Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Efficient Optimal Backhaul-aware Placement of Multiple Drone-Cells Based on Genetic Algorithm*

Xiaohui Li
School of Electrical Engineering and Telecommunications
University of New South Wales
Sydney 2052, Australia
xiaohui.li@unsw.edu.au

Li Xing
Department of Electrical and Electronic Engineering
University of Melbourne
Melbourne 3010, Australia
lxing@student.unimelb.edu

***Abstract*— Drone-cell is a promising solution for providing GSM/3G/LTE/5G mobile networks to victims and rescue teams in disaster-affected areas. One of the challenges that drone-cells are facing is the limitation of reliable backhaul links. In this paper, we study the optimal placement problem of a group of drone-cells deployed in a disaster area with limited backhaul communication ranges, aims to maximize the number of served users. Two approaches, exhaustive search algorithm and a computationally efficient genetic algorithm are proposed, and the optimal 2D backhaul-aware placements of multiple drone-cells are found for each approach. We also introduce a restart-strategy to enhance searching efficiency and avoid local optimums for the proposed genetic algorithm. Simulations show that the proposed genetic algorithm can save the computing time up to 99.927% compared with the exhaustive search algorithm, and the restart-strategy helps the probability of finding the global optimum by the proposed genetic algorithm increased from 12% to 92%.**

***Index Terms*— Drone-cell, emergency communication, backhaul limits, placement optimization, genetic algorithm.**

I. Introduction

Establishing emergency communication networks has long been a difficult problem in disaster areas, where communication can save lives. With the capability of rapidly moving communication supply towards demand when required, low-altitude unmanned aerial vehicles equipped with base stations, i.e., drone-cells, have recently attracted a lot of attention [1]. As a promising solution to temporarily provide cellular networks in an area that has lost coverage, drone-cells can serve as aerial base stations with a quick deployment opportunity [2]. One of the biggest challenges, however, is to determine the optimal placement of drone-cells so that users can benefit the most.

A considerable amount of literature has been published on drone-cells‘ placement problem. Reference [2] formulated the 3D placement problem into a mixed-integer non-linear problem (MINLP) and solved the problem by bisection search and the interior-point optimizer of the MOSEK solver. Reference [3]–[5] studied the problem of deploying unmanned aerial base stations to serve mobile users based on apriori user density function, which reflects the traffic demand at a certain position in the operating area. Reference [6], [7] considered the drone-cells' deployment problem without apriori user locations. Besides the mobility, another major difference between a ground cell and a drone-cell is that the latter one relies on wireless links for the backhaul connection, while a ground-cell usually has a fixed wired backhaul link. Therefore, one of the major limitations in drone-cells' deployment is the availability of reliable wireless backhaul links. While researchers have considered various aspects regarding the deployment of drone-cell for wireless coverage, the backhaul limitations of drone-cell have not been treated in many details. The backhaul link of drone-cell may be satellite-based, dedicated, or in-band. Satellite-based backhauling requires the drone to be equipped with a satellite transceiver for establishing the connection via a backhaul satellite. Dedicated backhauling could be free-space optical communication (FSO) or mmWave link between the drone and core networks. As for in-band backhauling, the main technology currently used in the wireless backhaul links of LTE or Wi-Fi is based on RF microwave [8]. Considering that the satellite transceiver is both expensive and energy inefficient, the drone-cells with dedicated or in-band backhauling are more practical.

Existing research recognizes the critical role played by the backhaul limitation while design and deploying the drone-cells include [1], [8], [9]. Reference [9] proposed a framework that utilizes the flying capabilities of the UAV-BSs as the main degree of freedom to find the optimal precoder design for the backhaul links, user-base station association, UAV-BS 3D hovering locations, and power allocations by exhaustive search. Reference [1] investigated the optimal 3D placement of a drone-cell over an urban area with users having different rate requirements, considering both the wireless backhaul peak rate and the bandwidth of a drone-cell as the limiting factors. Particularly, Reference [1] adopt the branch-and-bound method and exhaustive search in the

*This work was supported by the Australian Research Council. Also, this work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571.

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

step size of 100 meters to search drone-cell's 3D location for maximizing the total number of served users and sum-rates, which has the same problem as [9] that the solutions are inefficient and cannot guarantee accurately optimal results. Moreover, these two approaches are simply assuming that the drone-cell are connected to remote ground-cells, which is not practical for ignoring the communication range constraint of the backhaul link. In [4], the authors introduces using the robust extended Kalman filter to estimate users' locations based on the received signal strength indication, and proposes a decentralized algorithm to find a locally optimal solution for coverage maximizing while considering both the collision avoidance and backhaul limitation of drone-cells, the results, however, may not be globally optimal.

In this paper, we study the optimal placement problem of a group of drone-cells deployed in a disaster-affected area with limited backhaul communication ranges, aims to maximize the number of served users. Particularly, the backhaul communication of drone-cells can be achieved by wireless link direct to a ground base station (GBS) or through the wireless links to the other drone-cells as the bridge to the GBS, which is connecting to the core network. The problem is an NP-Hard problem. We first proposed an exhaustive search algorithm that can find the quasi-optimal placements for the drone-cells. After that, a computationally efficient algorithm based on genetic algorithm (GA) is proposed to cope with the complexity of the problem; however, it could potentially produce, on some occasions, poor solutions. A subsequent restart-strategy for the GA to enhance searching efficiency and avoid local optimum is introduced. The rest of this paper is organized as follows. In Section 2, the system model is presented. Two algorithms, including exhaustive search algorithm and GA to find the optimal backhaul-aware placement of multiple drone-cells are described in Section 3, followed by a detailed presentation of the results and related discussions in Section 4.

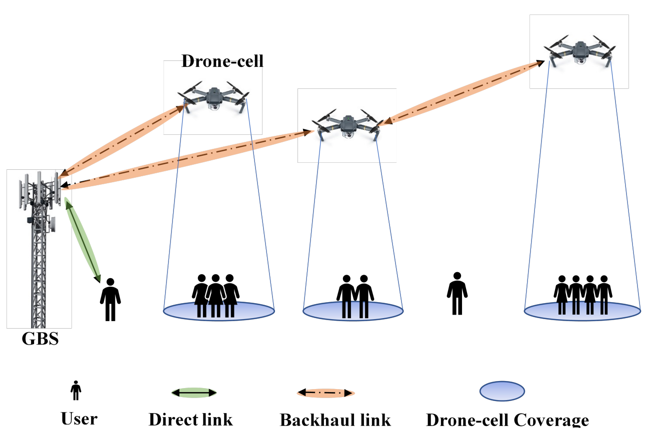

Fig. 1. An illustration of the considered scenario.

II. System Model

Consider a relative lager disaster-affected where most of the existing g cellular GBSs were manufactured because of the disaster, resulting in that the users in such an area have no access to the cellular network. In this case, a group of drone-cells is deployed to build a multi-tier drone cellular network. The first tier includes GBSs that, on the one hand, provide direct access links to users within its coverage and on the other hand, provides backhaul links to drone-cells. The second tier represents the drone cells and the users associated with them. An illustration of the considered scenario is shown in Fig. 1. With this regard, we considered a transmission system consisting of m drone-cells labeled $i = 1, 2, \ldots, m$ with k surviving GBS. Some drone-cells should be able to directly communicate to any of the GBS through the backhaul link. These drone-cells can also bridge the other remote drone-cells to GBSs through backhaul links so that all the drone-cells are connecting with the core network. We assume that some very remote drone-cell may rely on a long backhauling bridge consists of two or more drone-cells to connect to the GBS. Since the network is built for emergency use, we assume the drone-cells try to serve as many users as possible, regardless of their rate requirements. Practically, the number of available drone-cells is often limited, while the disaster-affected areas tend to be relatively large. Hence we deploy the drone-cells to try to cover a maximum number of users, and some of the isolated users may remain uncovered.

According to [10], the optimal altitude and the corresponding coverage radius of the drone-cell can be numerically solved for a certain environment (urban, suburban) and a given path loss threshold. Therefore, we deploy all the drone-cells at the same altitude in one horizontal plane. Hence, we consider the drone-cells' coordinates at that plane as coordinates on the surface. The coverage of the drone-cell is disk-like with the drone-cell' projection on the ground as the center and a coverage radius of R_d. Let the set $G = \{G_1, G_2, \cdots, G_k\}$ denotes the coordinates of the GBSs' locations, $D = \{D_1, D_2, \cdots, D_m\}$ denotes the coordinates of the drone-cells' deployment locations , where $G, D \in R^2$. Considering the limitation of the communication range of such backhaul links, we require that the drone-cells and GBSs, which should be able to communicate by direct backhaul link, are within certain ranges ensure the backhaul links are reliable. We will consider the deployment of drone-cells at any time satisfying the following constraints:

$$|D_i, G_h| \leq r_1 \tag{1}$$

If drone-cell D_i and the corresponding GBS G_h are connected by a direct backhaul link; and

$$|D_i, D_h| \leq r_2 \tag{2}$$

If drone-cells D_i, D_h are connected by a direct backhaul link. Here $|,|$ denotes the standard Euclidean distance between two points. r_1, r_2 are some given constants describing the communication ranges between drone-cells and a GBS and another drone-cell, beyond which the wireless backhaul links may be unreliable. Practically, r_1, r_2 should always hold that:

$$r_1 \geq R_g + R_d \tag{3}$$

$$r_2 \geq 2R_d \tag{4}$$

Where R_g is the coverage radius of the GB. The values of r_1, r_2, R_g, R_d are influenced by actual signal effects in the actual environment.

To obtain heterogeneity in spatial user distribution, we adopt a doubly Poisson cluster process: $mat\acute{e}rn$ cluster process [10]. Where the centers of user clusters are created by a homogeneous Poisson process. The users within each cluster are uniformly scattered in circles with radius R_c around parent points by using another homogeneous spatial Poisson process [1]. The density function of clustered users in location z is

$$f(z) = \begin{cases} \frac{1}{\pi R_c^2}, & if \|z\| \leq R_c \\ 0, & otherwise \end{cases} \tag{5}$$

Suppose there are uncovered users labelled $i = 1, 2, \ldots, n$. Let the set $U = \{U_1, U_2, \cdots, U_n\}$ denotes the coordinates of the locations of all uncovered uses within the operating area, exclude the users that already been covered by any of the GBSs through direct link. Our target is to find the optimal 2D deployment $D = \{D_1, D_2, \cdots, D_m\}$ so that the number of served users is maximized. Our deployment optimization problem is formulated as follows:

$$\max_{D, \{I_j\}} \sum_{j=1}^{n} W_j \tag{6}$$

Subject to:

$$R_g + R_d \leq |D_i, G_h| \leq r_1 \tag{7}$$

$$2R_d \leq |D_i, D_h| \leq r_2 \tag{8}$$

$$W_j = \begin{cases} 1, & if\ |D_i, U_j| \leq R_d, \forall i, j \\ 0, & otherwise \end{cases} \tag{9}$$

Where (7),(8) indicate that: besides the backhaul link constraints, the coverage area of a drone-cell is not allowed to overlap with the coverage area of any GBS or another drone-cell. Moreover, $W_j = 1$ if user j is covered by any drone-cell and $W_j = 0$ otherwise. As one of the important features of a drone-cell is its fast mobility, drone-cells can change their deployment rapidly to follow the movement of users if needed. Moreover, for battery-life saving purpose, if a drone-cell flies to a suitable location, it can stay there for a while until the network reaches a particular pre-determined threshold of user-dropped out. Therefore, in this paper, we find the deployment of the drone-cells for only one snapshot of the users' positions. The users' positions are assumed to be apriori, the methods of acquiring such positions information can be seen in [7], [11].

III. Backhaul-aware Placement Algorithms

Generally, the optimal placement of drone-cells is a NP-hard problem [12]. Adding a new dimension to the problem, which is the backhaul link constraints of the drone-cells, makes the problem even more complicated. To find the backhaul-aware placement of a group of drone-cells ensuring that a maximum number of users can be served, we first propose an exhaustive search algorithm that solves the problem using brute force. After that, we introduce a much more efficient solution an algorithm based on GA with restart-strategy. We then compared the performance of the two algorithms through a bunch of simulations.

A. Exhaustive Search Algorithm

Before describing the algorithm, we first investigate the optimal placement problem for a single drone-cell. Which is very similar to a classic circle placement problem: given a set of N points $P = \{p_0, p_1, \ldots, p_N\}$, in a 2-D plane, and a fixed disk of radius R, find a location to place the disk such that the total number of the points covered by the disk is maximized.

One of the most efficient solutions for the circle placement problem is an $O(N^2)$ greedy algorithm based on finding the maximum *Clique* (graph theory) in an intersection graph. This is the approach developed by [13] in 1984. The key observation is that finding the maximum number of points covered by a single disk with fixed radius R is the same as finding a single point covered by the maximum number of disks with radius R. Moreover, if a point is covered by k disks, then those disks must have their mutual distances all less than $2R$. Therefore, an intersection graph consists of N vertices can be created. Each of the vertices is corresponding to one disk, and two vertices are connected by an edge if the centers of those disks are less than $2R$ distance apart. If a single point is covered by k disks, then these disks' corresponding vertices are all connected to each other by edges and therefore form a *Clique*. Reference [13] then proposed a greedy algorithm to find the largest *Clique*, which solved the circle placement problem accordingly, more details can be found in [13].

When solving the single drone-cell's optimal placement problem by the solution from [13], it must be noted that (7),(8) as the deployment constraints should always be satisfied. Which means the drone-cell can only be deployed in a bounded closed region $\mathcal{U}$ decided by the locations of the GBSs and the other drone-cells that have got their locations fixed. We now make use of the notion of *deployment region*.

Definition 1. Deployment region: A deployment region is the limited bounded closed region that a drone-cell can to be deployed with the locations of GBSs and the other drone-cells are fixed, and constraints (7), (8) are satisfied.

An example of the deployment regions $\mathcal{U}$ of a drone-cell

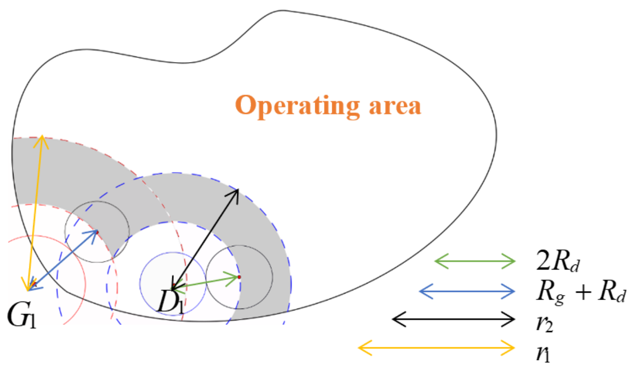

Fig. 2. The deployment region $\mathcal{U}$ (grey area) of a drone-cell decided by a GBS G_1 and a deployed drone-cell D_1.

is shown in Fig. 2. We now describe the exhaustive search algorithm in its entirety in Algorithm 1. The variables with the subscript like $(j1, ..., jt)$ mean these variables are decided by the locations of the first drone-cell D_1 to the tth drone-cell D_t. To explain the proposed exhaustive search algorithm in more detail. We take the example of deploying a group of $m = 3$ drone-cells and introduce applying the exhaustive search algorithm to find their optimal placements. We start with finding the deployment region $\mathcal{U}^1$ of the first drone-cell D_1. Let set $U^1 = \left\{U_1^1, U_2^1, \cdots, U_{N1}^1\right\}$ denotes the locations of all the $N1$ users within $\mathcal{U}^1$. Visit each $U_{j1}^1 \in U^1$, where $j1 \in \{1, 2, \cdots, N1\}$. Let $D_1 = U_{j1}^1$, find the number of users n_{j1} that can be covered by D_1 and the corresponding deployment regions $\mathcal{U}_{j1}^2$ for the second drone-cell D_2. n_{j1} is simply the number of users $U_{j1}^1 \in U^1$ satisfy $\left|U_{j1}^1, D_1\right| \leq R_d$. Let set $\mathcal{U}^2 = \left\{\mathcal{U}_1^2, \mathcal{U}_2^2, \cdots, \mathcal{U}_{N1}^2\right\}$ denotes all the candidate deployment regions of D_2. For each $\mathcal{U}_{j1}^2 \in \mathcal{U}^2$, let set U_{j1}^2 denotes the locations of all the users within $\mathcal{U}_{j1}^2$. Visit each $U_{j1,j2}^2 \in U_{j1}^2$, let $D_2 = U_{j1,j2}^2$, find the number of users $n_{j1,j2}$ that can be covered by D_2 and the corresponding deployment regions $\mathcal{U}_{j1,j2,j3}^3$ for the third drone-cell D_3. Then, apply the solution from [13] to find the maximum number of users $n_{j1,j2,j3}$ can be covered by D_3 and its corresponding optimal placement $U_{j1,j2,j3}^3$. Finally, compare and find the combination $\{j1^*, j2^*, j3^*\}$ that maximizes $(n_{j1}+n_{j1,j2}+n_{j1,j2,j3})$ and the corresponding optimal placements $[D_1, D_2, D_3] = [U_{j1^*}^1, U_{j1^*,j2^*}^2, U_{j1^*,j2^*,j3^*}^3]$. Note that the solution generated by proposed Algorithm 1 is only an approximation of the global optimum of the considered placement problem, i.e., quasi-optimal. This is because the placements of the first $(m-1)$ drone-cells are limited to some locations of the users, not any points on the candidate deployment regions.

Lemma 1. The worst-case running time of the exhaustive search algorithm is $O(n^{2m})$.

The proof of this lemma will be given in a full version of this paper.

Algorithm 1 Exhaustive Search Algorithm for Optimal Backhaul-aware Deployment of Drone-cells.

Input:
$G = \{G_1, G_2, \cdots, G_k\}, U = \{U_1, U_2, \cdots, U_n\}, m, R_d, R_g$
Output:
$D = \{D_1, D_2, \cdots, D_m\}$

$t = 1$;
while $t < m - 1$ **do**
Find the set $U^t \in U$ contains the coordinates of users within all the candidate deployment regions of the tth drone-cell D_t;
Visit all users in U^t, for each of the user $U_{j1,...,jt}^t \in U^t$, let $D_t = U_{j1,...,jt}^t$, calculate the number of users $n_{j1,...,jt}$ that D_t can cover, record $n_{j1,...,jt}$ and its corresponding $U_{j1,...,jt}^t$ as the candidate of D_t. Then, find the set U^{t+1} contains the coordinates of users within all the candidate deployment regions of the $(t+1)$th drone-cell D_{t+1};
Visit all users in U^{t+1}, for each of the user $U_{j1,...,j(t+1)}^{t+1} \in U^{t+1}$, let $D_{t+1} = U_{j1,...,j(t+1)}^{t+1}$, calculate the number of users $n_{j1,...,j(t+1)}$ that D_{t+1} can cover and its corresponding $U_{j1,...,j(t+1)}^{t+1}$ as the candidate of D_{t+1};
$t = t + 1$;
end while
Find all the candidate deployment regions $\mathcal{U}_m$ for the mth drone-cell D_m. For each candidate deployment region $\mathcal{U}_{j1,...,jm}^m \in \mathcal{U}_m$, apply the solution from [13] to find the maximum number of users $n_{j1,...,jm}$ can be covered by D_m and its corresponding optimal placement $U_{j1,...,jm}^m$;
Compare and find the combination $\{j1^*, j2^*, \ldots, jm^*\}$ that gives **max** $(n_{j1} + n_{j1,j2} + \ldots + n_{j1,...,jm})$, and the corresponding optimal placement candidates set $U^* = \{U_{j1^*}^1, U_{j1^*,j2^*}^2, \ldots, U_{j1^*,...,jm^*}^m\}$;
return $D = U^*$;

B. Genetic Algorithm

Metaheuristic algorithms such as GA, simulated annealing, and particle swarm optimization are often used in solving NP-Hard problems. Here we use GA to find the optimal backhaul-aware placement of the drone-cells. The GA is a heuristic search algorithm developed by John Holland in the 1960s and first published in 1975 [14]. Inspired by Darwin's theory of evolution, GA has long been a widely used non-deterministic optimization method. Specifically, GA simulates the evolution of a population of chromosomes to optimize a problem. Similarly to organisms adapting to their environment over the generations, the problems to be solved by GA are simulated as a process of biological evolution. The chromosomes in GA adapt to a fitness function over an iterative process. The next generation of chromosomes is generated by using biology-like operators such as the crossovers, the mutations, and the inversions. The chromosomes with low fitness function value will be eliminated gradually, while the percentage of those with high fitness function value is increased. In this way, after a certain number of iterations, it is possible to evolve solutions with high fitness function values.

In this work, we use GA to simulate the evolution of a population of drone-cells' placements adapting to the fitness function defined simply by the total number of users covered by the drone-cells. In detail, the GA begins by encoding and creating a random initial population. Since the coordinates of the deployed drone-cells are real numbers, direct value encoding is used for our problem. Each chromosome is a 2D array with a size of m by 2, which stores the 2D coordinates of the group of m drone-cells. The initial Let $\mathcal{P}_{size}$ denotes the number of chromosomes in the population, i.e., the population size. The initial population is generated randomly, with each drone-cell coordinates in the chromosomes bounded in the operating area. Then we calculate the fitness function $f(I)$ for each of the chromosomes I. Where $f(I)$ equals the number of users covered by the drone-cells with their coordinates satisfy the backhaul constraints (7),(8). Afterward, we start the iterations of evaluation. Let $iter\_max$ denotes the maximum number of iterations. In each of the next generations, the children chromosomes from the parent ones are produced following three steps:

1) Mutation: In this step, the following adaptive mutation operator is applied with a mutation probability P_m to some genes g (coordinates of a drone-cell in the X-axis or Y-axis) of the chromosomes.

$$g' = g\left[1 \pm rnd\left(1 - \frac{f}{f_{\text{best}}}\right)^2\right] \tag{10}$$

Where $rnd \in [0, 1]$ is a random value, f is the fitness of the current chromosome, f_{best} is the current highest fitness achieved by the population. It is noticeable that the influence of the mutation operator becomes tiny if f is close to f_{best}, this is used to limit adverse moves and protect the best chromosome of every population. Moreover, to improve the efficiency of the GA and the consistency of the final results. We adopt Simulated-Annealing Mutation (SAM) [15] operator that uses the Simulated-Annealing stochastic acceptance function internally to limit adverse moves. Specifically, we calculate the difference between the fitness before and after the mutation δ_f. If the fitness after mutation decreases, the annealing probability $\exp(\delta_f/T)$ is given to determine whether to accept the mutation or not. Where T is an adjustable constant. Let $T = \alpha T$ after each iteration, where $\alpha \in [0, 1]$ is a constant to let T gradually decreases after each iteration.

2) Crossover: To protect the high-fitness chromosomes and improve the speed for the GA converges to the global optimum, we adopt the following adaptive crossover operator proposed in [16] with a crossover probabilities P_{c1} and P_{c2}.

$$\left\{ \begin{array}{c} [x_n, y_n] \\ [x_m, y_m] \end{array} \right\} \xrightarrow{crossover} \left\{ \begin{array}{c} [x_n, y_m] \\ [x_m, y_n] \end{array} \right\} \tag{11}$$

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{max} - f_{avg}}, & f' \geq f_{avg} \\ P_{c1}, & f' < f_{avg} \end{cases} \tag{12}$$

Where f_{avg} is the current average fitness of the population, f' is the larger fitness of two chromosomes selected to perform the crossover.

3) Select: In this step, we first calculate the fitness function $f(I)$ for each of the chromosomes I again. Find the worst 20% chromosomes in terms of fitness. Then replace them with the chromosome that has the highest fitness.

Once reach the maximum number of iterations $iter\_max$, return the chromosome I^* that has the highest fitness in the population as the final solution $D = I^*$ for the backhaul-aware placement of drone-cells. Currently, there is no theory on algorithm parameters that applies to all GA applications. The parameters selected by the GA should be based on the specific problem being solved. After a bunch of trails and tests, we find that the combination of the algorithm parameters in table I can deliver the best performance in terms of probability of finding the global optimum by the GA.

TABLE I
GA PARAMETER VALUES

Parameters	Values
T	500
α	0.99
P_m	0.2
P_{c1}	0.4
P_{c2}	0.2

However, the backhaul-aware placement of drone-cells could be a multi-peak problem with lots of local optimums. As shown in Fig. 3(a), an example shows that the probability of finding the global optimum by the GA runs 50 times is only 12%, for the backhaul-aware placement problem of two drone-cells with 1 GBS, $iter\_max = 2000$ and the population size $\mathcal{P}_{size} = 100$. Fig. 3(a) also shows that the placement problem is a multi-peak problem with many local optimums. To this situation, we now introduce the 'restart strategy', which is very simple but surprisingly effective for improving the probability of finding the global optimum by the GA to solve the backhaul-aware placement problem, without increasing the computation complexity.

Restart strategy: instead of running the GA with a large number of iterations, we now run the GA with a much smaller number of iterations, but restart the GA for a number of times N_{start}. For instance, instead of running the GA with 2000 iterations for the example in Fig. 3(a), we now let $iter\_max = 50$, and restart the GA for 40 times, while all the other parameters such as $\mathcal{P}_{size}$ remain same. Note that, the total number of iterations calculated is $50 \times 40 = 2000$. Thus the total computation complexity remains unchanged.

Reference [17] indicates that a restart strategy is a very economical approach for hard computational problems. Expressly, the restart strategy acknowledges that we do not have a particularly effective solution to multi-peak problems. Compare with other complex jump-out strategies, a simple restart strategy may be a more practical solution for avoiding local optimums, especially when the problem is difficult, and the success probability for finding the global optimum is minimal. Another advantage of the restart strategy is that it does not need to make any changes to the algorithm and reinitialize it. As shown in Fig. 3(b), the probability of finding the global optimum has been increased to 92% by restarting the GA with $iter\_max = 50$ for 40 times, for solving the same placement problem in Fig. 3(a). In the following part of this paper, all the GA we mentioned are GA with restart strategy. Notably, applying the proposed GA to solve the backhaul-aware placement problem for more drone-cells requires a larger $\mathcal{P}_{size}$ and/or N_{start}, to guarantee finding the global optimum.

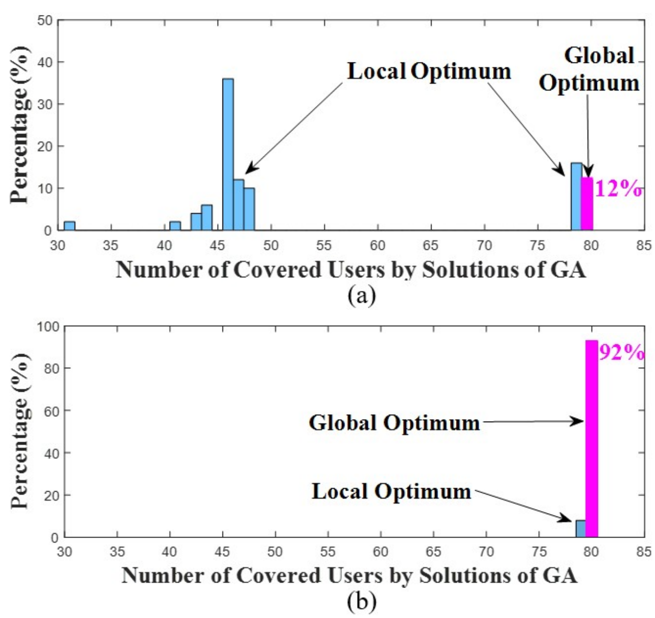

Fig. 3. Distribution and percentage of global optimum versus local optimums finds by the GA without restart strategy (a) and with restart strategy (b), each runs for 50 times.

IV. Simulation Results

In this section, the performance of the proposed exhaustive search algorithm and the GA with the restart strategy is evaluated using Matlab. We simulated the deployment of a team of drone-cells in a 10km $\times$ 10km quadrangle area with only one operational GBS with its coordinates G. The locations of the users are generated by the fore-mentioned doubly Poisson *matérn* cluster process, with the clustered user density of λ_c and the non-clustered user density of λ_{nc}. The number of user clusters is N_c. For simplicity, all user clusters are set to be disk-like, with random radii no larger than R_c. The coverage radius of the GBS and the drone-cell are R_g and R_d, respectively. The limited backhaul link distances between a GBS and a drone-cells and between two drone-cells are r_1 and r_2, respectively. The parameters of the GA are set the same as TABLE I. The detailed simulation parameters are as shown in TABLE II.

TABLE II
SIMULATION PARAMETER VALUES

Parameters	Values	Parameters	Values
G	(0,0)	λ_c	9e-5 users/m^2
λ_{nc}	3e-6 users/m^2	N_c	15
R_c	250 m	R_d	500 m
R_g	1000 m	r_1	3500 m
r_2	3000 m		

We first apply our proposed exhaustive search (ES) algorithm and the GA to find the optimal backhaul-aware placements in a 10km $\times$ 10km quadrangle area with the same user distribution for 1,2,3 and 4 drone-cells, respectively. The computing times for both algorithms are compared in TABLE III. Where m is the number of drone-cells, T_{ES} is the computing time of ES, T_{GA} is the computing time of GA, P_{save} is the percentage of the computing time saved by GA, $\mathcal{P}_{size}$ is the population size of GA, N_{start} is the number of restart of GA, P_{Global} is the success rate for the GA to find the global optimal placement over 20 runs. $iter\_max$=50 for all the examples. The simulations were run by MATLAB 2019b installed on an HP desk computer with the 7th generation Intel Core i7 processor and 16GB RAM.

As shown in TABLE I, the computing time of the ES increased dramatically from 0.78 seconds to 276970 seconds, with the number of drone-cells increases from 1 to 4. Meanwhile, the computing time of the GA increased with much less speed. Consequently, the percentage of computing time saved by GA rises significantly. In particular, the proposed GA can save the computing time up to 99.927%, with the success rate of $P_{Global} = 95\%$ over 20 runs for the example of 4 drone-cells. Moreover, the trend of P_{save} in TABLE III suggests that a higher percentage of the computing time saved by GA could be achieved when solving the placement problem with more drone-cells. These results verified the computational efficiency of the proposed GA. The corresponding optimal backhaul-aware placements of the four examples finds by the GA are as shown in Fig. 4, where the black circles are the coverage of the drone-cells. The GBS is located at the origin. The red dash circles denote the limited backhaul link range. Specifically, A drone can only have backhaul links if it is within the red dash circle of the GBS or another drone-cell. As indicated in Fig. 4, some of the drone-cells have to be placed at the edge of the red dash circles to cover a maximum number of users.

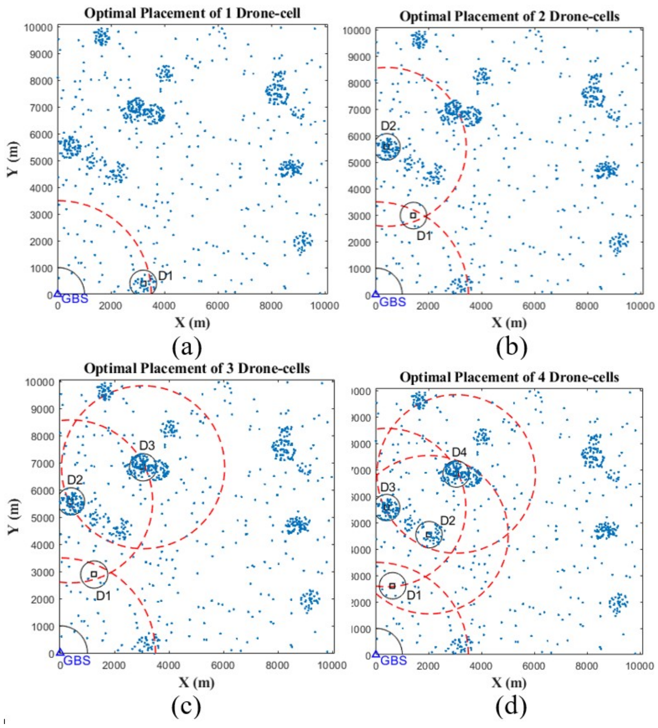

Fig. 4. Optimal backhaul-aware placements finds by the GA for 1 drone-cell (a), 2 drone-cells (b), 3 drone-cells (c), and 4 drone-cells (d).

TABLE III
COMPUTING TIME COMPARISON

m	T_{ES} (s)	T_{GA} (s)	P_{save}	$\mathcal{P}_{size}$	N_{start}	P_{Global}
1	0.78	0.82	-5.13%	20	30	100%
2	15.4	8.1	47.4%	50	50	100%
3	1449	23.3	98.39%	100	60	100%
4	276970	202	99.927%	200	200	95%

Fig. 5 indicates the success rate P_{Global} for the GA to find the global optimal placement of 3 drone-cells with different population size $\mathcal{P}_{size}$ and number of restart N_{start}. $iter\_max = 50$ for all the examples. According to Fig. 5, for different values of N_{start}, P_{Global} does not increase with the population size $\mathcal{P}_{size}$ increases. However, when N_{start} enhances, P_{Global} does increase accordingly, for different $\mathcal{P}_{size}$. Therefore, in terms of improving P_{Global}, increasing N_{start} would be a more efficient choice than increasing $\mathcal{P}_{size}$.

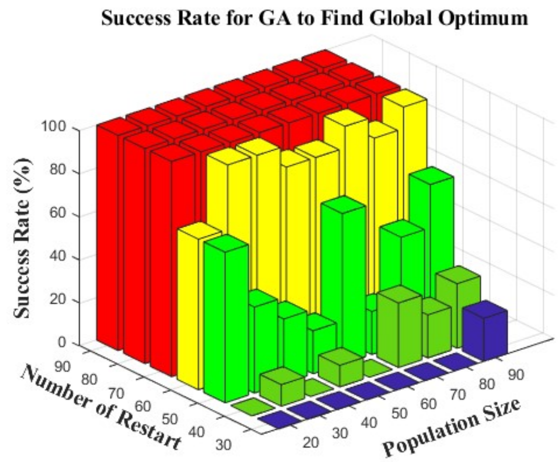

Fig. 5. The success rate for finding the global optimum by the GA with different population size $\mathcal{P}_{size}$ and number of restart N_{start}.

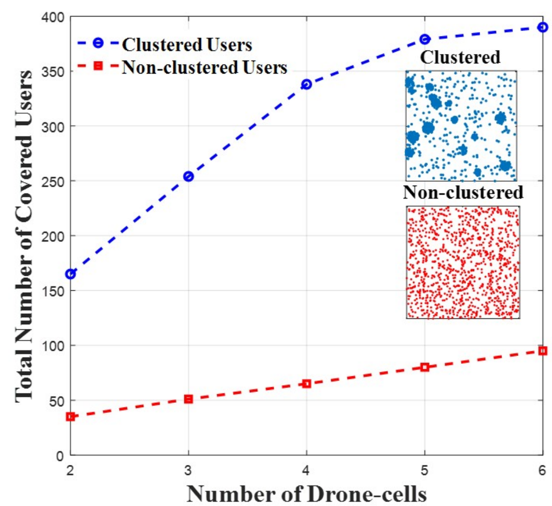

Fig. 6. The maximum number of users can be covered with a different number of drone-cells, for clustered and non-clustered user distribution.

Finally, we investigate the influence of different user distribution models to the optimal backhaul-aware placement results. Specifically, by changing the clustered user density of λ_c and the non-clustered user density of λ_{nc}, we simulated two user distribution scenarios: clustered and non-clustered. The total number of users and all the other parameters are set to be the same for both scenarios. Then we run the ES and GA to find the optimal backhaul-aware placement for a different number of drone-cells. We then compared the total number of users that can be covered N_{total} under the optimal placements, as shown in Fig. 6. It can be seen from Fig. 6 that a certain number of drone-cells can always cover significantly more users for the scenario with clustered user distribution than the one with non-clustered user distribution. Moreover, the increasing speed of the N_{total} decreased when adding the number of drone-cells m for the clustered scenario, while N_{total} increased with a constant speed as m increases for the non-clustered scenario. Since we have verified the high efficiency and accuracy for the proposed ES and GA to find the optimal placement. The slow down of N_{total}'s increasing speed could be contributed to that the user clusters with more users were effectively identified by the proposed algorithms, so that they have the priority to be covered by the first several drone-cells. The user clusters covered lately tend to have fewer users. Consequently, the N_{total}'s increasing speed reduced when adding more drone-cells to the network. For non-clustered scenario, the effectiveness of proposed algorithms tend to be less significant, because N_{total} is mainly decided by m, not the optimal placements, as long as constraints (7), (8) are satisfied. Thus, our proposed algorithms are more suitable for solving the drone-cell's optimal backhaul-aware placement problem for the scenario with clustered users.

V. Conclusion

In this paper, we considered the optimal placement problem of a team of drone-cells with limited backhaul communication distance, aiming at maximizing the total number of users covered by the drone-cells. We proposed two approaches, exhaustive search algorithm and a computationally efficient GA to find the optimal 2D backhaul-aware placements of multiple drone-cells. We also introduced a restart-strategy that helps proposed GA to avoid local optimums. Simulations show that the proposed GA can significantly save the computing time compare with the exhaustive search algorithm and the restart-strategy is verified to be a simple but very effective technique that significantly increases the success rate for the GA to find the global optimum. The proposed method relies on accurate location information of users. One future research direction is to extend the current method to the case where more precise location estimation can be acquired by some estimation tools such as robust Kalman filter [18]–[21]. The applications of the proposed method is not limited to drone-cells, another future research direction is to apply the current method to drones for surveillance purposes [22]–[24].

References

[1] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3d drone placement in 5g+ wireless networks," in *2017 IEEE international conference on communications workshops (ICC workshops)*. IEEE, 2017, pp. 109–114.

[2] 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 *2016 IEEE international conference on communications (ICC)*. IEEE, 2016, pp. 1–5.

[3] 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*, vol. 7, no. 6, pp. 994–997, 2018.

[4] A. V. Savkin and H. Huang, "Deployment of unmanned aerial vehicle base stations for optimal quality of coverage," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 321–324, 2018.

[5] H. Huang and A. V. Savkin, "A method for optimized deployment of unmanned aerial vehicles for maximum coverage and minimum interference in cellular networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2638–2647, 2018.

[6] X. Li, "Deployment of drone base stations for cellular communication without apriori user distribution information," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 7274–7281.

[7] X. Li and L. Xing, "Optimal deployment of drone base stations for cellular communication by network-based localization," in *2018 37th Chinese Control Conference (CCC)*. IEEE, 2018, pp. 7282–7287.

[8] M. Coldrey, U. Engström, K. W. Helmersson, M. Hashemi, L. Manholm, and P. Wallentin, "Wireless backhaul in future heterogeneous networks," *Ericsson Review*, vol. 91, pp. 1–11, 2014.

[9] A. Fouda, A. S. Ibrahim, I. Guvenc, and M. Ghosh, "Uav-based in-band integrated access and backhaul for 5g communications," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–5.

[10] R. K. Ganti and M. Haenggi, "Interference and outage in clustered wireless ad hoc networks," *IEEE Transactions on Information Theory*, vol. 55, no. 9, pp. 4067–4086, 2009.

[11] A. V. Savkin and H. Huang, "Range-based reactive deployment of autonomous drones for optimal coverage in disaster areas," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.

[12] H. Huang, A. V. Savkin, M. Ding, and M. A. Kaafar, "Optimized deployment of drone base station to improve user experience in cellular networks," *Journal of Network and Computer Applications*, vol. 144, pp. 49–58, 2019.

[13] B. M. Chazelle and D.-T. Lee, "On a circle placement problem," *Computing*, vol. 36, no. 1-2, pp. 1–16, 1986.

[14] H. Holland John, "Adaptation in natural and artificial systems," *Ann Arbor: University of Michigan Press*, 1975.

[15] D. Adler, "Genetic algorithms and simulated annealing: A marriage proposal," in *IEEE International Conference on Neural Networks*. IEEE, 1993, pp. 1104–1109.

[16] M. Srinivas and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 24, no. 4, pp. 656–667, 1994.

[17] B. A. Huberman, R. M. Lukose, and T. Hogg, "An economics approach to hard computational problems," *Science*, vol. 275, no. 5296, pp. 51–54, 1997.

[18] X. Kai, C. Wei, and L. Liu, "Robust extended Kalman filtering for nonlinear systems with stochastic uncertainties," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 40, no. 2, pp. 399–405, 2009.

[19] A. V. Savkin and I. R. Petersen, "Model validation for robust control of uncertain systems with an integral quadratic constraint," *Automatica*, vol. 32, no. 4, pp. 603–606, 1996.

[20] L. El Ghaoui and G. Calafiore, "Robust filtering for discrete-time systems with bounded noise and parametric uncertainty," *IEEE Transactions on Automatic Control*, vol. 46, no. 7, pp. 1084–1089, 2001.

[21] V. Malyavej and A. V. Savkin, "The problem of optimal robust Kalman state estimation via limited capacity digital communication channels," *Systems & Control Letters*, vol. 54, no. 3, pp. 283–292, 2005.

[22] A. V. Savkin and H. Huang, "Asymptotically optimal deployment of drones for surveillance and monitoring," *Sensors*, vol. 19, no. 9, p. 2068, 2019.

[23] H. Huang and A. V. Savkin, "Towards the internet of flying robots: A survey," *Sensors*, vol. 18, no. 11, p. 4038, 2018.

[24] A. V. Savkin and H. Huang, "Proactive deployment of aerial drones for coverage over very uneven terrains: A version of the 3d art gallery problem," *Sensors*, vol. 19, no. 6, p. 1438, 2019.