS/FlyBS and finds optimal positions of all FlyBSs. Furthermore, we investigate the performance of two proposed approaches for the joint association and positioning based on the genetic algorithm (GA) and particle swarm optimization (PSO). It is shown that both solutions improve the satisfaction of users with provided data rates in comparison with a competitive approach. We also demonstrate trade-offs between the GA and the PSO. While the PSO is of a lower complexity than the GA, the GA requires a slightly lower number of active FlyBSs to serve the users.

INDEX TERMS Flying base station, genetic algorithm, mobile networks, particle swarm optimization, UAV, user satisfaction.

I. INTRODUCTION

THE increasing requirements of mobile users on wireless communication call for an ultra-dense deployment of static base stations (SBSs) [1]. However, this solution is not always reasonable from an economic point of view. An example of an uneconomical case is an event in which people stay for a short period of time (a few hours) and then move away. This situation occurs mostly during large social events, such as live concerts or sport activities. In these cases, exploitation of the BSs mounted on unmanned aerial vehicles (UAVs), also known as flying BSs (FlyBSs), is beneficial [2]. Nevertheless, a deployment of the FlyBSs brings many challenges, such as finding an optimal position of each FlyBS, properly associating user equipments (UEs) to the FlyBSs [3], mitigating interference [4], [5], and deciding how many FlyBSs should be deployed in a given area [6], [7].

The current research addressing the problem of FlyBS positioning can be divided into works dealing with a single FlyBS or those concerning multiple FlyBSs. The positioning of a single FlyBS is considered in, e.g., [2], [8]–[11]. The objective of the authors in [8] is to find the optimal position of the FlyBS to maximize the data rate of the UE. The FlyBS is seen as a relay between the SBS and the UE. The goal is, then, achieved by a designed algorithm, which finds a position of the FlyBS so that line of sight (LOS) communication takes place on both BS-FlyBS and FlyBS-UE links. The optimal 3D placement of the FlyBS to provide coverage to all UEs is proposed in [9]. The authors derive an optimal position of the FlyBS based on a geometry and a path loss model for a single FlyBS. Similarly, in [10], the authors propose an optimal 3D placement algorithm for a single FlyBS. The algorithm performs an exhaustive search over a closed region to find the 3D position of one FlyBS. The authors assume heterogeneous quality of service (QoS) requirements, represented by signal to noise ratio (SNR). The positioning is investigated also in [2], where the authors

confirm improvement in channel quality, throughput, and energy efficiency by the FlyBS positioned according to the UEs' requirements in a scenario with the UEs moving in a crowd. A joint optimization of the FlyBS's position, bandwidth allocation, transmission power, and transmission rate is proposed in [11]. The authors transform a non-convex optimization problem into a monotonic optimization and solve the problem via the polyblock algorithm [12]. A drawback of all above-mentioned works ([2], [8]–[11]) is that these assume just one FlyBS, and their extension toward multiple FlyBSs is neither easy nor straightforward.

The positioning of multiple FlyBSs is proposed, for example, in [13], where the objective is to provide SNR above a predefined threshold. The authors solve the positioning via linear programming. The drawback of the proposed approach is that it does not take interference into account. Thus, without managing the problem of interference (e.g., by interference alignment technique [14]), such simplification leads to a general coverage optimization problem (as addressed in [15]–[20]). The interference among individual FlyBSs is assumed in [21], where the authors propose an algorithm for optimization of the FlyBSs' 3D positions. Thus, when compared to the previous papers, the positioning of the FlyBSs is based on signal to interference plus noise ratio (SINR) of the UEs instead of SNR. The authors focus on the stochastic geometry approach; however, actual data requirements of the UEs are not considered.

Furthermore, evolutionary algorithms [22] can be exploited for the positioning of FlyBSs, as considered in [23]–[25]. To be more specific, the authors in [23] adopt particle swarm optimization (PSO) [26] for the positioning. The PSO finds an optimal solution via an evolutionary process inspired by nature, which acts similar to the flocking of birds or swarms of insects. The authors in [23] propose an algorithm for the positioning of the FlyBSs to provide coverage to all UEs, considering connection quality of the FlyBSs' backhaul. The authors focus on a provisioning of a certain level of SINR for the UEs, but they do not consider the allocation of bandwidth to the UEs. The PSO is also exploited in [24], where the authors find an optimal placement of the FlyBSs to satisfy the UEs' required SINR with a minimum number of the deployed FlyBSs. However, the authors do not tackle the problem of bandwidth allocation, which is a critical factor to satisfy the data rates required by the UEs. Furthermore, the authors define SINR requirements as the ratio of the area covered by two or more FlyBSs to the sum of the FlyBSs' coverage areas. Such definition leads to coverage optimization instead of UE data rate satisfaction. Another evolutionary algorithm, the genetic algorithm (GA) [22], is used in [25] for an optimization of the trajectories of the UAVs. The authors show that the proposed solution based on the GA is efficient and can be run on a graphical processing unit (GPU), exploiting parallel architecture of the GPU. However, the paper does not consider communication of the UAVs with the UEs.

Regarding the problem of the UEs' association to the Fly-

BSs, the authors in [27] propose two algorithms to partition an area served by the FlyBSs via association of the UEs to the FlyBSs. The objective of the first algorithm, based on optimal transport theory, is to maximize a fairness of the UEs' data rates under a hovering time constraint. The purpose of the second iterative algorithm is to determine a minimal hovering time to satisfy the UEs' data rate requirements. Nevertheless, the positioning of the FlyBS to improve the UEs' satisfaction is not addressed in [27].

The main disadvantage of all above-mentioned works is that these try to either solely optimize positioning of the FlyBSs or association of the UEs. Nonetheless, the positioning of the FlyBSs and the association of the UEs should be optimized jointly, as these two challenges are closely related. The joint positioning of the FlyBSs and association of the UEs is addressed in [28], where the problem is translated into a clustering problem. This problem is solved by the k-means algorithm, which determines positions of the FlyBSs and associations of the UEs, respectively. The k-means clusters the UEs and associates them to the FlyBSs based on the Euclidean distance. However, the k-means does not incorporate any information regarding the communication channel, which is of paramount importance for the deployment of the FlyBSs.

In this paper, we focus on a joint solution for the positioning of the FlyBSs and the association of the UEs, exploiting also information related to the communication channel. We propose two novel algorithms for the joint positioning and association, considering the UEs' requirements on data rates. The first developed algorithm for the joint positioning and association is based on the PSO, while the second exploits the GA. Unlike other works, our objective is to maximize the UEs' satisfaction with the provided data rates. We show that the proposed joint positioning and association based on both PSO and GA notably outperforms a competitive state-of-the-art algorithm if the same amount of FlyBSs is deployed. We also discuss trade-offs between the PSO-based and GA-based solutions and assess their pros and cons.

The remainder of this paper is organized as follows. In the next section, we define system model and formulate the problem. In Section III, the proposed algorithms for the association and the positioning are described, and implementation aspects are discussed. The simulation scenario, a description of the competitive algorithm, and the performance evaluation are provided in Section IV. The last section summarizes major conclusions.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first define the system model for the positioning of the FlyBSs and for the association of the UEs. Then, we formulate the objective of the paper.

A. SYSTEM MODEL

We consider a set $\boldsymbol{N}$ of N UEs, where $n \in \boldsymbol{N}$ is a specific UE, a set $\boldsymbol{K}^S$ of K^S representing conventional SBSs, and a set $\boldsymbol{K}^F$ of K^F corresponding to the FlyBSs. Furthermore, we

define a set of all base stations (BSs) as $\boldsymbol{K} = \boldsymbol{K}^S \cup \boldsymbol{K}^F$ with $K = K^S + K^F$ representing the total number of BSs. Note that the label "BS" represents both the SBSs and the FlyBSs in this paper. The positions of the BSs are defined as $\boldsymbol{V} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_K\}$, where $\boldsymbol{v}_k \in \mathbb{R}^3$, $k \in \boldsymbol{K}$ represents a position of the k-th BS. In the same way, we define a set of the UEs' positions $\boldsymbol{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_N\}$ with the position of the n-th UE denoted as $\boldsymbol{u}_n \in \mathbb{R}^3$, $n \in \boldsymbol{N}$. An activity status of the BS is indicated by a binary parameter, $\boldsymbol{\rho} = \{\rho_1, \dots, \rho_K\}$. Setting $\rho_k = 1$ and $\rho_k = 0$ means that the k-th BS is being turned on and off, respectively.

The n-th UE and the k-th BS communicate over a radio channel, with SINR defined as:

$$\gamma_{n,k} = \frac{\rho_k P_k^{tx} |h_{n,k}(\boldsymbol{u}_n, \boldsymbol{v}_k)|^2}{\beta_{n,k}\sigma^2 + \sum_{l \in \boldsymbol{K}, l \neq k} \rho_l P_l^{tx} |h_{n,l}(\boldsymbol{u}_n, \boldsymbol{v}_l)|^2}, \quad (1)$$

where P_k^{tx} is the transmission power of the k-th BS, $h_{n,k}(\boldsymbol{u}_n, \boldsymbol{v}_k)$ is the channel realization between the k-th BS and the n-th UE, σ^2 is the noise power, and $\beta_{n,k} \in \boldsymbol{\beta} | \beta_{n,k} \in \{0, 1\}, \forall n \in \boldsymbol{N}, \forall k \in \boldsymbol{K}$ is the amount of bandwidth allocated for communication of the n-th UE with the k-th BS. The matrix $\boldsymbol{\beta} \in \boldsymbol{B}$ contains the bandwidth allocations of all BSs, with $\boldsymbol{B}$ representing a set of all feasible bandwidth allocations. Note that SINR is calculated only for active BSs (i.e., the BSs with $\rho_k = 1$). The bandwidth allocation also contains information about the UE's association to the BSs. The UE is considered to be associated to the BS to which it has non-zero $\beta_{n,k}$. We assume that the UE is associated to a single BS (i.e., the bandwidth for each UE is allocated at most to one BS). Then, the data rate provided by the k-th BS to the n-th UE via channel with the bandwidth B_k is defined as:

$$c_{n,k} = \beta_{n,k} B_k \log_2(1 + \gamma_{n,k}) \quad (2)$$

B. OBJECTIVE FORMULATION

Our objective is to find the positions of the FlyBSs and associate the UEs to the BSs in order to maximize the number of UEs satisfied with their experienced data rate. Without loss of generality, we focus on downlink direction. Note that the n-th UE is assumed to be satisfied if it experiences data rate $c_{n,k}$ equal to or higher than the minimum required data rate c_n^{min} (i.e., the UE is satisfied if $c_{n,k} \geq c_n^{min}$). The BSs that are unused or cannot improve the UEs' satisfaction are turned off to save energy. Thus, our objective is to determine the optimal positions of the FlyBSs $\boldsymbol{V}^*$, the association of the UEs $\boldsymbol{\beta}^*$ (represented via bandwidth allocation), and the status of the BSs (on/off) $\boldsymbol{\rho}^*$. This objective is formulated as:

$$\boldsymbol{\beta}^*, \boldsymbol{V}^*, \boldsymbol{\rho}^* = \underset{\boldsymbol{\beta} \in \boldsymbol{B}, \boldsymbol{V} \in \mathbb{R}^{3 \times K}, \rho_k \in \{0,1\}}{\operatorname{argmax}} \sum_{n \in \boldsymbol{N}} \sum_{k \in \boldsymbol{K}} \left[c_{n,k} \geq c_n^{min} \right]$$
$$(3)$$

$$\text{subject to } \sum_{n \in \boldsymbol{N}} \beta_{n,k} \leq 1, \forall k \in \boldsymbol{K}, \quad (4)$$

$$\sum_{k \in \boldsymbol{K}} [\beta_{n,k} > 0] \leq 1, \forall n \in \boldsymbol{N}, \quad (5)$$

where the operator [.] is equal to 1 if the condition (e.g., $c_{n,k} \geq c_n^{min}$) is fulfilled, otherwise it is equal to 0. The constraint (4) ensures that the BSs do not allocate more bandwidth than available. Furthermore, the constraint (5) ensures that each UE can be associated to a maximum of one BS.

III. PROPOSED SOLUTION

The defined objective is an NP-hard problem (due to its definition as a non-convex function). Hence, to find the optimal positions of the FlyBSs, we exploit two evolutionary algorithms: GA and PSO [22]. The evolutionary algorithms iteratively search for the optimum within the search space, using several operations introduced later in this section.

First, we describe a general algorithm for the association of the UEs to the BSs, including a bandwidth allocation and a decision on the number of active BSs. Then, we integrate the association algorithm into the proposed algorithms for positioning of the FlyBSs based on the PSO and the GA, respectively. Last, a discussion of the practical implementation aspects is provided at the end of this section.

A. ASSOCIATION OF THE UES AND BANDWIDTH ALLOCATION

The objective of the UEs' association is to determine the serving BS for each UE and to allocate bandwidth to the UEs to satisfy the UEs' required data rates c_n^{min} (i.e., each UE is allocated exactly with the bandwidth required to reach the c_n^{min}). Then, based on the association, we decide which BSs should be turned off, as those BSs do not improve the UEs' satisfaction. Note that we do not target a problem of minimization of the number of active BSs. Such a problem is not straightforward, and possible extension of our proposed algorithms is left for future research.

The proposed association of the UEs and the bandwidth allocation is described in Algorithm 1. In the initial phase, the active FlyBSs are randomly deployed within the area (line 1). The SINR between each BS and UE is calculated according to (1) from a path loss model, following the same approach as the authors in [23] (line 4).

Then, the n-th UE is temporarily associated to the serving BS s_n (i.e., to the BS with the highest experienced SINR to minimize the bandwidth required to satisfy the UE's c_n^{min} (line 6)). Based on the temporal association, a set of vectors $\boldsymbol{A} = \{\boldsymbol{A}_1, \boldsymbol{A}_k, \dots, \boldsymbol{A}_K\}$ is created. Each vector $\boldsymbol{A}_k$ from $\boldsymbol{A}$ represents a list of the UEs associated to individual BSs. The list of UEs is created by adding the n-th UE to the vector $\boldsymbol{A}_{s_n}$ corresponding to the serving BS s_n (i.e., $\boldsymbol{A}_{s_n} \leftarrow \boldsymbol{A}_{s_n} \cup n$ (line 7)). Next, we create the set $\boldsymbol{K}'$ containing indices of the set $\boldsymbol{A}$ sorted according to the number of UEs served by each BS in descending order (line 9). For purposes of the bandwidth allocation in the next steps, a temporal set, $\boldsymbol{N}' \subseteq \boldsymbol{N}$, is created (line 10). Subsequently, $\boldsymbol{A}_k$ is emptied for each BS in $\boldsymbol{K}$ (line 11). Based on the ordered set $\boldsymbol{K}'$, the bandwidth for communication is allocated to the UEs to

fulfill the UEs' data rate requirements. The BSs allocate the bandwidth until no bandwidth is left (lines 13 to 24).

The bandwidth is allocated according to the UEs' SINR in descending order (i.e., the UE with the highest SINR is allocated first). In terms of the algorithm, the UE n^* with the highest $\gamma_{n^*,k'}$ is selected first (line 15). Next, the bandwidth required to satisfy the UE n^* at the k-th BS is calculated as $\beta^{req} = c_{n^*}^{min}/log_2\left(1 + \gamma_{n^*,k'}\right)$ (line 16). If the k'-th BS has enough bandwidth, the β^{req} is allocated to the UE n^* and the UE n^* is associated to the k'-th BS (lines 17 to 19). The associated UE n^* is removed from the set $\boldsymbol{N'}$, and the available bandwidth of the k'-th BS is updated (lines 20 and 21). The bandwidth allocation continues until there is not enough remaining bandwidth that can satisfy any UE. Then, the remaining bandwidth of each BS is divided among all the UEs served by the given BS (line 25). Finally, bandwidth allocation is divided by the B_k to obtain normalized bandwidth allocation $\beta_{n,k}$ (line 27), and the BSs serving no UE are turned off (line 28).

Algorithm 1 Association of UEs and bandwidth allocation.

1: Deploy FlyBSs by generating random positions $\boldsymbol{V}$; set $\rho_k = 1, \forall k \in \boldsymbol{K}$.
2: **for** $n \in \boldsymbol{N}$
3:      **for** $k \in \boldsymbol{K}$
4:          Calculate $\gamma_{n,k}$ via (1).
5:      **end for**
6:      $s_n \leftarrow \operatorname{argmax}_{k \in \boldsymbol{K}} \gamma_{n,k}$
7:      $\boldsymbol{A}_{s_n} \leftarrow \boldsymbol{A}_{s_n} \cup n$
8: **end for**
9: $\boldsymbol{K'} \leftarrow$ indices of $\boldsymbol{A}$ sorted in descending order.
10: $\boldsymbol{N'} \leftarrow \boldsymbol{N}$
11: $\boldsymbol{A}_k \leftarrow \emptyset, \forall k \in \boldsymbol{K}$
12: **for** $k' \in \boldsymbol{K'}$
13:      **while** $B_{k'} > 0$
14:          **for** $n \in \boldsymbol{N'}$
15:              $n^* \leftarrow \operatorname{argmax}_{n \in \boldsymbol{N'}} \gamma_{n,k'}$
16:              $\beta^{req} = \frac{c_{n^*}^{min}}{log_2\left(1+\gamma_{n^*,k'}\right)}$
17:              **if** $B_{k'} \geq \beta^{req}$
18:                  $\beta_{n^*,k'} \leftarrow \beta^{req}$
19:                  $\boldsymbol{A}_{k'} \leftarrow \boldsymbol{A}_{k'} \cup n^*$
20:                  $\boldsymbol{N'} \leftarrow \boldsymbol{N'} \backslash n^*$
21:                  $B_{k'} \leftarrow B_{k'} - \beta^{req}$
22:              **end if**
23:          **end for**
24:      **end while**
25:      $\beta_{n',k} \leftarrow \beta_{n',k} + \frac{B_{k'}}{|\boldsymbol{A}_{k'}|}, \forall n \in \boldsymbol{A}_{k'}$
26: **end for**
27: $\beta_{n,k} \leftarrow \frac{\beta_{n',k}}{B_k}, \forall k \in K$
28: $\{\rho_k \leftarrow 0 | \forall k' \in \boldsymbol{K'}, \boldsymbol{A}_{k'} = \emptyset\}$

B. POSITIONING OF FLYBSS VIA PARTICLE SWARM OPTIMIZATION

In this subsection, we describe the proposed algorithm for optimization of the FlyBSs' positions based on the PSO and its integration with the association. We exploit a common PSO described in [26] and adapt it to the objective defined in (3). The PSO searches for the optimal solution via a set of $l \in L$ particles $\{\boldsymbol{W}^1(t), \boldsymbol{W}^2(t), \dots, \boldsymbol{W}^L(t)\}$ over iterations represented by the discrete time t. In our case, each particle contains the positions of all FlyBSs (i.e., $\boldsymbol{W}^l(0) = \boldsymbol{V}$). The search is done by updating the positions of the FlyBSs via a velocity vector $\boldsymbol{D}^l(t)$ calculated as:

$$
\begin{aligned}
\boldsymbol{D}^l(t) = \phi\boldsymbol{D}^l(t-1) + c_p\phi_1\left(\boldsymbol{W}^{l,local} - \boldsymbol{W}^l(t-1)\right) + \\
c_g\phi_2\left(\boldsymbol{W}^{global} - \boldsymbol{W}^l(t-1)\right),
\end{aligned}
\tag{6}
$$

where ϕ is the inertia weight determining the convergence speed, ϕ_1 and ϕ_2 are positive random variables, and c_p and c_g are the personal and global learning coefficients, respectively. The velocity vector represents a weighted sum of the previous velocity vector $\boldsymbol{D}^l(t-1)$, the difference between the FlyBSs' positions of the l-th particle $\boldsymbol{W}^l(t-1)$, and the l-th particle's local best solution $\boldsymbol{W}^{l,local}$ (i.e., historically the best FlyBSs' positions of the l-th particle), and the difference between the l-th particle $\boldsymbol{W}^l(t-1)$ and the global best solution $\boldsymbol{W}^{global}$. The global best solution $\boldsymbol{W}^{global}$ contains the best particle (i.e., the position of the FlyBSs with the highest targeted metric) out of all particles L, up to the current iteration t. In our objective, $\boldsymbol{D}^l(t)$ is a directional vector of the l-th particle, represented by the positions of all FlyBSs between the time instants t and $t-1$. Note that $\boldsymbol{D}^l(t)$ is calculated separately for each FlyBS of the l-th particle.

An example of the FlyBS position update is shown in Figure 1, where a single selected FlyBS at a position from the corresponding l-th particle $\boldsymbol{W}^l(t-1)$ is updated by $\boldsymbol{D}^l(t)$, considering the local and the global best positions of the selected FlyBS according to (6).

Each particle has its suitability represented by a cost function stored in Q^l. The suitability of the FlyBSs' positions and the UE's association of the l-th particle is defined by the cost function Q^l, reflecting our objective to maximize the UEs' satisfaction according to (3). Thus, the cost function is
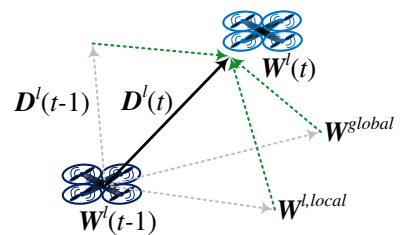

FIGURE 1: Update of FlyBS position via the proposed algorithm based on PSO.

formulated as:

$$Q^l = \begin{cases} \sum_{n \in \boldsymbol{N}} \sum_{k \in \boldsymbol{K}} c_{n,k} & \text{if } c_{n,k} \geq c_n^{min}, \forall n \in N \\ \sum_{n \in \boldsymbol{N}} \sum_{k \in \boldsymbol{K}} \left[c_{n,k} \geq c_n^{min} \right] & \text{otherwise.} \end{cases}$$
(7)

The search for the optimal solution of the objective function is then achieved by updating the positions of the FlyBSs corresponding to each particle $\boldsymbol{W}^l(t)$ via a maximization of the particles' local best cost ($Q^{l,local}$) and a global best cost (Q^{global}). In other words, the new position of the FlyBS is determined based on the best position of the given FlyBS represented by the l-th particle in the past, and the FlyBS's best position among all particles L. If all UEs are satisfied with the provided data rates, the remaining bandwidth is allocated to the UEs so that the sum of the UEs' data rates is maximized.

The proposed algorithm for the positioning based on the PSO with integration of the UEs' association is described in Algorithm 2. The PSO algorithm starts with the updated association with the unused BSs turned off (as explained in the previous subsection) (line 1). Based on the association, the particles are initialized (line 2). Then, the cost function of each particle is calculated via (7) (line 3). The particle with the highest cost is set as $\boldsymbol{W}^{global}$ and the cost of this particle is set to Q^{global} (lines 4 and 5). Then, the PSO iteratively updates the FlyBSs' positions until a maximum number of iterations M_{it} is reached (line 8). For each updated FlyBSs' position, the UEs' are re-associated (lines 9). Based on the updated FlyBSs' positions and the UEs' association, a suitability of the particle is evaluated via the cost function (7) (line 10). Then, we check if the updated positions improve the local solution (lines 11 to 13) or even the global solution (lines 15 to 17). Once all M_{it} iterations are completed, the $\boldsymbol{W}^{global}$ contains the set of FlyBSs' positions with the highest cost (suitability).

C. POSITIONING OF FLYBSS VIA GENETIC ALGORITHM

In this subsection, we describe the proposed algorithm for optimization of the FlyBSs' positions based on the GA. We exploit a common GA described in [22] and adapt it to the optimization problem defined in (3). The GA consists of a population $\boldsymbol{G} = \{\boldsymbol{g}^1, \ldots, \boldsymbol{g}^L\}$ with a size L. The population is composed of individuals $\boldsymbol{g}^l$ representing possible solutions (i.e., sets of the positions of the FlyBSs). Each individual consists of genes $\boldsymbol{g}_k^l$ corresponding to the positions of FlyBSs (i.e., $\boldsymbol{g}_k^l = \boldsymbol{v}_k$).

The first step of the GA is to generate an initial population with the size L. After that, a crossover operation inherent to all genetic algorithms is applied to the initial population. The crossover operation is understood as a mechanism during which new offspring are created from two selected parents. While each parent represents one of the previous positions of the given FlyBS, the new offspring defines a new possible position of the FlyBS. The selection of the parents j_1 and j_2 is done via roulette wheel selection (RWS). The RWS selects parents based on their probability of survival defined by the

---

Algorithm 2 PSO for FlyBS positioning & UEs' association

1: Associate UEs & allocate bandwidth by Algorithm 1 with unused BSs turned off.
2: Initialize particles $\boldsymbol{W}^l(0)$, $l = 1, \ldots, L$ based on assoc.
3: $Q^{l,local} \leftarrow Q^l \left(\boldsymbol{W}^l(0) \right)$ via (7)
4: $Q^{global} \leftarrow \text{argmax}_{l \in L} Q^{l,local}$.
5: $\boldsymbol{W}^{global} \leftarrow \text{argmax}_{l \in L} \boldsymbol{W}^l(0)$.
6: **for** $t = 1, \ldots, M_{it}$
7:    **for** $l = 1, \ldots, L$
8:       $\boldsymbol{W}^l(t) = \boldsymbol{W}^l(t-1) + \boldsymbol{D}^l(t)$ via (6).
9:       Assoc. UEs & alloc. bandwidth by Algorithm 1 with unused BSs turned off.
10:       $Q^l(t) \leftarrow Q^l \left(\boldsymbol{W}^l(t) \right)$ via (7).
11:       **if** $Q^l(t) > Q^{l,local}$
12:          $Q^{l,local} \leftarrow Q^l(t)$
13:          $\boldsymbol{W}^{l,local} \leftarrow \boldsymbol{W}^l(t)$
14:       **end if**
15:       **if** $Q^{l,local} > Q^{global}$
16:          $Q^{global} \leftarrow Q^{l,local}$
17:          $\boldsymbol{W}^{global} \leftarrow \boldsymbol{W}^{l,local}$
18:       **end if**
19:    **end for**
20: **end for**

---

cost function [29]. In our algorithm, the RWS is implemented by choosing the parent j via $\{j \in \mathbb{Z}, j \leq L | \sum_{l=1}^{l=j} F_l \geq \omega\}$ (i.e., by selecting a parent with fitness F_l equal to or larger than ω). The ω is selected randomly with uniform distribution $U(0, 1)$, and F_l is determined from the fitness function defined as:

$$F_l = \frac{e^{-\frac{\overline{Q^l}}{max\{Q^l\}}}}{\sum_{l \in L} e^{-\frac{\overline{Q^l}}{max\{\overline{Q^l}\}}}}$$
(8)

where $\overline{Q^l}$ is the normalized cost of the l-th individual calculated as $\overline{Q^l} = \frac{Q^l}{\sum_{l \in L} Q^l}$. Note that for the GA, we use the same cost function as expressed for the PSO in (7).

The number of offspring (new possible positions of the FlyBS) generated by the GA in each iteration is defined as $\lfloor L p_c \rfloor$, where p_c is the crossover ratio representing a percentage of the whole population selected as the parents. The positions of the FlyBSs belonging to the selected parents are combined via an arithmetic recombination. This means the generated positions of the FlyBSs are influenced by a recombination parameter α denoting portions of the positions, which are taken from each of the selected parents. The parameter α is selected randomly from the uniform distribution $U(0, 1)$, following an arithmetical crossover [29], where each offspring inherits a part of each parent's position.

The principle of crossover operation is illustrated in Figure 2a, where two new offspring l_1 and l_2 (i.e., new possible positions of the FlyBS) are generated from the selected parents j_1 and j_2 (i.e., positions of the FlyBS in the past). The crossover operation takes positions of the parents $\boldsymbol{v}_k^{j_1}$ and $\boldsymbol{v}_k^{j_2}$
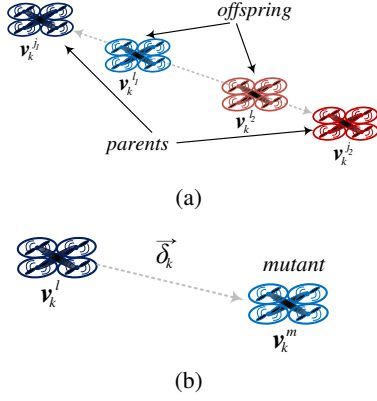
FIGURE 2: Update of the FlyBS position by the proposed algorithm based on genetic algorithm via: a) crossover and b) mutation.

and modifies them as follows:

$$\boldsymbol{v}_k^{l_1} = \left(\alpha \boldsymbol{v}_k^{j_1} + (1 - \alpha) \boldsymbol{v}_k^{j_2} \right) \tag{9}$$

$$\boldsymbol{v}_k^{l_2} = \left(\alpha \boldsymbol{v}_k^{j_2} + (1 - \alpha) \boldsymbol{v}_k^{j_1} \right) \tag{10}$$

To preserve a diversity in the population, a mutation is exploited besides the crossover operation. The mutation corresponds to the process during which the position of the FlyBS ($\boldsymbol{v}_k^l$) is modified by a vector $\vec{\delta_k}$ as follows:

$$\boldsymbol{v}_k^m = \boldsymbol{v}_k^l + \vec{\delta_k}, \forall k \in \boldsymbol{K}^F \tag{11}$$

The value of $\vec{\delta_k}$ is limited by the FlyBSs' deployment area (i.e., simulation area). Note that a direction of the vector $\vec{\delta_k}$ is selected randomly. In our case, the mutations are applied on randomly selected individuals with a probability p_m (i.e., $\lfloor L (L + Lp_c) p_m \rfloor$ mutants are generated in each iteration). The principle of mutation is depicted in Figure 2b, where the l-th FlyBS with the position $\boldsymbol{v}_k^l$ generates a mutant m with a new position corresponding to $\boldsymbol{v}_k^m$.

After the crossovers and mutations, whole population (i.e., parents, offspring, and mutants) is evaluated via the fitness function (8) to select proper individuals (i.e., sets of positions of the FlyBSs) for the new iteration.

The proposed GA solution exploiting the defined operations (crossover, mutation, etc.) is described in Algorithm 3. The algorithm is initialized by updated association with unused BSs turned off by Algorithm 1 (line 1). The population $\boldsymbol{G}$ is then generated based on the association (line 2) with the cost of each individual determined according to (7) (line 3). The crossovers and mutations are applied to the positions of FlyBSs in the $\boldsymbol{G}$ (lines 6 and 7) to generate new possible solutions (i.e., sets of positions of the FlyBSs) within the constrained area of search for the optimum. Due to the updated positions of the FlyBSs, the UEs' association and the bandwidth allocation are updated as well. The UEs are associated by considering only the BSs that are turned on (line 8). The cost of the updated population is calculated via (7) (line 9). Based on the fitness function (8), the fittest

individuals are selected for the next iteration (line 11). If the population is not diverse (i.e., the cost of all individuals is the same), the mutation percentage p_m is increased to $p_{mutate,high}$ to avoid premature convergence to a non-optimum solution; otherwise, p_m is kept at p_{mutate} (line 12). Then, the individual with the highest cost Q^l (suitability) is selected as the most suitable solution (line 13).

Algorithm 3 GA for FlyBS positioning & UEs' association

1: Assoc. UEs & allocate bandwidth by Algorithm 1 with unused BSs turned off
2: Initialize population $\boldsymbol{G}$ based on association
3: Calculate cost Q^l of $\boldsymbol{g}^l, \forall l \in L$ via (7)
4: **for** $t = 1, \ldots, M_{it}$
5:      **for** $l = 1, \ldots, L$
6:          Apply crossovers via (9) and (10).
7:          Apply mutations via (11).
8:          Assoc. UEs & alloc. bandwidth by Algorithm 1 with unused BSs turned off
9:          Calculate cost Q^l of $\boldsymbol{g}^l, \forall l \in L$ via (7)
10:      **end for**
11:      Select the fittest individuals to next iteration via (8)
12:      Check population diversity and adjust p_{mutate}.
13:      $\boldsymbol{g}^* \leftarrow \mathrm{argmax}_{l \in L} \, cost \left(\boldsymbol{g}^l \right)$
14: **end for**

D. PRACTICAL IMPLEMENTATION ASPECTS

In this subsection, we discuss aspects related to implementation of the UEs' association and the positioning of the FlyBSs. First, we focus on the mobile network entities where the algorithms for the joint positioning and association can be deployed and run. The most straightforward option is to implement the algorithm directly at the FlyBSs. On one hand, this option leads to a low latency of determining FlyBSs' positions and UEs' association, since the FlyBSs just exchange control information among themselves, and there is no need to communicate with the core network. On the other hand, this solution also drains batteries of the FlyBSs; thus, the operational time of the FlyBSs is reduced. Although the common UAVs, such as quad- or hexa-copters, can fly several hours if they are powered with hydrogen cells, any additional energy consumption is undesirable [2]. Thus, running the proposed algorithm directly at the FlyBSs is limited only to the scenarios where a short operation time of the FlyBSs is not a problem.

The second option is to run the algorithm in a fixed infrastructure, such as common SBSs, core network, or a base band unit (BBU) if the Cloud-Radio Access Network (C-RAN) is deployed [2]. In this case, the energy required to run the proposed algorithm for the positioning and association is not that critical. However, the latency (especially if the algorithm is run in the BBU connected through a non-ideal fronthaul [30]) is the main concern here and can result in incorrect positioning of the FlyBSs. As a consequence, this option is preferable if a higher latency does not degrade the

performance of the proposed algorithm, such as for slow-moving UEs (pedestrians), where the delay on the order of tens of milliseconds plays no role due to the slow movement of the UEs.

The proposed algorithms require information about the UEs' positions (these are assumed to be known in, e.g., [8] or [10]), their required data rates, and environment for estimation of the propagation losses (such as [31] or [32]) to determine the SINR for a given FlyBS's positions (as assumed, e.g., in [23] and outlined in [8]). The required data rate is known to the network, as this information is required for scheduling. The UEs' positions represent overhead on the order of tens of bytes, which is negligible.

IV. SIMULATION SCENARIO AND PERFORMANCE EVALUATION

Performance of the proposed solution is analyzed and compared with a competitive solution by simulations conducted in MATLAB.

A. SIMULATION SCENARIO

We assume a scenario in which the deployment of the FlyBSs is meaningful (i.e., the UEs benefit from increased data rate satisfaction). Thus, four small-cell SBSs (i.e., $K^S = 4$), with transmission power of 15 dBm, are deployed at positions [400 400, 400 1200, 1200 400, 1200 1200] in a simulation area of 1600 m x 1600 m, as shown in Figure 3. Moreover, up to twenty FlyBSs (i.e., $K^F = 20$), with the same transmission power as the SBSs, are deployed in the same area. The deployment of both the FlyBSs and the SBSs emulates a realistic case in which the FlyBSs cooperate with existing infrastructure, and interference among the FlyBSs and the SBSs plays an important role in the association and positioning. Thus, we also assume that all BSs transmit on the same frequency (i.e., each BS interferes with other BSs). A signal propagation for the SBSs is modeled according to [33] with path loss model $PL = 128.1 + 37.6 log_{10} d$, where d is a distance between the UE and the SBS. For the FlyBSs, we select a commonly used path loss model from [31], with Suburban environment parameters from [34]. A connectivity of the FlyBSs to the core network of the operator is assumed to be of a sufficient capacity to transfer all the UEs' data transmitted over the access link (from the UE to the FlyBS) as expected (e.g., in [2] and [35]). The major parameters of the simulations are summarized in Table 1.

B. PERFORMANCE EVALUATION

In this section, we provide a performance evaluation of the proposed solutions. The performance of the proposed algorithms based on GA and PSO is compared with a commonly exploited k-means algorithm (see, e.g., [28]) extended with the bandwidth allocation according to our proposed Algorithm 1 for a fair comparison. To the best of our knowledge, there is no other algorithm for comparison that solves joint association and positioning and targets maximization of the UEs' satisfaction. As exploiting the k-means requires that we
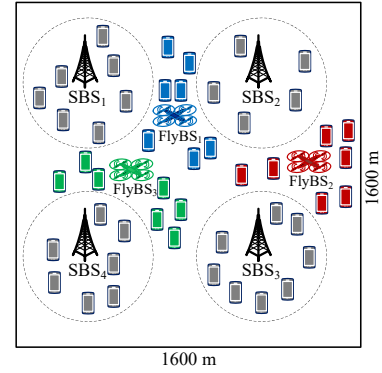

FIGURE 3: Simulation area with FlyBSs and SBSs and associated UEs (association to individual FlyBSs is indicated by colors).

know the number of FlyBSs to be deployed, we investigate the k-means with the same number of FlyBSs as the number of FlyBSs required by our proposed algorithm based on the PSO. This allows a fair comparison of the k-means and the PSO in terms of the UE's satisfaction, which is our major objective.

In Figure 4, we show the ratio of the satisfied UEs to the achieved throughput (i.e., the UEs for which $c_{n,k} \geq c_n^{min}$) for c_n^{min} set to the same value for all the UEs and ranging between 1 and 20 Mbit/s. For all compared algorithms, the ratio of satisfied UEs is decreasing with increase of both c_n^{min} and the number of UEs. The decrease in satisfaction is

TABLE 1: Simulation parameters.

| Parameter	Value
Simulation area	1600m x 1600m
Carrier frequency	2 GHz
Number of SBSs/Maximal number of FlyBSs	4/20
Tx power of SBS/FlyBS	15/15 dBm
Bandwidth of SBS/FlyBS	20/20 MHz
SBS/FlyBS/UE height	20/20/1.5 m
Maximal number of iterations GA/PSO/k-means	100/100/100
Population size GA/Number of particles PSO	100/100
GA - $p_c/p_{mutate}/p_{mutate,high}$	0.8/0.3/0.8
PSO - $\phi/\phi_1/\phi_2/c_g/c_p$	4.1/2.05/2.05/1.5/1.5
Number of simulation drops	1000 drops
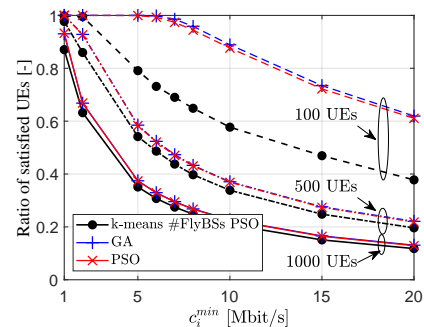
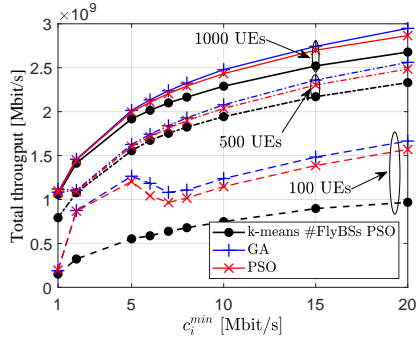FIGURE 4: Ratio of the UEs satisfied with experienced throughput.

FIGURE 5: Total throughput of all UEs in the network.

FIGURE 6: Number of deployed flying base stations in order to reach the UEs' satisfaction presented in Figure 4.

because, while the UEs require higher data rates (i.e., higher c_n^{min}), the BSs still have a limited amount of bandwidth that can be allocated to the UEs. The gain of the proposed algorithms based on GA and PSO compared to the k-means and 100 UEs is up to 30 % and 31%, respectively. Although the gain decreases with more users in the area, the gain of the proposed algorithms is above 6% for almost all values of c_n^{min}, even for 1000 UEs. The improvement in the UEs' satisfaction is achieved by the positioning of the FlyBSs, association of the UEs, and turning off the BSs that are not necessary. All these aspects lead to a higher level of the received signal and/or lower interference from neighboring BSs.

Figure 5 depicts a total throughput, which is defined as the sum of data rates $c_{n,k}$ over all UEs. It is demonstrated that the total throughput is increased by the proposed algorithms (GA and PSO) when compared to the k-means. The gain typically ranges between 19% and 47%. Again, the highest gain is achieved for a lower number of UEs. For example, for 100 UEs, both proposed algorithms lead to an improvement in the total throughput by approximately 30% for $c_n^{min}=$ 1 Mbit/s with respect to the k-means. The highest gain is observed for $c_n^{min} = 2$ Mbit/s and 100 UEs; in this case, the proposed algorithms based on GA and PSO outperform the k-means by 128% and 126%, respectively. This notable gain is because the k-means fails to handle small groups of the UEs requiring relatively high data rates. In contrast, both GA- and PSO-based algorithms converge to a suitable deployment and association that avoids redundant interference.

Figure 5 also shows an interesting phenomenon, as the total throughput for 500 UEs is higher than that for 1000 UEs if $c_n^{min} = 1$ Mbit/s. This is due to the definition of the cost function in (7), and the fact that the primary objective is to maximize the UEs' satisfaction while the throughput maximization is only a secondary objective (see (7)). Consequently, if all UEs' requirements are satisfied with the experienced throughput (i.e., the case with 500 UEs and c_n^{min} = 1 Mbit/s, as shown in Figure 4), both the GA and the PSO start maximizing the total throughput as well. In contrast, if some of the UEs remain unsatisfied (i.e., the case with 1000 UEs deployed in the area and $c_n^{min} = 1$ Mbit/s), the GA and the PSO aim to maximize the UEs' satisfaction while the
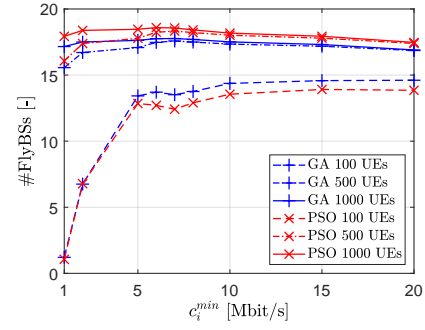
throughput maximization does not take place. Consequently, the total throughput for 1000 UEs is slightly lower than the total throughput achieved for 500 UEs if $c_n^{min} = 1$ Mbit/s.

The same behavior can be seen also for 100 UEs, where the throughput gradually increases as long as $c_n^{min} \leq 5$ Mbit/s since all 100 UEs are always satisfied (i.e., the proposed algorithms are able to further maximize the throughput according to the secondary objective). However, for c_n^{min} higher than 5 Mbit/s, the total throughput starts decreasing (see the throughput for c_n^{min} between 5 and 7 Mbit/s). For these values of c_n^{min}, the case when all UEs are satisfied (i.e., when throughput maximization takes place) occurs for some simulation drops while, in other drops, not all UEs are satisfied (i.e., the secondary objective of throughput maximization does not take place). The cases when all UEs are satisfied and when some UEs are not satisfied are represented by different slopes of the total throughput over c_n^{min}. Combination of both cases for c_n^{min} between 5 and 7 Mbit/s results in a decrease in the total throughput. Nonetheless, with a further increase in c_n^{min} above 7 Mbit/s, the total throughput starts increasing again, following the slope corresponding to the second case (some UEs are not satisfied), with a slower increase in the total throughput comparing to $c_n^{min} \leq 5$ Mbit/s. The slope of the second case ($c_n^{min} > 7$ Mbit/s) is lower because the secondary objective takes place in no (or almost no) drops. Note that the first non-satisfied UEs are those with the worst channel quality. Thus, the total throughput still increases even if the secondary objective is not considered. This is due to the allocation of bandwidth to the UEs with a higher channel quality.

The number of active FlyBSs required to maximize the UEs' satisfaction is presented in Figure 6. The number of FlyBSs for the k-means is not shown, as the k-means cannot change the number of active FlyBSs, and we set it to the number of active FlyBSs required by the PSO, as explained earlier. From Figure 6, we can see that the number of active FlyBSs with required throughput increases for low c_n^{min}, but for c_n^{min} above 5 Mbit/s, the number of active FlyBSs starts slowly decreasing for 500 and 1000 UEs. The decrease in the number of active FlyBSs for a larger c_n^{min} is due to the fact that the additional FlyBSs increase interference more than
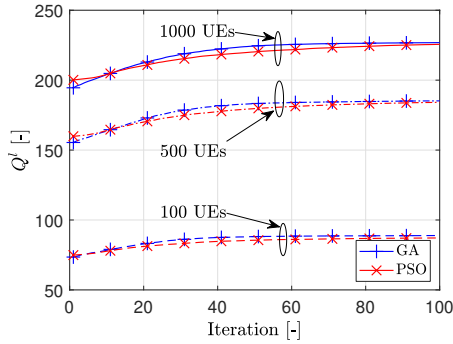
FIGURE 7: Evolution of cost over iterations of the GA and PSO for c_n^{min} = 10 Mbit/s.

the amount the UEs can gain from the improved level of the useful signal provided by the serving BS.

The proposed algorithms based on GA and PSO find the solution iteratively. An evolution of the cost function Q^l with iterations for $c_n^{min} = 10$ Mbit/s is shown in Figure 7. The figure depicts the number of UEs satisfied with their data rates in each iteration (following (7)). The value of the cost function (i.e., the number of satisfied UEs) iteratively improves as the algorithms based on both the GA and the PSO find better positions of the FlyBSs. It is shown that the positioning based on the GA provides slightly higher values of the cost function in comparison to the PSO (difference is on the order of a few percent. However, the cost function converges almost to its maximum in roughly 30 or 40 iterations for both algorithms. Then, the cost function remains almost constant. Such a low number of iterations required for the convergence makes the proposed solutions promising for real networks.

The performance analysis presented in the previous figures shows that the GA slightly improves the UEs' satisfaction with respect to the PSO (by up to 2%, see Figure 4). At the same time, the GA increases the total throughput by 1%~10% compared to the PSO (see Figure 5). In addition, the GA converges to these gains while requiring approximately one FlyBS less than the PSO (see Figure 6), so the operational cost is slightly reduced by the GA as well. However, this gain is at the cost of a higher time complexity of the algorithm based on the GA. The GA is of a higher time complexity than the PSO because of the nature of the base GA and PSO algorithms (see, for example, [36]). We express the time complexity as the time required to complete one iteration of the algorithm (i.e., one update of the FlyBSs' positions for each population/particle). The iteration takes 0.22 and 0.13 seconds, on average, for the GA and the PSO, respectively. Note that the times of each iteration are obtained at a desktop PC with Intel i7-7700K@4.2 GHz CPU and 32 GBs of RAM.

V. CONCLUSION

In this paper, we have proposed an algorithm for the joint positioning of the FlyBSs and association of the UEs to maximize the number of UEs satisfied with the experienced data rates. The developed algorithm is presented in two

variants: one based on the GA and one on the PSO. We show that both approaches improve the UEs' satisfaction compared to the commonly used k-means by up to roughly 30%. Also, a gain in the total throughput of all UEs is observed for both proposed algorithms. The gain typically varies between 19% to 47%, but reaches its maximum of more than 100% for scenarios with a lower number of UEs and medium to high data rates. We also show that the GA slightly increases the UEs' satisfaction and the total throughput while reducing the number of required FlyBSs compared to the PSO. This improvement is, however, at the cost of a higher time complexity.

Future research should be focused on development of solutions that allow application of the proposed evolutionary-based algorithms as well as those that allow high mobility and efficient mitigation of interference among all BSs [14].

REFERENCES

[1] M. Kamel, W. Hamouda, and A. Youssef, "Ultra-dense networks: A survey," IEEE Communications Surveys & Tutorials, vol. 18, no. 4, 2016.

[2] Z. Becvar, M. Vondra, P. Mach, J. Plachy, and D. Gesbert, "Performance of mobile networks with uavs: Can flying base stations substitute ultra-dense small cells?" in European Wireless 2017, 2017, pp. 1–7.

[3] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for cell association in uav-enabled cellular networks," IEEE Communications Letters, vol. 21, no. 9, pp. 2053–2056, 2017.

[4] X. Lin, V. Yajnanarayana, S. D. Muruganathan, S. Gao, H. Asplund, H.-L. Maattanen, M. Bergstrom, S. Euler, and Y.-P. E. Wang, "The sky is not the limit: Lte for unmanned aerial vehicles," IEEE Communications Magazine, vol. 56, no. 4, pp. 204–210, 2018.

[5] J. Chen, Q. Wu, Y. Xu, Y. Zhang, and Y. Yang, "Distributed demand-aware channel-slot selection for multi-uav networks: A game-theoretic learning approach," IEEE Access, vol. 6, pp. 14 799–14 811, 2018.

[6] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," IEEE Communications Magazine, vol. 54, no. 5, pp. 36–42, 2016.

[7] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," arXiv preprint arXiv:1803.00680, 2018.

[8] J. Chen and D. Gesbert, "Optimal positioning of flying relays for wireless networks: A los map approach," in IEEE International Conference on Communications (ICC), 2017.

[9] M. M. Azari, F. Rosas, K.-C. Chen, and S. Pollin, "Ultra reliable uav communication using altitude and cooperation diversity," IEEE Transactions on Communications, vol. 66, no. 1, pp. 330–344, 2018.

[10] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3d placement of an unmanned aerial vehicle base station for maximum coverage of users with different qos requirements," IEEE Wireless Commun. Letters, 2017.

[11] R. Fan, J. Cui, S. Jin, K. Yang, and J. An, "Optimal node placement and resource allocation for uav relaying network," IEEE Communications Letters, vol. 22, no. 4, pp. 808–811, 2018.

[12] Y. J. A. Zhang, L. Qian, J. Huang et al., "Monotonic optimization in communication and networking systems," Foundations and Trends® in Networking, vol. 7, no. 1, pp. 1–75, 2013.

[13] P. Yang, X. Cao, C. Yin, Z. Xiao, X. Xi, and D. Wu, "Proactive drone-cell deployment: Overload relief for a cellular network under flash crowd traffic," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 10, pp. 2877–2892, 2017.

[14] Y. Cao, N. Zhao, F. R. Yu, M. Jin, Y. Chen, J. Tang, and V. C. M. Leung, "Optimization or alignment: Secure primary transmission assisted by secondary networks," IEEE Journal on Selected Areas in Communications, vol. 36, no. 4, pp. 905–917, April 2018.

[15] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," Mobile networks and Applications, vol. 10, no. 4, 2005.

[16] H. Huang and A. V. Savkin, "An algorithm of efficient proactive placement of autonomous drones for maximum coverage in cellular networks," IEEE Wireless Communications Letters, 2018.

[17] X. Zhang and L. Duan, "Fast deployment of uav networks for optimal wireless coverage," IEEE Transactions on Mobile Computing, 2018.

[18] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in ACM Int. Conference on Embedded Networked Sensor Systems, 2003.

[19] H. Ghazzai, E. Yaacoub, M.-S. Alouini, Z. Dawy, and A. Abu-Dayya, "Optimized lte cell planning with varying spatial and temporal user densities," IEEE Trans. on Vehicular Technology, vol. 65, no. 3, 2016.

[20] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage." IEEE Communications Letters, vol. 20, no. 8, 2016.

[21] F. Lagum, I. Bor-Yaliniz, and H. Yanikomeroglu, "Strategic densification with uav-bss in cellular networks," IEEE Wireless Communications Letters, vol. 7, no. 3, pp. 384–387, 2018.

[22] T. Back, Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms. Oxford university press, 1996.

[23] W. Shi, J. Li, W. Xu, H. Zhou, N. Zhang, S. Zhang, and X. Shen, "Multiple drone-cell deployment analyses and optimization in drone assisted radio access networks," IEEE Access, vol. 6, 2018.

[24] E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu, "On the number and 3d placement of drone base stations in wireless cellular networks," in IEEE Vehicular Technology Conference (VTC-Fall), 2016.

[25] V. Roberge, M. Tarbouchi, and G. Labonté, "Fast genetic algorithm path planner for fixed-wing military uav using gpu," IEEE Transactions on Aerospace and Electronic Systems, 2018.

[26] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," Swarm intelligence, vol. 1, no. 1, pp. 33–57, 2007.

[27] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Wireless communication using unmanned aerial vehicles (uavs): Optimal transport theory for hover time optimization," IEEE Transactions on Wireless Communications, vol. 16, no. 12, pp. 8052–8066, 2017.

[28] B. Galkin, J. Kibilda, and L. A. DaSilva, "Deployment of uav-mounted access points according to spatial user locations in two-tier cellular networks," in Wireless Days (WD), 2016, 2016, pp. 1–6.

[29] M. Gen and R. Cheng, Genetic algorithms and engineering optimization. John Wiley & Sons, 2000, vol. 7.

[30] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud ran for mobile networks - a technology overview," IEEE Communications surveys & tutorials, vol. 17, no. 1, pp. 405–426, 2015.

[31] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal lap altitude for maximum coverage," IEEE Wireless Commun. Let., vol. 3, no. 6, 2014.

[32] I. Bor-Yaliniz, S. S. Szyszkowicz, and H. Yanikomeroglu, "Environment-aware drone-base-station placements in modern metropolitans," IEEE Wireless Communications Letters, vol. 7, no. 3, pp. 372–375, 2018.

[33] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects," Tech. Rep. 36.814, 2010.

[34] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-d placement of an aerial base station in next generation cellular networks," in IEEE International Conference on Communications (ICC), 2016.

[35] A. Fouda, A. S. Ibrahim, I. Guvenc, and M. Ghosh, "Uav-based in-band integrated access and backhaul for 5g communications," arXiv preprint arXiv:1807.07230, 2018.

[36] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in International conference on evolutionary programming. Springer, 1998, pp. 611–616.

JAN PLACHY received the BSc and MSc degree in telecommunication engineering from the Czech Technical University in Prague, Czech Republic in 2012 and 2014, respectively. Currently, he works towards the Ph.D. degree at the Department of Telecommunication Engineering at the same university with a topic Allocation of Communication and Computation Resources for Big data in Mobile Networks. He was on internships at CEA-Leti, France (2014), EURECOM, France (2016) and Bar-Ilan University, Israel (2017). In 2015, he has joined the 5G Mobile Research Lab funded by the Czech Technical University of Prague, focusing on key aspects and challenges related to future mobile networks and emerging wireless technologies.

He has coauthored 4 conference papers and 2 journal papers. His research interests cover optimization of mobility management and radio resource management of both communication and computing resources in future mobile networks.

ZDENEK BECVAR (M'08-SM'17) received M.Sc. and Ph.D. in Telecommunication Engineering from the Czech Technical University in Prague, Czech Republic in 2005 and 2010, respectively. Now, he is Associate Professor at the Department of Telecommunication Engineering, Czech Technical University in Prague, Czech Republic. From 2006 to 2007, he joined Sitronics R&D center in Prague focusing on speech quality in VoIP. Furthermore, he was involved in research activities of Vodafone R&D center at Czech Technical University in Prague in 2009. He was on internships at Budapest Politechnic, Hungary (2007), CEA-Leti, France (2013), and EURECOM, France (2016). From 2013 to 2017, he was a representative of the Czech Technical University in Prague in ETSI and 3GPP standardization organizations. In 2015, he founded 5Gmobile research lab at CTU in Prague focusing on research towards 5G mobile networks and beyond.

He is a member of more than 20 program committees at international conferences or workshops and he has published 4 book chapters and more than 70 conference or journal papers. He works on development of solutions for future mobile networks with special focus on optimization of mobility and radio resource management, device-to-device communication, self-optimization, power control, architecture of radio access network, and small cells.

PAVEL MACH received the M.Sc. and Ph.D. degrees in telecommunication engineering from Czech Technical University in Prague, Czech Republic, in 2006 and 2010, respectively. He is currently a postdoc researcher with the Department of Telecommunication Engineering, Czech Technical University in Prague. In 2015, he joined the 5G Mobile Research Lab, funded by the Czech Technical University, focusing on key aspects and challenges related to future mobile networks and emerging wireless technologies.

He has coauthored more than 60 papers in international journals and conferences. His research interests include radio resource management in emerging wireless technologies.

RADEK MARIK received his M.S. and Ph.D. degrees from the Czech Technical University, Prague, Czech Republic, in 1984 and 1992, respectively, both in Technical Cybernetics. For 15 years he worked mainly in the field of computer vision at the Czech Technical University in Prague and the University of Surrey, UK. Since 1997, he was with Protys s.r.o., Czech Republic, working on research contracts with Rockwell Automation in the field of software diagnostics with the focus on metaprogramming and automated design of software tests. From 2007 he worked on technology trends identification, knowledge-based software comprehension and semantically based product interoperability on research projects at CA Technologies. He joined Czech Technical University teams in 2011. He is a (co)-author of about 80 papers and has filed two patents in complex networks analysis and machine learning in signal processing.

MICHAL VONDRA is a postdoctoral researcher at the Czech Technical University in Prague, Czech Republic. He received his BSc and MSc degree in Telecommunication Engineering and Radioelectronics from the Czech Technical University in Prague, in 2008 and in 2010, respectively. In 2015, he received his Ph.D. degree in Telecommunication Engineering from the Czech Technical University in Prague. His thesis "Allocation of Resources in Network with Small Cells" was awarded Dean's Award for the Dissertation thesis. From January to June 2014, he was on internship at Performance Engineering Laboratory at University College Dublin (Ireland) where he participated in the TRAFFIC project focused on vehicular communication. From February 2016 to May 2018, he was a Postdoctoral Researcher at the Communication Systems department (RSLab), KTH Royal Institute of Technology, Stockholm (Sweden) where he collaborated with industry partners, such as Ericsson, British Telecom, and Airbus within the European project ICARO-EU funded by EIT Digital.

He has published more than 25 conference papers, journal papers or book chapters. His research interest includes mobility management in wireless networks, vehicular ad-hoc networks, intelligent transportation systems, and direct air-to-ground communication.

• • •