IET Research Journals

Research Article

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Fairness Aware Multiple Drone Base Station Deployment

Alper Akarsu[1,2], *Tolga Girici*[1]

[1] *Electrical and Electronics Engineering, TOBB University of Economics and Technology, Ankara, Turkey*
[2] *Command Control and Combat Systems, HAVELSAN, Ankara, Turkey*
* *E-mail: aakarsu, tgirici@etu.edu.tr*

Abstract: The recent advances in drone technology significantly improves the effectiveness of applications such as border surveillance, disaster management, seismic surveying and precision agriculture. The use of drones as base stations to improve communication in the next generation wireless networks is another attractive application. However, the deployment of drone base stations (DBSs) is not an easy task and requires a carefully designed strategy. Fairness is one of the most important metrics of tactical communications or a disaster-affected network and must be considered for the efficient deployment of DBSs. In this study, fairness-aware multiple DBS deployment algorithm is proposed. As the proposed algorithm uses particle swarm optimization (PSO) that requires significant processing power, simpler algorithms with faster execution times are also proposed and the results are compared. The simulations are performed to evaluate the performance of the algorithms in two different network scenarios. The simulation results show that the proposed PSO based method finds the 3-Dimensional (3D) locations of DBSs, achieving the best fairness performance with minimum number of DBSs for deployment. However, it is shown that the proposed suboptimal algorithm performs very close to the PSO-based solution and requires significantly less processing time.

1 Introduction

The applications of drones are significantly increased in the last decade. The use of drones open a new dimension and change the conventional way of thinking for solving engineering problems and designing critical applications. The drone based applications span from border surveillance [1], disaster management [2], seismic surveying [3] to precision agriculture [4]. Once considered as a futuristic concept, drone based order delivery now became possible due to the efforts of research community and industry. Amazon claims that PrimeAir, which is a drone-based order delivery system, will increase the overall safety and efficiency of the transportation system [5]. Amazon's project shows how spread the drones will be used in the future.

Recently, the use of drones as a base station to improve communication in the next generation wireless networks has attracted many researchers. There are growing number of papers related to the placement of drones for improving wireless networks. The air-ground channel model for low altitude platforms such as drones and quadcopters was investigated in [6]. The ray tracing simulation results were used to analyze the behaviour of the channel and experimental data was fitted to explicit mathematical formulas. The authors in [7], presented a closed form approximation for the probability of Line of Sight (LoS) between the aerial platform and the user. The optimal altitude of a DBS was also derived in order to maximize the cell coverage. In [8], both the optimal altitude of a DBS and minimum transmission power to maximize coverage is derived. The authors also investigated the use of two interfering DBSs and found the optimal distance between DBSs to maximize the coverage of the rectangle shaped cell. For the first time, single 3D DBS placement problem was formulated in [9] to maximize the number of covered users in a cell. In [10], a single DBS placement to serve users having different rate requirements was investigated taking into account the limited-capacity wireless backhaul link. The work aimed to maximize the number of served users and sum rate of users. Authors in [11] proposed an optimal placement algorithm based on a circle packing for a DBS deployment that maximizes the number of covered users with the minimum transmission power. In [12], the deployment of multiple DBSs, each using a directional antenna was considered using circle packing theory. However, it was assumed that all DBSs have the same altitude. The work in [13] also assumed the same altitude for all DBSs, and determined the minimum number of required DBSs and their 2D locations in the horizontal plane to cover all users. However, the authors did not consider the inter-cell interference. Authors in [14], proposed a particle swarm optimization based algorithm to find the number of required DBS and their positions under the coverage and capacity constraints.

The works presented so far consider 3D placement of a DBS or more than one DBSs to maximize served users and capacity. Some of these works do not take into account the inter-cell (i.e. inter-DBS) interference which is not a practical assumption. In studies related to the placement of multiple DBSs, some studies assume either fixed altitudes of DBSs and determine 2D placements of them, or place the DBSs at the target area and determine the altitudes of DBSs. None of these studies investigated the 3D placement of multiple DBSs considering both capacity and fairness amongst users.

We propose a fairness aware multiple DBS deployment scheme, which maximizes the sum of logarithms of achievable rates of users. Maximizing the sum of logarithmic rates provides proportional fairness, which is a good trade-off between total throughput and max-min fairness. Maximizing total throughput results in total neglect of some of the users. On the other hand aiming at max-min fairness sacrifices too much from some users in order to improve the worst user. The sum log rate approach is considered, for example, in [15] . The authors investigated the fairness performance of users in a cellular system with a full-duplex base station using log sum rate and direct sum rate maximizing algorithms. In [16], the sum log rate approach is used for the fair allocation of resources in an interference-free a heterogeneous network. In order to verify the fairness performance of our sum log rate based deployment algorithm, the fairness metrics are calculated and compared with the other algorithms which have different complexities. As the problem is highly nonlinear and nonconvex, we apply a metaheuristic solution based on the concept of particle swarm optimization (PSO). We also propose a near-optimal method that has much less run-time and PSO. We perform several numerical evaluations in order to compare these two methods along with some other benchmarks.

The paper is organized as follows. Section 2 presents the system and channel model. Section 3 includes the problem formulation,

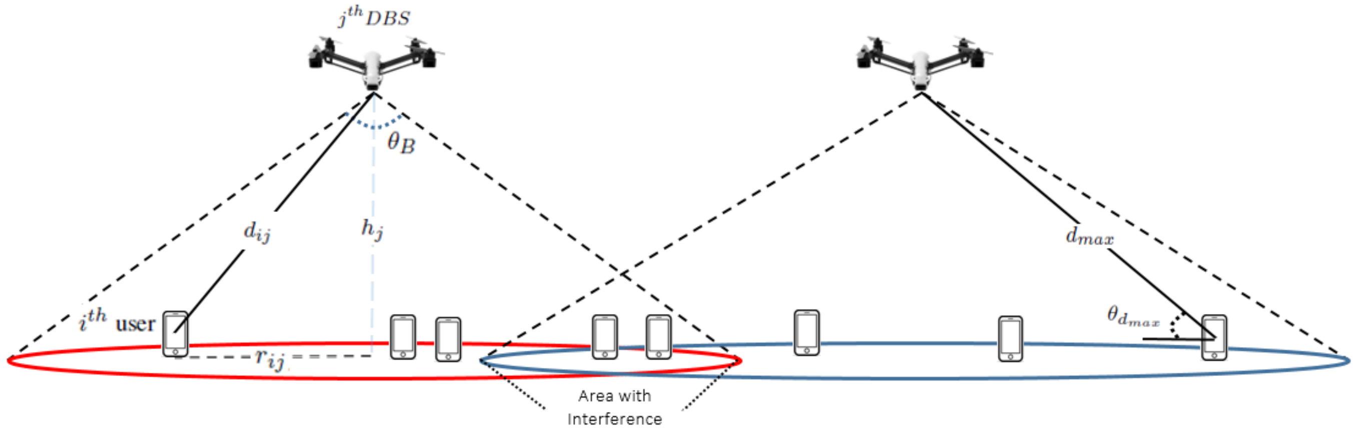


Fig. 1: System model for two drone BSs (DBSs). Coverage areas of the two DBSs overlap, resulting in interference.

proposed and benchmark solutions. In Section 4 we present the simulation results. Finally Section 5 concludes the paper.

2 System Model

We consider the downlink transmission from DBSs to the cellular users. We assume two different user distribution scenarios. In the first scenario, cellular users are uniformly distributed over the cellular area, which has a maximum radius of R_{max}, whereas in the second scenario users are non-uniformly distributed. In the latter case, parent points are first uniformly generated in the cellular area. Then children (i.e. users) are uniformly distributed around the parent points to achieve heterogeneity in spatial user distribution.

Users are denoted by the set U and their locations are given by (x_i, y_i), where $i \in U$. It is assumed that the users are stationary and each user is served by a DBS from the set D. We determine the locations of DBSs which are denoted by (x_j, y_j, h_j), where $j \in D$. The minimum and maximum altitude of a DBS is limited to comply drone regulations. We assume transmission power of all DBSs are fixed. Another assumption is that the backhaul link has sufficient bandwidth and operates on a different frequency band, therefore avoiding interference to the user terminals. For multiple access, Time Division Multiple Access (TDMA) is adopted and each DBS assigns equal fraction of time slots to each user it serves. We also assume that each user is connected to the DBS with highest Signal to Noise Ratio (SNR). Figure 1 illustrates the system model for a special case of two DBSs.

In [6], Air-to-Ground path loss model was analyzed using the results of a ray tracing simulation software. The simulation samples are classified in three groups. In the first group, called LoS or Near LoS, the rays travel freely between transmitter and receiver. In the second group, there is no direct ray between transmitter and receiver (NLoS) but receiver still receives strong rays via reflection and refraction. In the last group, there is NLoS between transmitter and receiver but in this case, receiver suffers from deep fading. The authors in [6] ignore the last group because its samples constitutes less than %3 of the total samples. The samples in the first and second groups are used to provide statistically generic Air-to-Ground path loss model. Authors' work in [7], refines the path loss model and provides the mathematical formulas, which are easy to work with. The probability of LoS calculation is simplified by the approximation presented as:

$$P_{LoS}(h_j, r_{ij}) = \frac{1}{1 + \alpha \exp(-\beta(\frac{180}{\pi} arctan(\frac{h_j}{r_{ij}}) - \alpha))} \quad (1)$$

where r_{ij} is the distance between the i^{th} user and the j^{th} DBS's projection on to the horizontal plane and equals to $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2)}$, α and β are environment dependent constants.

The path loss models for LoS and NLoS links are formulated as [11]:

$$PL^{ij}_{LoS} = 10 \log \left(\frac{4\pi d_{ij} f_c}{c} \right)^{\gamma} + \eta_{LoS} \quad (2)$$

$$PL^{ij}_{NLoS} = 10 \log \left(\frac{4\pi d_{ij} f_c}{c} \right)^{\gamma} + \eta_{NLoS} \quad (3)$$

where d_{ij} is the distance between the i^{th} user and j^{th} DBS, γ is the path loss exponent, f_c is the operating frequency (1/s), c is the speed of light, η_{LoS} and $\eta NLoS$ are the mean additional losses (in dB, due to shadowing) for the LoS and NLoS connections, respectively.

Terrain profile (LoS or nLoS) is probabilistically dependent on the distance, which also depends on the DBS location. Therefore each candidate DBS 3D location set would randomly result in a different profile. In order to optimize the DBS coordinates we adopt the mean path loss model given in [11]:

$$\overline{PL_{ij}} = PL^{ij}_{LoS} P_{LoS}(h_j, r_{ij}) + PL^{ij}_{NLoS}(1 - P_{LoS}(h_j, r_{ij})) \quad (4)$$

We assume that each DBS uses a fixed transmit power of P_T. Let G_{ij} be the transmit antenna gain of DBS j in the direction of user i. We assume an additive white gaussian noise (AWGN) channel with power spectral density N_o.

As seen in the system model the channel model that we used and the interference from nearby DBSs depend only on the locations of the users and DBSs. The algorithms that we will propose in the subsequent section requires the positions of the users for implementation. Almost all mobile phones and tactical radios today have an in-built GPS module. Positions of users can be obtained using GPS and fed back to the DBSs or a central controller, to be used in the proposed algorithms. Even if GPS service is not available, DBSs can send pilot signals and mobile nodes can compute their relative locations using triangulation techniques.

3 Deployment of Multiple DBSs

Drone Base Stations (DBSs) are suitable for a number of applications. For instance, terrestrial communication infrastructure may be damaged so that all communication has to be provided by the DBSs. After a disaster, each user requires to communicate through text, voice or video. Considering this fact, the fairness amongst users become critically important. Another possible application is tactical networks where each mobile node should be connected to the system and receive a fair rate. In order to meet these capacity and fairness requirements, we propose a multiple DBSs deployment algorithm

IET Research Journals, pp. 1–7

based on the sum of logarithms of the data rate of each user. As mentioned in the Introduction, maximizing log-sum rate provides proportional fairness. In this section we formulate the problem of 3D positioning of a number of DBSs in order to maximize proportional fairness.

Besides, as the proposed algorithm depends on particle swarm optimization (PSO) which requires significant processing power, we investigate the fairness performance of more time efficient multiple DBSs placement algorithms, namely, K-means clustering, K-means clustering with optimized altitude, X-means clustering and basic geometric approach.

For a given set of DBS coordinates, each user has to select a DBS. This is called user-DBS association. We assume that the i^{th} user selects the DBS with maximum received power. Let α_i be the DBS selected by the i^{th} user.

$$\alpha_i = \arg\max_j R_{ij}(x_j, y_j, h_j), \tag{5}$$

where $R_{ij}(x_j, y_j, h_j)$ is the received signal power from the j^{th} DBS at the i^{th} user terminal and is calculated as [12]

$$R_{ij}(x_j, y_j, h_j) = 10^{\frac{P_T + G_{ij} - \overline{PL_{ij}}}{10}}, \tag{6}$$

where W is the channel bandwidth, P_T is the transmission power (in dB) of a DBS, G_{ij} is the gain of the DBS antenna (in dB) and is approximated by [12]

$$G_{ij} = \begin{cases} \frac{29000}{\theta_B^2}, & \text{if } r_{ij} \leq\ h_j \tan\frac{\theta_B}{2} \\ 0, & \text{if } r_{ij} > h_j \tan\frac{\theta_B}{2} \end{cases}, \tag{7}$$

where θ_B is the half power beamwidth of the DBS's antenna. Capacity of a DBS is equally shared amongst the users in the time domain. Let $U_j(\mathbf{x}, \mathbf{y}, \mathbf{h})$, be the total number of users connected to DBS j.

$$U_j(\mathbf{x}, \mathbf{y}, \mathbf{h}) = \sum_{i \in U} I(\alpha_i = j), \quad \forall j \in U, \tag{8}$$

where $\mathbf{x}, \mathbf{y}, \mathbf{h} \in \mathbb{R}^{|D|}$ are the locations of DBSs in the 3D space and $I(\alpha_i = j) \in \{0, 1\}$ defined as

$$I(\alpha_i = j) = \begin{cases} 1, & \text{if} \quad \alpha_i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Finally, the fairness aware multiple DBS deployment problem can be formulated as

$$\begin{aligned} &\max_{\mathbf{x},\mathbf{y},\mathbf{h}} \sum_{i \in U} \log\left(\frac{1}{U_{\alpha_i}} W \log_2\left(1 + \frac{R_{i\alpha_i}(x_{\alpha_i}, y_{\alpha_i}, h_{\alpha_i})}{N_o W + \sum\limits_{j \neq \alpha_i} R_{ij}(x_j, y_j, h_j)}\right)\right), \\ &\text{subject to} \\ &\sqrt{x_j^2 + y_j^2} \leq R_{max} \qquad \forall j \in D, \\ &h_{min} \leq h_j \leq h_{max} \qquad \forall j \in D. \end{aligned} \tag{10}$$

where $N_0 W$ is the noise power.

Another alternative (benchmark) is locating the DBSs in a way to maximize the total throughput. The problem of multiple DBS deployment for direct sum rate maximization can be written as,

$$\begin{aligned} &\max_{\mathbf{x},\mathbf{y},\mathbf{h}} \sum_{i \in U} \left(\frac{1}{U_{\alpha_i}} W \log_2\left(1 + \frac{R_{i\alpha_i}(x_{\alpha_i}, y_{\alpha_i}, h_{\alpha_i})}{N_0 W + \sum\limits_{j \neq \alpha_i} R_{ij}(x_j, y_j, h_j)}\right)\right) \\ &\text{subject to the same constraints given in (10).} \end{aligned} \tag{11}$$

In problems defined in (10) and (11) the rate expressions involve interference, which depends on the optimization parameters (**x**,**y**,**h**). Moreover User-DBS association is also dependent on the optimization parameters, which makes the problems highly nonlinear. Therefore it is impossible to solve these problems using standard convex programming techniques. Hence we apply a meta heuristic algorithm, called Particle Swarm Optimization (PSO), which was introduced by J. Kennedy and R. Eberhart in 1995 [17]. PSO algorithm is inspired from the swarm behavior of animals such as birds or fishes in nature. The algorithm generates a population of random solutions called *particles* and improves the solution in each iteration by moving particles in the hyperspace. The positions of particles are updated in each iteration by considering the position of a particle achieving the best solution found in the swarm, called *global best*, and the position of the particle achieving the best solution for that particle, called *local best*. The algorithm keeps track of the global best value and its position and the local best values of the particles and their locations. In our case, the stopping criteria is chosen as the convergence of particles to a point. The details of the PSO algorithm we apply to the problems presented in (10) and (11), is provided in Algorithm 1. We call these approaches as LogPSO and LinPSO, respectively.

In this algorithm first N_{par} candidate solutions and velocity of each particle are randomly generated in Line 1. Then the algorithm iterates until convergence (Lines 2-22). Convergence means the positions of the particles being sufficiently close to each other. At each iteration the algorithm calculates the fitness values of each particle (denoted by $\mathbf{F}(i)$), which is the log-sum rate of users corresponding to that candidate solution in Lines 3-5. Then, the globally best solution is updated in Lines 6-11. Local best position of each particle is updated in Lines 12-17. Finally based on these values, the positions of each particle are updated in Line 18-21. Here, c_1 and c_2 are the algorithm parameters. In the literature, it is common to take $c_1 = 0.72$ and $c_2 = 2.3$.

A high number of particles improves the chance of finding the globally optimal solution. Although the implementation of the PSO is easy, it requires significant amount of processing power if the number of particles and the problem dimensions are large. Therefore, we also propose less complex algorithms to solve multiple DBS deployment problem and compare their performance results.

Algorithm 1 LogPSO and LinPSO

1: Generate the random matrix of candidate solutions $\mathbf{P} \in \mathbb{R}^{N_{par} \times 3|D|}$ and calculate $\mathbf{V}$ by: $\mathbf{V} = \mathbf{P} \times 0.3 \times (rand(npar, 3 \times |D|) - 1)$
2: **while** not converge **do**
3:   **for** $i = 1$ *to* N_{par} **do**
4:     $\mathbf{F}(i)$ = calculate (10) given the DBS positions $\mathbf{P}(i, :)$ (or calculate (11) for LinPSO)
5:   **end for**
6:   $newgbest = max(\mathbf{F})$;
7:   $index = argmax(\mathbf{F})$;
8:   **if** $newgbest > gbest$ **then**
9:     $gbest = newgbest$
10:     **gbestP** $= \mathbf{P}(index, :)$
11:   **end if**
12:   **for** $i = 1$ *to* N_{par} **do**
13:     **if** $\mathbf{F}(i) >$ **lbest**(i) **then**
14:       **lbest**$(i) = \mathbf{F}(i)$
15:       **lbestP**$(i, :) = \mathbf{P}(i, :)$
16:     **end if**
17:   **end for**
18:   **for** $i = 1$ *to* N_{par} **do**
19:     $\mathbf{V}(i, :) = c_1 \times \mathbf{V}(i, :) + c_2 \times \texttt{rand}(1, 3|D|)(\mathbf{lbestP}(i, :) - \mathbf{P}(i, :)) + c_2 \times \texttt{rand}(1, 3|D|)(\mathbf{gbestP}(i, :) - \mathbf{P}(i, :))$
20:   **end for**
21:   $\mathbf{P} = \mathbf{P} + \mathbf{V}$
22: **end while**

3.1 K-means Clustering (K-Cov)

K-means is an unsupervised learning method to solve clustering problems. Given the observations and the number of clusters (e.g. $|D|$ clusters), the algorithm finds the positions of $|D|$ number of centroids which minimize the associated cost [18]. In our case, observations are user positions and centroids are the locations of $|D|$ number of DBSs in the horizontal plane. Here, the cost is the sum of the squared Euclidean distance between DBSs and their associated users. The problem is formulated by,

$$\min_{j \in D} \sum_{i \in U} \|u_i - \mu_j\|^2 \tag{12}$$

where $u_i \in \mathbb{R}^2$ is the positions of i^{th} user and $\mu_j \in \mathbb{R}^2$ is the positions of j^{th} DBS. Stuart Lloyd proposed an efficient iterative algorithm in 1982 [19]. This algorithm first randomly determines the positions of the centroid and then connects each user to the closest centroid. Then the centroid positions are updated by takin the mean of the user locations connected to each centroid. This operation is repeated until there is no significant change in the centroid locations. Using Lloyds algorithm, we find the location of each DBS in the horizontal plane and its associated users. We now define a very simple approach in order to determine the altitude of each DBS, as follows,

$$H = \max\left(h_{min}, \frac{d_{max}}{\tan(\frac{\theta_B}{2})}\right), \tag{13}$$

where d_{max} is the horizontal projection of the distance between the DBS and the most distant user served by it. We follow this approach in order to lower the DBS as much as possible and still cover all its associated users. We call this simple approach as *K-Cov*.

3.2 K-means Clustering with Optimized Altitude (K-APSO)

In this algorithm, first we benefit from the time efficient implementation of K-means clustering in order to obtain the locations of DBSs in the horizontal plane. Secondly, we benefit from the performance of the PSO algorithm to only find the altitudes of DBSs. Compared to the Algorithm 1, the time consuming search of particles is reduced as the particle search dimension is $|D|$ instead of $3 \times |D|$ dimension. Hereafter, we call this algorithm K-APSO. The details of the K-APSO algorithm is presented in Algorithm 2.

Algorithm 2 K-APSO

Apply Lloyd's algorithm to find x_i and y_i for $i \in D$
$[\mathbf{x}, \mathbf{y}] = Lloyd(\mathbf{locNx}, \mathbf{locNy})$
Generate the matrix of random candidate solutions (for altitudes) $\mathbf{A} \in \mathbb{R}^{N_{par} \times |D|}$ calculate $\mathbf{V}$ by: $\mathbf{V} = \mathbf{A} \times 0.3 \times (rand(npar, |D|) - 1)$
while not converge **do**
  for $i = 1$ *to* N_{par} **do**
    $\mathbf{F}(i)$ = calculate (10) given the DBS altitudes $\mathbf{A}(i,:)$
  end for
  $newgbest = max(\mathbf{F})$;
  $index = argmax(\mathbf{F})$;
  if $newgbest > gbest$ **then**
    $gbest = newgbest$
    $\mathbf{gbestA} = \mathbf{A}(index,:)$
  end if
  for $i = 1$ *to* N_{par} **do**
    if $\mathbf{F}(i) > \mathbf{lbest}(i)$ **then**
      $\mathbf{lbest}(i) = \mathbf{F}(i)$
      $\mathbf{lbestA}(i,:) = \mathbf{A}(i,:)$
    end if
  end for
  for $i = 1$ *to* N_{par} **do**
    $\mathbf{V}(i,:) = c_1 \times \mathbf{V}(i,:) + c_2 \times \mathtt{rand}(1,|D|)(\mathbf{lbestA}(i,:) - \mathbf{A}(i,:)) + c_2 \times \mathtt{rand}(1,|D|)(\mathbf{gbestA}(i,:) - \mathbf{A}(i,:))$
  end for
  $\mathbf{A} = \mathbf{A} + \mathbf{V}$
end while

Algorithm 3 X-Cov

1: Start with $|D| = 3$
2: Apply Lloyd's algorithm to find x_i and y_i for $i \in D$
3: $[\mathbf{x}, \mathbf{y}] = Lloyd(\mathbf{locNx}, \mathbf{locNy})$
4: using (13) calculate the altitudes of DBSs
5: **while** Splitting of a DBS occurs and $|D| < K_{max}$ **do**
6:   Apply Lloyds's algorithm to split positions $\mathbf{x}$ and $\mathbf{y}$ in two
7:   $\mathbf{x}_i$ splits to $\mathbf{x}_i^s$ and $\mathbf{y}_i$ splits to $\mathbf{y}_i^s$ for $i \in D$
8:   **for** $i = 1$ *to* $|D|$ **do**
9:     oldVal = Calculate (10) using $\mathbf{x}_i$ and $\mathbf{y}_i$
10:     newVal = Calculate (10) using $\mathbf{x}_i^s$ and $\mathbf{y}_i^s$
11:     **if** newVal-oldVal >0 **then**
12:       Update $\mathbf{x}_i$ using $\mathbf{x}_i^s$
13:       Update $\mathbf{y}_i$ using $\mathbf{y}_i^s$
14:     **end if**
15:   **end for**
16:   Update $|D|$
17: **end while**

3.3 X-means Clustering (X-Cov)

Although K-means clustering is a fast algorithm to determine the locations of DBSs in the horizontal plane, it suffers from two shortcomings. The first issue is that the number of DBSs to be deployed has to be provided by a network planner. Secondly, K-means clustering does not consider interference between the DBSs and places the given number of DBSs over the network. These problems are mitigated by the method called X-means which is proposed by [20]. We adopted X-means clustering approach and modified the algorithm to determine the number of DBSs required and their locations in the horizontal plane. Considering the algorithmic efficiency of the proposed algorithm, the altitude of each DBS is calculated according to (13). The pseudo code of the proposed X-means clustering-based DBS deployment approach is presented in Algorithm 3, which we call X-Cov hereafter.

The X-Cov algorithm starts with 3 clusters (i.e. DBSs) and performs 3-means clustering in order to determine the locations of the 3 DBSs (Line 1-2). Then the algorithm checks each cluster one-by-one and splits it into two. Then it determines and applies the split that results in the best performance improvement, if there is any (Lines 8-15). The algorithm lasts until either there is no more improvement or the maximum number of available DBSs is reached.

3.4 Basic Geometric Approach (BG)

This approach places the given number of DBSs over the network by using simple geometric manner, without taking into account the user locations. Here, we place a DBS in the center of a cell and then place the remaining DBSs symmetrical around the origin. The altitude of each DBS is calculated using (13). This is the simplest algorithm by far amongst the proposed algorithms and serves as a benchmark. We call this method as BG hereafter.

4 Simulation Results

We use Matlab software and a computer with Intel i7 quad core processor with a clock speed of 2.6GHz and 4GB RAM to carry out our simulations. For each number of DBSs ($|D|$) available, we run the proposed algorithms 100 times and average out the results to obtain the final results we present in the tables and figures. We consider two different user distribution scenarios. In the first scenario (Scenario I), the users are uniformly distributed over the network, whereas in the second scenario (Scenario II), 10 parent points are uniformly generated over the circular area and then child points are

Table 1 Simulation Parameters

Parameter	Definition	Value
f_c	Carrier Frequency	$2GHz$
R_{max}	Radius	$1500m$
W	Bandwidth	$20MHz$
U	Number of Users	100
N_0	Noise Power Spectral Density	$-170dBm/Hz$
α, β	Environmental Parameters	$9.61,\ 0.16$
η_{LoS}, η_{NLoS}	Mean Pathloss	$1dB,\ 20dB$
θ_B	DBS Antenna Beamwidth	140°
P_T	DBS Transmission Power	$30dBm$
N_{par}	Number of Particles	100
P_{conv}	Convergence of Particles	10^{-4}
h_{min}, h_{max}	Min. and Max. Altitude	$200m,\ 800m$
N_{MC}	Monte Carlo Simulations	100
L_{TH}	Low Data Rate Threshold	$500Kbps$

uniformly distributed in the range $[0, 300]$ metres around the parent points. The simulation parameters are provided in Table 1. The air to ground path loss parameters are obtained from [6]. For the radius of $1.5km$, the total area is calculated as $7.06km^2$. In order to measure the fairness performance of the deployment strategies, we calculated 3 different metrics namely,

1. Jain's fairness index,
2. Max/Min ratio,
3. The number of users having data rate under the specified threshold ($500Kbps$ in our case).

Jain's fairness index is widely used to evaluate the fairness performance of the networks [21] and it is calculated by [22]

$$J = \frac{\left(\sum_{i=1}^{|U|} C_i\right)^2}{|U| \sum_{i=1}^{|U|} C_i^2} \tag{14}$$

where C_i is the data rate of i^{th} user. Jain's metric being close to 1 means a good fairness performance.

The second fairness metric, Max/Min ratio, is defined as the ratio between the highest and lowest data rate of the users [21]. The last fairness metric we use is the number of users whose data rates are lower than the specified threshold. We choose this metric considering the fact that in case of a disaster, it is critical to ensure that every user should receive up-to-date information from the network.

In Table 2, we provide the fairness metrics of the algorithms when there are maximum of 7 DBSs available (denoted as N_{DBS}^A) for deployment. The metrics show that the LogPSO method outperforms all other algorithms considering all fairness metrics for both Scenario I and II. However the proposed K-APSO algorithm is almost as good as LogPSO in terms of proportional fairness. Considering its simplicity, K-Cov also performs surprisingly well. X-Cov performs especially well in the case of heterogeneous user distribution. The reason is that when the users are distributed in a clustered manner it is easier to split them into clusters and use a suitable number of DBSs. The proposed K-Cov and K-APSO also performs almost optimal in terms of Jain's Index and average number of low rate users. On the other hand LinPSO has poor fairness performance, as it does not aim fairness. Finally the benchmark BG method performs the worst, which shows the necessity of intelligent placement of DBSs.

The methods LogPSO, K-Cov and K-APSO perform quite similarly. However, as it can be seen from Table 3, which shows the average execution times of the algorithms to solve a deployment problem when $N_{DBS}^A = 7$, the execution time of the LogPSO is much longer than those of the less complex algorithms. These results indicate that K-Cov, K-APSO and X-Cov can be used in practical implementations with a negligible loss of performance. On the other hand LogPSO and LinPSO may be problematic in scenarios with high mobility, as the topology may significantly change during computations.

Table 2 Fairness Performance of Algorithms

Algorithm	SumLog x10^3		Max/Min		Jain's Index		Low Rate Users	
	I	II	I	II	I	II	I	II
LogPSO	1.618	1.618	17.03	12.4	0.82	0.73	0.03	0
LinPSO	1.571	1.610	311.8	61.03	0.37	0.65	7.15	2.19
K-Cov	1.615	1.609	20.29	24.76	0.80	0.69	0.14	0.27
K-APSO	1.615	1.611	23.05	25.54	0.81	0.69	0.18	0.31
X-Cov	1.587	1.615	27.25	26.66	0.78	0.71	0.54	0.37
BG	1.574	1.574	65.78	25.15	0.63	0.62	3.52	17.16

Table 3 Execution Times of the Algorithms (s)

	LogPSO	LinPSO	K-Cov	K-APSO	X-Cov	BG
N_{DBS}	35.24	36.32	0.032	0.82	0.18	0.003

In Figure 2, sum log rate (proportional fairness) performance of the algorithms for different number of DBSs are shown for Scenario I. As expected, the LogPSO achieves the best capacity for each case. The worst performing algorithms are BG and LinPSO. BG is not a smart algorithm and proposed to understand to what extent complex algorithms for the deployment of DBSs problem is required. The results of BG in both scenarios prove that the DBS deployment algorithms must be considered carefully. LinPSO performs second worse because it focuses only on the users with good channel conditions and do not consider the others which are located far away from them. K-APSO and K-Cov perform almost as good as the LogPSO method, with much shorter run time. The performance of X-Cov is in between the best and the worst performing algorithms.

Using excessive number of DBSs may result in interference, which has a detrimental effect on the performance. We present Table 4 in order to understand the actual average number of DBSs that the algorithms use given a certain number of DBSs(denoted as N_{DBS}). This can also be defined as the total number of DBSs, where each DBS serves at least one user in the network. The results are obtained for Scenario II, in which the deployment problem is more difficult because the clustered distribution of users may bring serving DBSs closer, causing excessive interference. Deployment algorithms should be aware of this situation and avoid deploying DBSs too close to each other. Results show that K-means based algorithms, namely K-Cov and K-APSO, do not provide a mechanism to determine N_{DBS} and simply place all N_{DBS}^A DBSs over the network which cause interference hence performance loss. The proposed LogPSO can achieve this by throwing the unnecessary DBSs out of the circular area as a result of swarm intelligence. Same behavior is observed

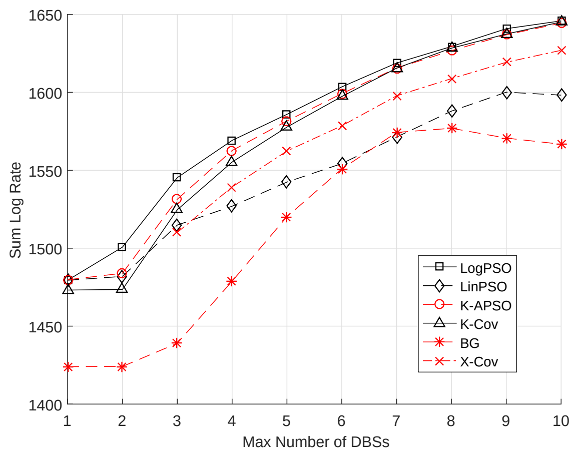


Fig. 2: Sum Log Rate Performance for Scenario I

Table 4 Average Number of Deployed DBSs

	LogPSO	LinPSO	K-Cov	K-APSO	X-Cov	BG
N_{DBS} (Max=4)	3.94	3.99	4	4	4	3.92
N_{DBS} (Max=7)	5.91	6.17	7	7	6.89	6.29
N_{DBS} (Max=10)	7.03	7.50	10	9.74	8.07	7.92

in LinPSO as it is also based on swarm intelligence. These algorithms understand that the overall performance of the network may get better when some DBSs are not used then these DBSs are moved to the positions where they can not serve to any user. As an example, when $N^A_{DBS} = 10$, LinPSO and LogPSO use around 7 DBSs to serve the users. The method X-Cov finds a somewhat suitable N_{DBS} by applying the splitting mechanism. Although BG does not have any specific mechanism, as the distance between DBSs in the circular line gets closer, some DBSs become redundant and no user is connected to them. However, BG does not reduce N_{DBS} in order to manage the interference because the redundant DBSs continue to transmit as a source of interference in their fixed locations which are determined without considering user locations. As a result of strong interference, BG achieves the worst performance results.

In Figure 3, the sum rate performance of the proposed algorithms versus N^A_{DBS} is presented. As expected, LinPSO provides the best sum-rate as LinPSO is designed to maximize sum rate capacity. When there are 5 DBSs available for the deployment, LinPSO exceeds the capacity of second best LogPSO and third best K-APSO by %11.2 and %15 respectively. The increasing number of DBSs narrows the performance difference of K-APSO and K-Cov algorithms. The reason is that the users are guaranteed to be in a coverage of a DBS when the altitude is at the minimum. When there is not enough DBS in the network, altitude of a DBS is critical and need to be planned carefully to cover the users in its area. The performance of X-Cov algorithm does not provide good results but still much better than BG.

Figure 4, the sum log rate of the algorithms is shown for Scenario II. Again, LogPSO achieves the best results for each case. Different than the results obtained in Scenario I, LinPSO and X-Cov provide acceptable performance behind LogPSO. However, K-Cov and K-APSO which produce good performance in Scenario I, fail to locate DBSs efficiently in Scenario II. The reason for this situation is that the unawareness of K-means clustering algorithm to the interference incurred at the user terminals. K-Cov and K-APSO uses all the DBSs available without considering the interference. Figure 6 shows the aforementioned situation when $N^A_{DBS} = 10$. Figure 6a shows that the LogPSO method considers 4 DBSs as redundant (and throws them far away) and suitably deploys the other 6. K-Cov tries

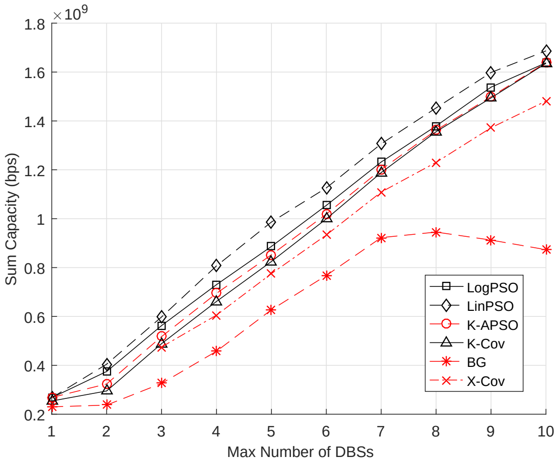


Fig. 3: Sum Rate Performance for Scenario I

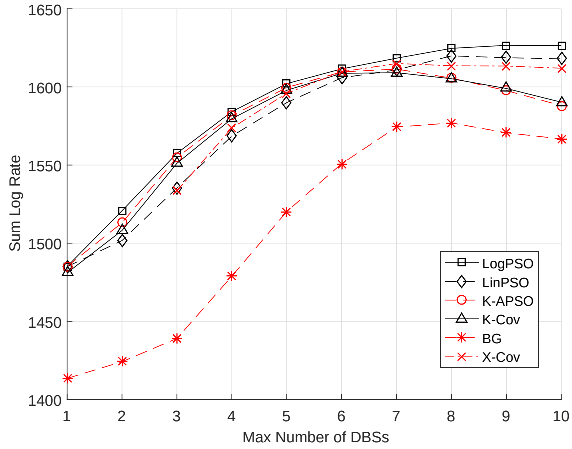


Fig. 4: Sum Log Rate Performance for Scenario II

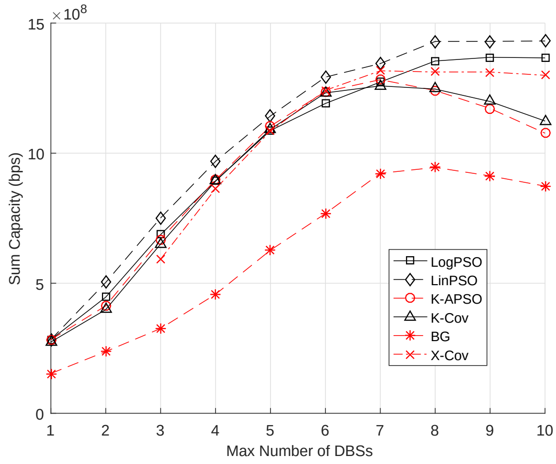


Fig. 5: Sum Rate Performance for Scenario II

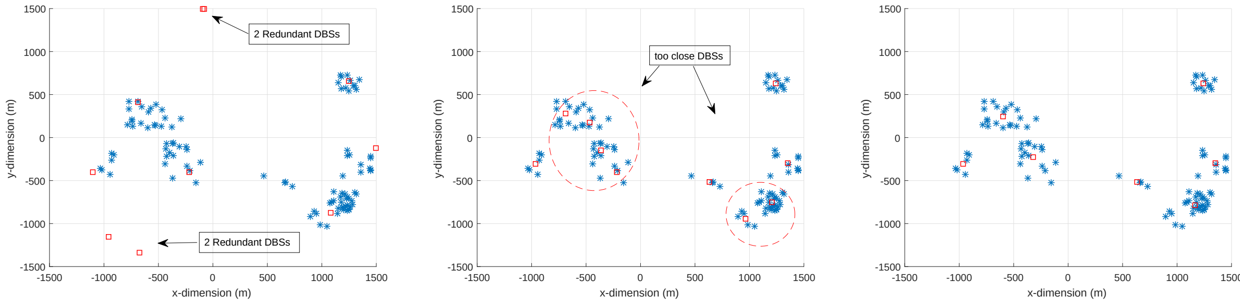
to deploy all available 10 DBSs and creates interference. Finally X-Cov uses 7 of the available 10 DBSs. Please note Figure 6c that it uses one extra DBSs to cover a few isolated nodes.

The capacity reducing effect of interference is clearly shown in the Figure 4 and Figure 5, when more than 6 DBSs are deployed in the network. As X-Cov uses K-means clustering considering the local capacities in each splitting decision, it limits the number of DBSs deployed and hence interference. As a result, X-Cov is an alternative to PSO based algorithms if the processing power is limited. Figure 5, shows that the capacity limit is achieved when there are 8 DBSs in the network. The capacity limit is much higher in Scenario I, because the network serves the users with more DBSs without bringing them spatially too close. We do not investigate the N_{DBS} required to achieve capacity limit of Scenario I, as this is not a motivation of this article.

5 Conclusion and Future Work

In this paper, we have studied the multiple DBS deployment considering the fairness amongst users. The proposed algorithm, which maximizes the sum of user's log rates by using Particle Swarm Optimization, termed LogPSO has provided the best fairness performance in two different network scenario and uses the optimal number of the available DBSs. It is observed that the PSO requires significant processing power and may not be suitable for some applications. However, the proposed K-means and X-means clustering

IET Research Journals, pp. 1–7



(a) LogPSO (b) K-Cov (c) X-Cov

Fig. 6: Deployment of DBSs when $N_{DBS}^{A} = 10$

based algorithms can be used as an alternative to the LogPSO in networks with different user distributions. The best algorithm to plan the deployment of multiple DBSs can be chosen according to the processing power of the system and performance requirements of the network.

As for the future work, we plan to address dynamic management of DBSs in case there exists mobile users in the network. Considering various mobility models, we will analyze the continuous coverage of users by applying markov chain modelling on the number of DBSs required. Another direction for the future work would be a device-to-device (D2D) enabled DBS network, where the DBS coverage can be augmented by users that have D2D communication capability. D2D technology also enables terminal relaying. In addition to pathloss and shadowing that are considered in this work, there is also short term multipath fading. D2D relaying has potential to provide diversity, reduce outage probabilities and improve the fairness performance.

6 References

1 Sun, Z., Wang, P., Al-Rodhaan, M.A., et al.:’BorderSense: Border patrol through advanced wireless sensor networks’,Ad Hoc Networks, 2011, 9, (3), pp. 468-477
2 Maza, I.,Caballero,F.,Dios,J.M., et al.:’Experimental results in multi-UAV coordination for disaster management and civil security applications’, J. Intelligent and Robotic Systems,2011,61,(1-4),pp.563-585
3 S udarshan, K.V.S., Montano, V., Nguyen, A., et al.:’A Heterogeneous Robotics Team for Large-Scale Seismic Sensing’, IEEE Robotics and Automation Letters IEEE, 2017, 2, (3), pp. 2377-3766
4 Pederi, Y.A., Cheporniuk, H.S.:’Unmanned Aerial Vehicles and new technological methods of monitoring and crop protection in precision agriculture’. IEEE International Conference Actual Problems of Unmanned Aerial Vehicles Developments, Kiev, Ukraine, Dec 2015, pp.298- 301
5 ’Amazon PrimeAir’, https://www.amazon.com/b?node=8037720011, accessed 10 August 2017
6 Al-Hourani, A., Kandeepan, S., Jamalipour, A.: ’Modeling air-to-ground path loss for low altitude platforms in urban environments’. Proc. 2014 IEEE Global Communications Conf., Austin, TX, USA, Dec 2014, pp. 2898-2904
7 Al-Hourani, A., Kandeepan, S., Lardner, S.: ’Optimal LAP altitude for maximum coverage’, IEEE Wirel. Commun. Lett., 2014, 3, (6), pp. 569-572
8 Mozaffari, M., Saad, W., Bennis, M., et al.:’Drone Small Cells in the Clouds: Design, Deployment and Performance Analysis’. IEEE Global Commun. Conf.,San Diego, CA, Dec 2015, pp. 1-6
9 Bor-Yaliniz, R.I., El-Keyi, A.,Yanikomeroglu, H.: ’Efficient 3-D placement of an aerial base station in next generation cellular networks’. IEEE Int. Conf. on Commun., Kuala Lumpur, Malaysia, May 2016, pp. 1-5
10 Kalantari, E., Shakir, M.Z.,Yongacoglu, A.: ’Backhaul-aware robust 3D drone placement in 5G+ wireless networks’. IEEE Int. Conf. on Commun. Workshops., Paris, France, May 2017, pp. 109-114
11 Alzenad, M., El-keyi, A., Lagum, F., et al.:’3D Placement of an Unmanned Aerial Vehicle Base Station (UAV-BS) for Energy-Efficient Maximal Coverage’, IEEE Wireless Commun. Letters, 2017, PP, (99), pp. 1-1
12 Mozaffari, M., Saad, W., Bennis, M., et al.: ’Efficient Deployment of Multiple Unmanned Aerial Vehicles for Optimal Wireless Coverage’, IEEE Commun. Letters, 2016, 20, (8), pp. 1647-1650
13 Lyu, J., Zeng, Y., Zhang, R.,et al.: ’Placement Optimization of UAV-Mounted Mobile Base Stations’, IEEE Commun. Lett., 2017, 21, (3), pp. 604-607
14 Kalantari, E., Yanikomeroglu, H.,Yongacoglu, A.: ’On the Number and 3D Placement of Drone Base Stations in Wireless Cellular Networks’. Proc. IEEE 63rd Vehicular Technology Conf., Montreal, Canada, Sept 2016, pp. 1-6
15 Cirik, A.C.:’Fairness Considerations for Full Duplex Multi-User MIMO Systems’, IEEE Wireless Commun. Lett., 2015, 4, (4), pp.361–364
16 Calabuig, D., Monserrat, J.F., Cardona, N.:’Fairness-Driven Fast Resource Allocation for Interference-Free Heterogeneous Networks’, 2012, 16, (7), pp. 1092-1095
17 Eberhart, R., Kennedy, J.: ’A new optimizer using particle swarm theory’. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, Oct 1995, pp. 39-43
18 Kanungo, T., Mount, D.M., Netanyahu, N.S., et al.: ’An efficient k-means clustering algorithm: analysis and implementation’, IEEE Trans. on Pattern Analysis and Machine Intelligence, 2002, 24, (7), pp. 881-892
19 Lloyd, S.P.:’Least squares quantization in PCM’, IEEE Trans. Information Theory, 1982, 28, (2), pp.129-137
20 Pelleg, D., Moore, A.W.: ’X-means: Extending K-means with Efficient Estimation of the Number of Clusters’. Proceedings of the Seventeenth International Conference on Machine Learning, San Francisco, CA, USA, June 2000, pp. 727-734
21 Shi, H., Prasad, R.V., Onur, E., et al.: ’Fairness in Wireless Networks:Issues, Measures and Challenges’, IEEE Commun. Surveys Tutorials, 2014, 16, (1), pp. 5-24
22 Jain, R., Chiu, D., Hawe, W.:’A quantitative measure of fairness and discrimination for resource allocation in shared systems’. Tech. Rep. DEC-TR-301, Sep. 1984