Ant System Inspired Heuristic Optimization of UAVs Deployment for k-Coverage Problem

Krzysztof Trojanowski, Artur Mikitiuk, and Jakub Grzeszczak(✉)

Cardinal Stefan Wyszyński University in Warsaw, Wóycickiego 1/3, 01-938 Warsaw, Poland
{k.trojanowski,a.mikitiuk,jakub.grzeszczak}@uksw.edu.pl

Abstract. When ad-hoc connectivity for a group of ground users has to be delivered, one can use a network of Unmanned Aerial Vehicles (UAV) equipped with Mobile Base Stations (MBS). In this research, we minimize the number of UAVs by effectively deploying UAVs over the zone where users are located. The proposed model divides zone into sectors of different areas and shapes depending on users' location and the ranges of MBSs. Deployment of UAVs in sectors is optimized by a method inspired by the Ant System approach and extended by a new problem-specific heuristic. We propose a new set of benchmark problems, called SCP2, for simulations. Simulation results show the algorithm's efficiency and reveal the most beneficial values of the algorithm's parameters.

Keywords: Unmanned Aerial Vehicles · Ant Systems · k-Coverage Problem

1 Introduction

A network of Unmanned Aerial Vehicles (UAV) equipped with Mobile Base Stations (MBS) can provide communication services for ground users in the case of disaster or festive areas management, military operations, or any other scenarios where ad-hoc connectivity is needed. The performance of the MBSs service depends on the locations of users and UAVs; thus, it can be a subject of optimization. In this research, we focus on minimizing the number of UAVs when establishing new connectivity for a group of users in a given zone. The number is minimized by the effective deployment of UAVs over the zone.

In our research, the optimization problem is stationary; that is, ground users represent, for example, routers placed in crucial locations which deliver WLAN service to the surrounding receivers. Hence, ground users can be regarded as immobile and previously known to the UAV swarm, so the swarm does not have to adapt over time to the changing positions of users.

Numerous publications address different variants of the problem of deploying UAVs for optimal wireless coverage. In [5], the authors present a method minimizing the number of MBs covering a set of immobile ground terminals with known

R. Wyrzykowski et al. (Eds.): PPAM 2022, LNCS 13826, pp. 414–428, 2023.
https://doi.org/10.1007/978-3-031-30442-2_31

locations on the horizontal plane. The method applies a spiral algorithm building its solution by placing the MBSs sequentially until the total coverage is satisfied. In [2], the authors use K-means clustering and a stable marriage approach to partition users into clusters and find 2-D coordinates of UAVs first. Then, the third coordinate, altitude, is optimized using search space-constrained exhaustive search and particle swarm optimization (PSO). In [8], the authors also use 3-D coordinates to represent UAVs' locations. The proposed method finds optimal deployment of UAVs iteratively invoking a clustering algorithm K-means and one of the population heuristics: Particle Swarm Optimization, Genetic Algorithm, or Artificial Bees Colony. In [6], the authors propose a problem-specific heuristic optimization method working on a discrete representation of a UAV location, where nodes of a square grid of $L \times L$ are considered for discrete locations. In [1], UAVs ensure connectivity for the people uniformly and randomly distributed over the area in previously unknown locations. The authors assume that the number of UAVs may be insufficient to cover the entire area, and initially, some users may not be in range. Hence, to achieve higher coverage, UAVs have to move over time. Such a network of UAVs offers intermittent coverage to the users. Therefore, its primary aim is to receive messages from users and route them to neighbor UAVs closer to the gateway or the gateway in range. The approach maximizes average people-to-drone connected time and the percentage of people in the communication range of UAVs for two cases: without and with mobility of UAVs. In [11], the authors maximize users' coverage probability using particle swarm optimization to find optimal locations of UAVs in 3-D space. A particle in a swarm represents coordinates for an entire group of N UAVs, which is a real-valued vector of $3N$ dimensions. In [10], the aim is to maximize the number of users covered by the net of UAVs without losing network connectivity in urban disaster scenarios where the area consists of streets, parks, and buildings. Users move randomly, and UAVs deploy over the area using the tactical movement generation rules based on the Jaccard distance and artificial intelligence algorithms. In [4], the authors notice that due to the limited maximum distance of UAVs flight, the emergency network should consist of mobile base stations and terrestrial, portable base stations, and solve the problem of optimal deployment for the network consisting of base stations of two types.

We propose a model where the zone is divided into sectors of different areas and shapes depending on users' location and non-uniformly distributed inside the given zone and the range of MBSs. We do not need to find precise coordinates of UAV locations because any location within a sector has the same impact on the connectivity coverage. One or multiple UAVs can occupy every single sector. In this model, the deployment of a minimal number of UAVs represents a combinatorial problem. Due to its complexity, we apply a heuristic optimization method inspired by the Ant System approach but extended by a new problem-specific heuristic aimed at generating a single solution. The proposed zone model defines a new structure of a pheromone matrix and new rules of the pheromone deployment. For experimental verification, we created a set of benchmark problems and conducted experiments to show the algorithm's efficiency and reveal the most beneficial values of the algorithm's parameters.

The paper consists of five sections. The wireless communication system, its model, its hypergraph representation, and optimization criteria are described in Sect. 2. Section 3 presents an Ant System-inspired approach to optimize the deployment of UAVs. The experimental part of the research is described in Sect. 4. Section 5 concludes the paper.

2 The Optimization Problem

We optimize the number of UAVs equipped with base stations and their locations to provide connectivity for n immobile ground users in the given zone. We assume that all UAVs and MBSs have the same functionality and parameters. Moreover, a third-party entity provides connectivity to the UAVs, so their distances have no meaning for the network functionality. There are also some other simplifications in our model of the problem. The model lacks radio resource management, interference management, channel estimation, prediction, or energy efficiency. We also assume that all UAVs fly at the same altitude, offering the best compromise between flight safety and productivity. Due to the homogeneity of MBSs transmitters, the round areas with satisfying connectivity offered by MBSs have the same size. In formal terms, we consider a set of n immobile ground users $V = \{v_1, \ldots, v_n\}$, and a set of p MBSs $M = \{m_1, \ldots, m_p\}$. The connectivity radius of each MBS is r. We want to ensure a k-coverage for each user in V, that is

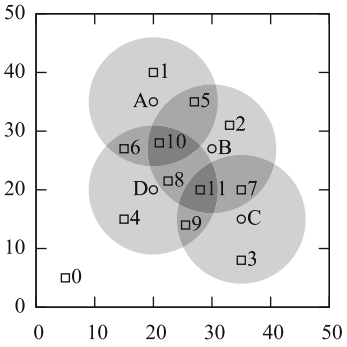
$$\forall v \in V \ \exists \mathbf{c} \in [M]^k \text{ such that } \forall m \in \mathbf{c} \ dist(v, m) \leq r, \tag{1}$$

where $\mathbf{c}$ indicates the set of k MBSs covering the ground user v, $[M]^k$ is the set of all the subsets of M with exactly k elements, and $dist(v, m)$ denotes the distance between the ground user v and the MBS m. Our optimization goal is to minimize the number of MBSs p.

For a continuous 2-D representation of the ground users' and MBSs' coordinates, the problem can be formulated as the Geometric Disk Cover (GDC) problem [9]. In GDC, we minimize the number of disks of a given radius covering a set of immobile ground users, which is an NP-hard problem. When we use a discrete model, where the area is a grid of small rectangular cells, the problem of the effective deployment of UAVs can be similar to generating an overview image. Cells of the observation areas represent regions of ground users' locations. For this model, the problem of minimizing the number of UAVs can be formulated as ILP (integer linear programming) problem [7].

2.1 The Model of a Wireless Communication System

The communication system has to ensure a k-coverage for each user, which means that each user requires at least k MBSs available in its connectivity range. Our goal is to minimize the number of UAVs by optimizing their locations.




```
UAVs covering          users in the range
users:                 of UAVs:
A [1,5,6,10]           0 []
B [2,5,7,8,10,11]      1 [A]
C [3,7,9,11]           2 [B]
D [4,6,8,9,10,11]      3 [C]
                       4 [D]
                       5 [A,B]
                       6 [A,D]
                       7 [B,C]
                       8 [B,D]
                       9 [C,D]
                       10 [A,B,D]
                       11 [B,C,D]
```

Fig. 1. Example of an operating wireless network with four users: A, B, C and D (circles) and 12 UAVs: $0, 1, 2, \ldots, 11$ (squares)—on the left, and lists of UAVs covering users and users in the range of UAVs—on the right

Figure 1 depicts an example deployment of 12 UAVs over the four users. Gray circles around users have a radius equal to the connectivity radius of MBSs carried by UAVs. Thus, locating MBS wherever in the circle guarantees connectivity to the user in the circle's center.

The circles divide the zone into sectors. Each sector represents a different set of users, which we can cover by MBS service. The shades of gray of the sectors represent the number of covered users; the darker sector, the more users. In the example, the UAVs are placed alone in each sector. For example, UAV no. 10 hovers in the sector where its MBS covers users A, B, and D. MBS of UAV no. 11 covers users B, C, and D. MBS of UAV no. 5—users A and B. And so on. The sector of UAV no. 0 is white, which means there are no users in its range. The precise coordinates of a UAV location have no meaning as long as it remains entirely in the respective sector.

2.2 Hypergraph Representation of the System

The example presented in Fig. 1 can be modeled as a hypergraph $H = (V, E)$ where V is a set of nodes (ground users), and E is a set of non-empty subsets of V (sectors of the zone) called hyperedges. There are four nodes and 11 hyperedges in this example. Let us label the hyperedges according to the UAVs' IDs and nodes according to the users. For example, the hyperedge no. 10 connects nodes A, B, and D. The hyperedge no. 11—nodes B, C, and D. The hyperedge no. 5—nodes A and B. And so on.

2.3 Representation of Solution

A solution **s** represents a set of UAVs assigned to the zone sectors, where each sector may contain zero, one, or more UAVs. In particular, the sector may have no UAV assigned when UAVs from other sectors deliver connectivity. On the

other side, there is no limit to the number of UAVs in one sector. Formally, $\mathbf{s} = \{e_i, e_j, \ldots, e_q\}$ where values $e_i, e_j, \ldots, e_q$ identify zone sectors (hyperedges in the hypergraph) where UAVs are located. The solution $\mathbf{s}$ is a so-called *multiset*. That is, some of the values in $\mathbf{s}$ may appear more than once if there is more than one UAV in the sector. When the sector identifier is absent in $\mathbf{s}$, it means that no UAV is needed in this sector.

Since multiple elements in $\mathbf{s}$ can have the same values, the operator $\cup$ applied in, for example, the expression $\mathbf{s} \cup \{e\}$ always adds a new element e to $\mathbf{s}$ even if an element with such value already exists in $\mathbf{s}$.

2.4 The Optimization Criteria

The optimization aims to minimize the number of UAVs over the zone while ensuring all ground users' connectivity parameters. Thus, the value of a solution is proportional to the number of UAVs in the network. Moreover, we want to avoid UAV overcrowding in sectors. Therefore, when we have two solutions containing the same number of UAVs, we also consider the diversity of the UAVs' distribution over sectors. The fewer UAVs occupying the same sectors, the better. As such, the fitness function f is evaluated simply as Eq. 2, but a secondary function (Eq. 3) can be used to pick between two similar in-length solutions.

$$f(\mathbf{s}) = len(\mathbf{s}) \tag{2}$$

$$f(\mathbf{s}) = len(\mathbf{s})/set(\mathbf{s}) \tag{3}$$

where $len(\cdot)$ returns the number of the hyperedge IDs in the solution, that is, the number of UAVs in the network, and $set(\cdot)$—the number of unique hyperedge IDs in $\mathbf{s}$. When the assignment of UAVs to hyperedges is unique, that is, each UAV occupies a different sector, the penalty component $len(\mathbf{s})/set(\mathbf{s})$ equals one. Otherwise, it is greater than one and rises as the uniqueness falls.

The fulfillment of sufficient connectivity conditions depends on the number of UAVs in the user vicinity. Every user needs access to at least k MBSs simultaneously. Otherwise, the solution is unfeasible. Therefore, we call this a k-coverage problem. For the example given in Fig. 1, k can be equal at most four because users A and C have four MBSs in their ranges, the lowest level of coverage among all users. Hence, the proposed deployment of UAVs can also represent a feasible solution for the k-coverage problem where $k = 1, 2, 3, 4$. One can notice that for $k = 4$, the deployment is feasible but not optimal because there exist deployments of fewer UAVs also delivering connectivity for $k = 4$.

3 The Optimization Method

For the aim of optimization, we apply an iterative heuristic approach inspired by the Ant Systems. Ant Systems have two main distinguishing characteristics which separate them from other heuristics, e.g., evolutionary or swarm approaches. First is a pheromone matrix containing trails left by artificial ants

when they create new solutions. The other is that ants are not representatives of solutions improved over subsequent iterations. Construction of solutions with respect to already deployed pheromone trails is the only job for ants. It is important to stress that ants are not information transmitters between iterations; the only information transferred is the pheromone information.

Typically, an ant constructs a solution by a sequence of probabilistic decisions. Every decision extends a partial solution by adding a new component to the solution until a complete solution is derived. The sequence of decisions can be viewed as a path through a corresponding decision graph, so ants find paths through the graph that correspond to reasonable solutions. Ants that have found reasonable solutions can mark the edges of the corresponding path in the graph with an artificial pheromone. This pheromone guides ants in the next iteration. The paths can improve in subsequent iterations due to the pheromone indicating beneficial decisions of ants.

In our problem model, a solution represents an assignment of UAVs to the hyperedges of the hypergraph. Therefore, we update the pheromone trail in the hyperedges contributing to the solution, and the update is inversely proportional to the solution's length. To ensure that the pheromone from older iterations does not influence the following iterations for too long, some percentage of the pheromone evaporates during an update step. Algorithm 1 presents the generic scheme of the Ant System.

Algorithm 1

```
1: Initialize pheromone values
2: repeat
3:     for all ant_k do
4:         construct k-th solution                          ▷ 1. ants find their paths
5:     for all pheromone values do
6:         decrease the value by a certain percentage       ▷ 2. evaporation
7:     for all pheromone trails contributing to solutions do
8:         increase the value   ▷ 3. intensification: ants pheromone is laid
9: until termination condition met
```

3.1 The Problem–Specific Step: Generation of a Solution

Each ant constructs one solution. The constructing method is a heuristic using problem-specific knowledge about the hypergraph representation of the system. The hypergraph H, elite set of hyperedges $\mathbf{e}_{\text{elit}}$, and the required coverage level k are the input of the method.

The set $\mathbf{e}_{\text{elit}}$ consists of hyperedges having a large number of nodes since they are regarded as the most efficient for covering. For each $v \in V$, we do the following two steps: For all hyperedges containing v, we calculate their cardinality, which is the number of vertices in the hyperedge. Then, all the hyperedges

having the highest cardinality become $\mathbf{e}_{\text{elit}}$ members. The pseudocode of the method generating $\mathbf{e}_{\text{elit}}$ is presented in Algorithm 2.

Algorithm 2

```
1: function GENERATEELITESET(H)
2:         ▷ Input: hypergraph H          ▷ Output: elite set of hyperedges e_elit
3:     e_elit ← ∅                          ▷ create an empty elite set of hyperedges e_elit
4:     for all v ∈ V do
5:         mc_v ← max_{e|v∈e}(card(e))     ▷ find the max cardinality of e among e
   containing v
6:         e_elit ← e_elit ∪ {e|(v ∈ e) ∧ (card(e) = mc_v)}
7:     return e_elit
```

The complexity of the for loop in this algorithm is $O(d_v)$, where d_v is the degree of vertex v, that is, the number of hyperedges v belongs to. The whole algorithm has complexity $O(n*d_{\max})$, where $d_{\max}$ is the maximum vertex degree in the hypergraph.

The method generating a solution $\mathbf{s}$ consists of five steps. The pseudocode of this method is presented in Algorithm 3. Please note, that the solution $\mathbf{s}$ is a multiset of hyperedge identifiers representing respective locations of UAVs (one identifier represents location of one UAV), whereas $\mathbf{e}_{\text{elit}}$ and $\mathbf{e}_{\text{init}}$ are regular sets of hyperedges.

#1 Stochastic selection of an initial set of hyperedges $\mathbf{e}_{\text{init}}$ among the hyperedges in $\mathbf{e}_{\text{elit}}$. The chances of being selected as a candidate to $\mathbf{e}_{\text{init}}$ depend on the pheromone trails. However, the candidate is omitted when its recruitment does not extend the set of covered nodes. The selection stops as soon as no nodes remain uncovered. The complexity of this step is $O(|\mathbf{e}_{\text{elit}}| * c_{\text{emax}})$ where c_{emax} is the maximal cardinality of a hyperedge in $\mathbf{e}_{\text{elit}}$.

#2 Sequential deployment of UAVs in $\mathbf{e}_{\text{init}}$. For each of these hyperedges, we try to deploy new UAVs. We assign as many new UAVs as necessary to guarantee the requested level k of coverage for all the nodes joined by this hyperedge. We process hyperedges in the same order as they were put into $\mathbf{e}_{\text{init}}$ in Step #1. The UAV assignment is asynchronous. It means that the coverage of nodes by UAVs already assigned is considered when adding the next ones. For $|\mathbf{e}_{\text{init}}| * c_{\text{imax}}$ nodes, where c_{imax} is the maximal cardinality of a hyperedge in $\mathbf{e}_{\text{init}}$, we have to verify whether these nodes have k-coverage by the UAVs already declared in $\mathbf{s}$. It takes $O(k * |\mathbf{e}_{\text{init}}|)$ operations. When the coverage is insufficient, we add additional UAVs to $\mathbf{s}$, which takes $O(k)$ operations. Thus, the complexity of this step is $O(k * |\mathbf{e}_{\text{init}}|^2 * c_{\text{imax}}))$.

#3 Dispersion of UAVs over the hyperedges in their neighborhood. All generated UAVs are shifted to a random adjacent hyperedge. We consider a hyperedge to be adjacent if it connects all but one of the nodes connected to a previous one. If no such hyperedges exist, the UAV is removed from the solution. In

Algorithm 3

function BUILDANEWSOLUTION(H, $\mathbf{e}_{\text{elit}}$, k)
▷ Input: hypergraph H, elite set of hyperedges $\mathbf{e}_{\text{elit}}$, k ▷ Output: $\mathbf{s}$

$V' \leftarrow V$ ▷ create a set of uncovered nodes V'
$\mathbf{e}_{\text{init}} \leftarrow \emptyset$ ▷ create an empty initial set of hyperedges $\mathbf{e}_{\text{init}}$
$\mathbf{s} \leftarrow \emptyset$ ▷ create an empty solution $\mathbf{s}$

repeat ▷ *Step #1*: ———— select hyperedges to the initial set
  randomize $e \in \mathbf{e}_{\text{elit}}$ ▷ select randomly w.r.t. pheromone trail levels
  if $\{V' \cap e\} \neq \emptyset$ **then** ▷ if any node in e remains uncovered
    $\mathbf{e}_{\text{elit}} \leftarrow \mathbf{e}_{\text{elit}} \setminus \{e\}$ ▷ e is removed from $\mathbf{e}_{\text{elit}}$
    $\mathbf{e}_{\text{init}} \leftarrow \mathbf{e}_{\text{init}} \cup \{e\}$ ▷ e is added to $\mathbf{e}_{\text{init}}$
    $V' = V' \setminus e$ ▷ all the nodes connected by e are removed from V'
until $V' = \emptyset$

for all $e \in \mathbf{e}_{\text{init}}$ **do** ▷ *Step #2*: —— build a preliminary version of the solution
  for all $v \in e$ **do** ▷ for all the nodes connected by e
    if v has l-coverage by the UAVs already declared in $\mathbf{s}$, where $l < k$ **then**
      add $(k - l)$ UAVs to the hyperedge e in $\mathbf{s}$

for all $e \in \mathbf{s}$ **do** ▷ *Step #3*: —— disperse UAVs to their neighbour hyperedges
  if $\mathcal{N}(e) \neq \emptyset$ **then**
    while there exist identifier e in $\mathbf{s}$ **do**
      randomize $e' \in \mathcal{N}(e)$ ▷ select randomly one of the neighbours
      replace e by e' in $\mathbf{s}$ ▷ move one UAV from the hyperedge e to e'
  remove all identifiers e from $\mathbf{s}$ ▷ remove UAVs from the hyperedge e

for all $e \in \mathbf{e}_{\text{init}}$ **do** ▷ *Step #4*: ———— repeat #2 to fix the solution
  for all $v \in e$ **do** ▷ for all nodes joined by the hyperedge e
    if v has l-coverage by the UAVs already declared in $\mathbf{s}$, where $l < k$ **then**
      add $(k - l)$ UAVs to the hyperedge e in $\mathbf{s}$

▷ *Step #5*: — remove redundant UAVs from the solution
divide hyperedges present in $\mathbf{s}$ into groups w.r.t. the number of joined nodes
label groups: $\{g^1(\mathbf{s}), g^2(\mathbf{s}), \ldots, g^m(\mathbf{s})\}$ according to the number of joined nodes
for $i \leftarrow 1$ to m **do** ▷ start with groups of hyperedges joining least nodes
  for all $e \in g^i(\mathbf{s})$ **do** ▷ take the hyperedges in $g^i(\mathbf{s})$ in a random order
    while all $v \in e$ have coverage higher than k **do**
      remove identifier e from $\mathbf{s}$ ▷ remove one redundant UAV

return s ▷ *Finish*: ———— return the obtained solution $\mathbf{s}$

this step, the inner loop (while) requires $O(k * |\mathbf{e}_{\text{init}}|)$ iterations and every iteration has cost $O(1)$. The outer loop (for) also requires $O(k * |\mathbf{e}_{\text{init}}|)$ iterations. Therefore, the complexity of this step is $O(k^2 * |\mathbf{e}_{\text{init}}|^2)$.

#4 Fixing the deployment of UAVs. Modifications introduced in Step #3 can make the solution unfeasible. Thus, we repeat Step #2 in this step to make the solution feasible again. The complexity of this step is the same as in Step #2, i.e. $O(k * |\mathbf{e}_{\text{init}}|^2 * c_{\text{imax}}))$.

#5 Removal of redundant UAVs. After Step #4, some UAVs can be redundant (the requested coverage level for all the nodes remains satisfied even without these UAVs). Therefore, we analyze groups of hyperedges regarding their cardinality, starting from one. Within each group, and in random order, we verify if the lack of any UAVs makes the solution unfeasible. If it does not, the redundant UAVs are removed from the hyperedge. Dividing hyperedges into groups has the complexity $O(k*|\mathbf{e}_{\text{init}}|)$. In the nested for loop, the while loop is invoked $k * |\mathbf{e}_{\text{init}}|$ times. A single while loop invocation is $O(c_{\text{emax}})$. Thus, the complexity of this step is $O(k * |\mathbf{e}_{\text{init}}| * c_{\text{emax}})$.

The whole Algorithm 3 has the complexity $O(|\mathbf{e}_{\text{elit}}|*c_{\text{emax}})+O(k*|\mathbf{e}_{\text{init}}|^2*c_{\text{imax}})+O(k^2 * |\mathbf{e}_{\text{init}}|^2)+O(k * |\mathbf{e}_{\text{init}}| * c_{\text{emax}})$. Since $|\mathbf{e}_{\text{init}}| \leq |\mathbf{e}_{\text{elit}}|$ and $c_{\text{imax}} \leq c_{\text{emax}}$, we can assess this complexity as $O(k * |\mathbf{e}_{\text{elit}}|^2 * (k + c_{\text{emax}}))$.

3.2 The Main Loop

The algorithm starts with the generation of the hypergraph $H = (V, E)$ from the input data with the area size and locations of the ground users. Then, the function GENERATEELITESET generates the set $\mathbf{e}_{\text{elit}}$. Next, we create a pheromone vector P of size $|\mathbf{e}_{\text{elit}}|$. For each of the hyperedges in $\mathbf{e}_{\text{elit}}$, the initial pheromone level in P equals the number of ants used by the algorithm.

Next, the main loop starts. The main loop of the algorithm corresponds to the one presented in Algorithm 1. In the beginning for each ant, we generate a new solution of UAVs deployment using the function BUILDANEWSOLUTION with arguments: H, $\mathbf{e}_{\text{elit}}$, and the requested level of coverage k. We then evaluate each new solution as the inverse of its length. Next, evaporation arises. Pheromone values for all hyperedges in $\mathbf{e}_{\text{elit}}$ are reduced by a fixed proportion ρ according to the formula: $P_e = (1 - \rho)P_e$. Finally, we update the pheromone trails. For each of the solutions, we want to reward those hyperedges whose membership in $\mathbf{e}_{\text{init}}$ gave a feasible solution. However, the reward value δ is inverse proportional to the solution length and equal to the solution score. Therefore, the new pheromone level for a hyperedge e is calculated according to the formula $P_e = P_e + \delta_e$. The coefficient δ_e is the sum of scores of those solutions, where e was a member of $\mathbf{e}_{\text{init}}$ and has assigned at least one UAV.

The main loop ends when the stopping condition is met, and then the best-found solution is returned.

4 Experiments

4.1 Benchmark

We evaluated the algorithm experimentally on the benchmark set of test cases called SCP2 [3]. The set consists of six classes of problems that differ in the number of ground users and their locations. Nodes of a rectangular grid over the square zone of size one unit define possible users' locations. There are two densities of grids with grid cell dimensions s_{grid} equal 0.04 by 0.04 or 0.02 by 0.02 units (676 or 2601 nodes in the zone, respectively). In every case, the number of users is smaller than the number of nodes: 100, 200, or 500. The users' locations are selected randomly among the nodes. However, we additionally shift the final user location from the node coordinates toward a random direction by a random distance smaller than 1.5 of s_{grid}. Two grids with different cell dimensions and three sizes of the ground users' set eventually give six classes of users' distribution. For every class, we generated 50 instances based on different locations of users on the grid and directions and distances of shifts.

The zone is divided into sectors as presented in the example Fig. 1. The MBS signal power defines the radius of circles around users, which we arbitrarily set to 0.05 units. Locations of sectors define the structure of hypergraphs obtained for each problem instance according to the rules described in Sect. 2.2. On average, hypergraphs representing problem instances in the two classes with 100 users have around 18.6 (with a range of 9 to 25) connected components for $s_{\text{grid}} = 0.02$ and 20 (13 to 27) for $s_{\text{grid}} = 0.04$. For the problem instances from the two classes with 200 users, we got the average of about 3.9 (from 1 to 10) connected components for $s_{\text{grid}} = 0.02$ and 3 (from 1 to 6) for $s_{\text{grid}} = 0.04$. When the number of users equals 500, the hypergraph always consists of one connected component. The decreasing number of connected components is not surprising because more intersections between users' surrounding areas occur when the number of users grows.

The last parameter of the problem is the minimum coverage level k, equal to 1, 2, 5, or 10. Eventually, we get a benchmark consisting of 24 classes of problems: six classes of users' distribution over the zone by four levels of the minimum coverage k.

4.2 Plan of Experiments

The algorithm has three parameters: the number of ants n_{ants}, evaporation coefficient ρ, and stopping condition parameter, that is, the maximum number of fitness function calls max_{nffc}. In the preliminary experiments, we observed that satisfying results were obtained for $\rho = 0.1$ and $n_{\text{ants}} = 10$. We set $max_{\text{nffc}} = 2000$, so an experiment takes 200 iterations. In the presented experiments, the coefficient ρ is also the subject of experimental tuning and varies from 0.1 to 0.5.

The experiments are divided into two groups. In the first group, we performed experiments with $\rho = 0.1$ for all 24 classes (six classes of SCP2 by four values of k). In the second group, we selected the two classes of problems that proved to be the most demanding in the first group of experiments: $s_{\text{grid}} \in \{0.02, 0.04\}$ for 500 ground users and $k = 10$. For these classes, we observed optimization progress for five different values of $\rho \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

We repeated experiments 32 times for every problem instance, excluded the best and worst results and calculated the mean number of UAVs for the remaining ones.

4.3 The Results

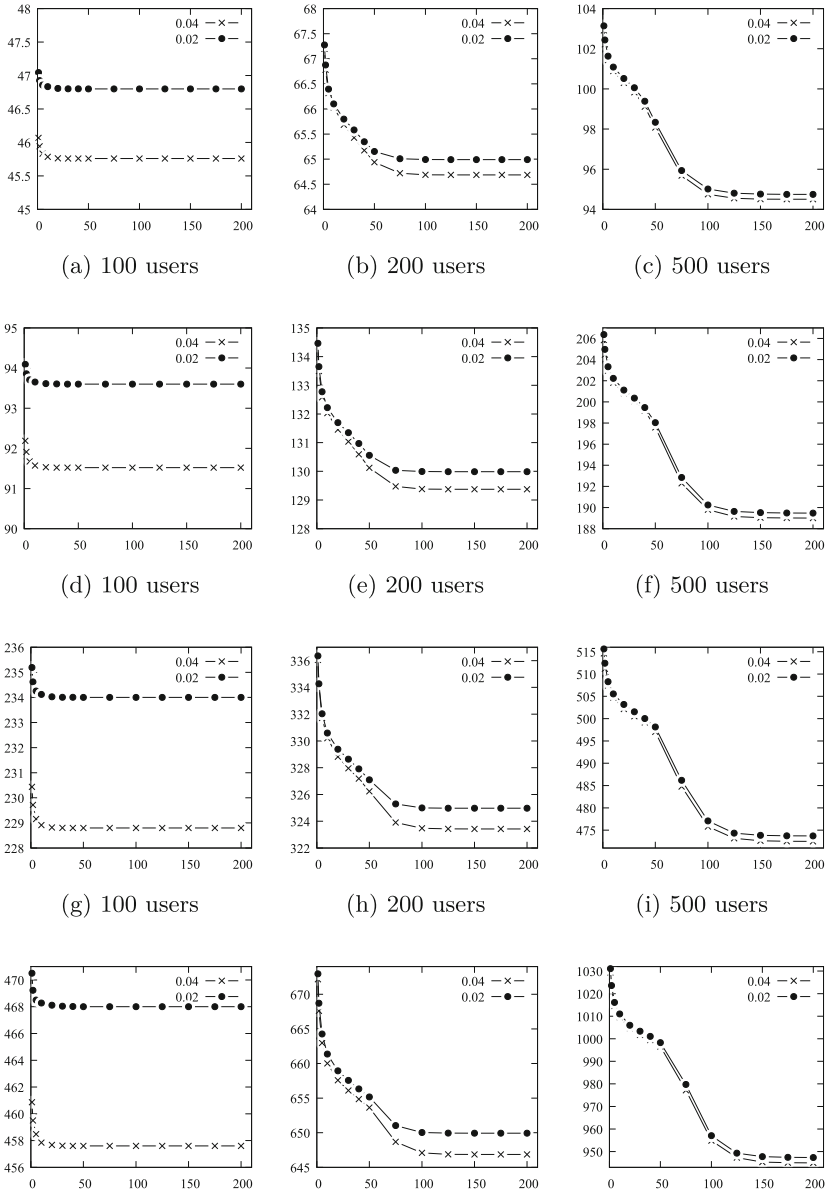
Figure 2 shows the results of our experiments with the benchmark described in Sect. 4.1. As the diversity level of UAVs' deployment serves only as a tie-breaker between two similar in-length solutions, our analysis concerns only the average number of UAVs for each class of the problem.

For smaller numbers of users in the area, which resulted in hypergraphs with plenty of connected components of smaller size, the proposed algorithm found itself near the optimal solution almost instantly. Even for the highest considered coverage of 10, it could not further improve the results after about a fifth of the given computational time. When the user population grows, the complexity of the corresponding hypergraph increases. Then our method needed much more time to stop improving.

Interestingly, the algorithm reached the suboptimal solutions after similar numbers of iterations regardless of the desired coverage level k. For classes of 100 users (Fig. 2a, 2d, 2g, 2j) this method reached the suboptimal solution within fewer than 10 iterations, barely improving for up until iteration 30. Classes with 200 users (Fig. 2b, 2e, 2h, 2k) showed steady improvement for about 70 iterations (700 evaluations), but further improvement stopped after about 100 iterations.

The most challenging classes of 500 users (Fig. 2c, 2f, 2i, 2l) almost doubled the time required to find suboptimal solutions, reaching as far as 150 iterations of varied improvement. The pace of this improvement also shows the ability of the algorithm's transition between different local optima, which initially slowed down the search for the best solution.

The results of the second group of experiments with different evaporation rates are presented in Fig. 3. One can see that a higher evaporation rate could significantly hasten the search process at the cost of a noticeable but relatively small tradeoff in the quality of the found solution.



(a) 100 users (b) 200 users (c) 500 users

(d) 100 users (e) 200 users (f) 500 users

(g) 100 users (h) 200 users (i) 500 users

(j) 100 users (k) 200 users (l) 500 users

Fig. 2. Mean numbers of UAVs for the best-found solutions For two densities of grids: with grid cell dimensions s_{grid} equal 0.04 by 0.04 and 0.02 by 0.02 units. For $k = 1$: (a), (b), and (c), $k = 2$: (d), (e), and (f), $k = 5$: (g), (h), and (i), and $k = 10$: (j), (k), and (l); X-axis represents the iteration number; $max_{\text{nffc}} = 2000$

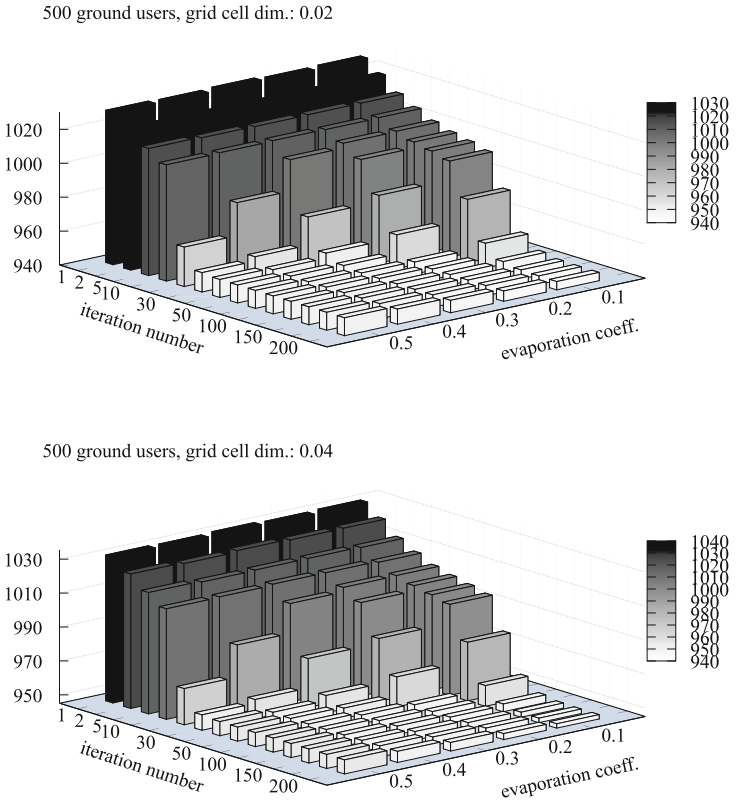


Fig. 3. Mean numbers of UAVs for the best-found solutions observed in selected iterations; $max_{\text{nffc}} = 2000$; evaporation coefficients: $[0.1, 0.2, 0.3, 0.4, 0.5]$, $k = 10$

5 Conclusions

In this paper, we proposed an Ant System-inspired algorithm for optimizing UAVs deployment for the k-Coverage Problem. The novelty of the presented approach lies in introducing a new model of the space where the network formed by UAVs serves the users and a problem-specific heuristic exploiting the model's features. Combining the Ant System method and the heuristic is also an innovative element.

We define a problem instance as a hypergraph of connections between network users, where each user corresponds to a single hypergraph's node. Hyperedges describe sectors of the problem area where a single UAV can connect simultaneously to a given set of users. The shape of each sector is defined by user positions and the range of a carried MBS.

We use the model-specific characteristics of the problem to construct an input space for the Ant System step of the algorithm and build new solutions using a proposed heuristic.

A generated solution is represented as a variable-length array of hyperedges, which can be translated to UAV positions within the problem area. We use the feedback loop of the algorithm to find a subset of the hyperedges that results in the shortest solutions while avoiding overcrowding within each sector. Among solutions of a given length, the one with the lowest average UAV count per sector within the solution is considered the best.

We tested our algorithm using three pairs of classes from the presented dataset. The number of nodes in each problem instance differed for each pair, and classes within pairs had different deployment characteristics. For each class, we repeated tests for four values of the required coverage level for a solution k—1, 2, 5, and 10.

The experiments' results correlate with some of the benchmark problem properties and reflect the expected behavior of the Ant System-based approaches. The density of the created hypergraph is closely related to the problem parameters and directly affects the time required to obtain good results. Similarly, choosing between different evaporation rates can hasten the search but at a slight cost to the quality of the obtained solutions.

References

1. Guinand, F., Guérin, F., Łubniewski, P.: Allowing people to communicate after a disaster using FANETs. In: Krief, F., Aniss, H., Mendiboure, L., Chaumette, S., Berbineau, M. (eds.) Communication Technologies for Vehicles. LNCCN, vol. 12574, pp. 181–193. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66030-7_16
2. Hydher, H., Jayakody, D.N.K., Hemachandra, K.T., Samarasinghe, T.: Intelligent UAV deployment for a disaster-resilient wireless network. Sensors **20**(21), 6140 (2020). https://doi.org/10.3390/s20216140
3. Grzeszczak, J., Mikitiuk, A., Trojanowski, K.: Station Coverage Problem 2 (SCP2) dataset (2022). https://jaga.blog.uksw.edu.pl/scp2/. Accessed 25 Apr 2022
4. Košmerl, J., Vilhar, A.: Base stations placement optimization in wireless networks for emergency communications. In: 2014 IEEE International Conference on Communications Workshops (ICC), pp. 200–205 (2014). https://doi.org/10.1109/ICCW.2014.6881196
5. Lyu, J., Zeng, Y., Zhang, R., Lim, T.J.: Placement optimization of UAV-mounted mobile base stations. IEEE Commun. Lett. **21**(3), 604–607 (2017). https://doi.org/10.1109/LCOMM.2016.2633248
6. Masroor, R., Naeem, M., Ejaz, W.: Efficient deployment of UAVs for disaster management: a multi-criterion optimization approach. Comput. Commun. **177**, 185–194 (2021). https://doi.org/10.1016/j.comcom.2021.07.006
7. Quaritsch, M., Kruggl, K., Wischounig-Strucl, D., Bhattacharya, S., Shah, M., Rinner, B.: Networked UAVs as aerial sensor network for disaster management applications. e & i Elektrotechnik und Informationstechnik **127**(3), 56–63 (2010). https://doi.org/10.1007/s00502-010-0717-2
8. Sawalmeh, A., Othman, N.S., Liu, G., Khreishah, A., Alenezi, A., Alanazi, A.: Power-efficient wireless coverage using minimum number of UAVs. Sensors **22**(1), 223 (2021). https://doi.org/10.3390/s22010223

9. Srinivas, A., Zussman, G., Modiano, E.: Construction and maintenance of wireless mobile backbone networks. IEEE/ACM Trans. Netw. **17**(1), 239–252 (2009). https://doi.org/10.1109/TNET.2009.2012474
10. Sánchez-García, J., García-Campos, J.M., Toral, S.L., Reina, D.G., Barrero, F.: An intelligent strategy for tactical movements of UAVs in disaster scenarios. Int. J. Distrib. Sens. Netw. **12**(3), 8132812 (2016). https://doi.org/10.1155/2016/8132812
11. Yuheng, Z., Liyan, Z., Chunpeng, L.: 3-D deployment optimization of UAVs based on particle swarm algorithm. In: 2019 IEEE 19th International Conference on Communication Technology (ICCT). IEEE, October 2019. https://doi.org/10.1109/icct46805.2019.8947140