drones

MDPI

Article

Deployment Method with Connectivity for Drone Communication Networks

Hirofumi Osumi [1], Tomotaka Kimura [1,*], Kouji Hirata [2], Chinthaka Premachandra [3] and Jun Cheng [1]

[1] Graduate School of Science and Engineering, Doshisha University, Kyoto 610-0321, Japan; jcheng@mail.doshisha.ac.jp (J.C.)
[2] Faculty of Engineering Science, Kansai University, Suita 565-0842, Japan; hirata@kansai-u.ac.jp
[3] Graduate School of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan; chintaka@shibaura-it.ac.jp
* Correspondence: tomkimur@mail.doshisha.ac.jp; Tel.: +81-774-65-6294

Abstract: In this paper, we consider a drone deployment problem in situations where the number of drones to be deployed is small compared to the number of users on the ground. In this problem, drones are deployed in the air to collect information, but they cannot collect information from all ground users at once due to the limitations of their communication range. Therefore, the drones need to continue to move until they collect the information for the all ground users. To efficiently realize such drone deployment, we propose two deployment methods. One is an integer linear programming (ILP)-based deployment method and the other is an adjacent deployment method. In the ILP-based deployment method, the positions of the drones at each point in time are determined by solving an ILP problem in which the objective function is the total number of users from whom data can be collected. In contrast, in the adjacent deployment method, drones are sequentially deployed in areas with probabilities determined according to the number of user nodes in adjacent areas at which other drones are already deployed. Through numerical experiments, we show that these deployment methods can be used to efficiently collect data from user nodes on the ground.

Keywords: drone communication networks; drone deployment problem; Integer Linear Programming

Citation: Osumi, H.; Kimura, T.; Hirata, K.; Premachandra, C.; Cheng, J. Deployment Method with Connectivity for Drone Communication Networks. *Drones* **2023**, *7*, 384. https://doi.org/10.3390/drones7060384

Academic Editor: Carlos Tavares Calafate

Received: 30 April 2023
Accepted: 5 June 2023
Published: 7 June 2023

1. Introduction

In recent years, drone technologies have attracted much attention [1–3] for object tracking [4], the transportation of goods [5,6], traffic surveillance [7], and remote sensing [8]. Moreover, drone technologies are expected to be useful in responding to natural disasters [9–12]. In the communication field, drone communication networks that comprise multiple drones are expected to be used in disaster situations. Base stations can be destroyed in a natural disaster, and thus victims on the ground will not be connected to the Internet. After the disaster, multiple drones can be flown to recover from disconnections, and the victims can use these drones as access points (Figure 1). Such drone communication networks can monitor the situation in disaster areas and help many victims.

To date, many drone communication networks have been considered [13–20]. Drone communication networks can be considered as a new form of mobile ad hoc network (MANET) where the drones can move more freely than mobile devices in conventional MANETs. Therefore, early studies considered improvements to the routing schemes used in conventional MANETs. In [13], based on optimized link state routing (OLSR), directional OLSR was proposed, assuming that each drone was equipped with a directional antenna. In [14], ad hoc on-demand distance vector (AODV) routing was applied to a drone communication network. These early studies did not fully consider the characteristics of drone communication networks. In particular, unlike the case for conventional MANETs, they did not consider that drone movements can be managed by ground control systems. In [15], a ground control system-based routing method was proposed. The effectiveness of

the routing method was demonstrated in experiments using real drones implementing the routing method and simulation experiments.

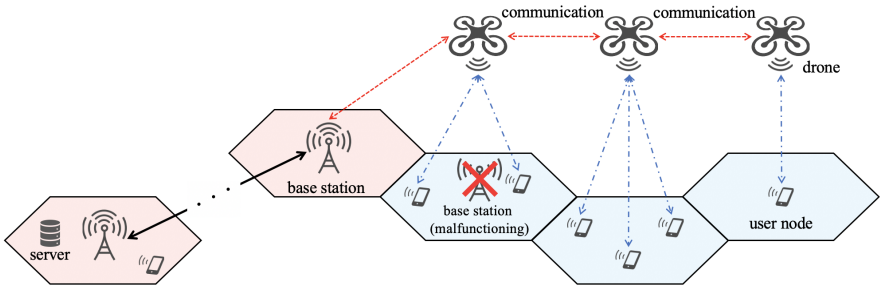

Figure 1. Drone networks.

In [18–28], the drone deployment problem was studied. In the drone deployment problem, the placement of drones was considered in order to efficiently collect data from user nodes on the ground. In [23], the placement of a single drone was considered, and the optimum altitude in terms of the maximum allowed path loss was analyzed. The placement of multiple drones was considered in [24–26]. In [24,25], without considering the direct connectivity between drones, drone placement could be determined because the drones were connected via satellite links. Since drones are not always equipped with satellite communication devices, direct connectivity between drones needs to be considered. In [26], the authors considered an optimization problem involving minimizing the number of drones to accommodate the traffic of users on the ground under the condition of the drones being connected with each other. Numerical experiments showed that the connected drone networks could be efficiently constructed. In [26], although it was assumed that a minimum number of drones is generally available, in practice, the number of users on the ground may be so large that the available drones may not cover all users at once.

In this paper, we consider a connected drone deployment problem; that is, which areas to place drones in to connect them to each other. By connecting drones, the coverage area can be extended and services can be provided to users in locations where the signals from the base station cannot directly reach. As shown in Figure 1, the drones can communicate with each other if they are placed in adjacent areas, and users in areas far from the base station can communicate with servers on the Internet using connections established by drones located in adjacent areas. For this problem, it was assumed that the number of drones to be deployed is much smaller than the number of users on the ground. The drones cannot collect information from all ground users at once due to the limitations of their communication range. The drones need to continue to move until they collect the information from all the ground users. Therefore, in our problem, it was necessary to consider the placement of the drones at each point in time, maintaining connectivity between them. In addition, it was judged to be desirable for the information from all ground users to be collected as quickly as possible. Therefore, in summary, the solution to the problem must satisfy the following conditions: the connection between the drones should be maintained in every time slot, and the connection requests of the users should be satisfied as quickly as possible.

To obtain such a solution, we considered two deployment methods: the integer linear programming (ILP)-based deployment method and the adjacent deployment method. In the ILP-based deployment method, at each point in time, the ground control system must solve the ILP problem, the objective function of which is to maximize the total number of users that can communicate with the drone at each time, and determine the positions of the drones that are connected with each other. As time elapses, the ground control system repeatedly solves the ILP problem. With this process, the drones can collect the information from all ground users. In contrast, the adjacent deployment method is a heuristic algorithm and, at each point in time, the ground control system deploys drones to an area adjacent to the areas in which drones are already placed. More specifically, there is a high probability

that a drone will be deployed to the area with the largest number of user nodes and a low probability that it will be deployed to other areas. Through numerical experiments, we show that ILP-based deployment is the fastest way to collect information from all users. Furthermore, we show that the performance of the adjacent deployment method is comparable to that of the ILP-based deployment method.

The contributions of this paper can be summarized as follows:

- We consider the connected drone placement problem; i.e., in which areas should drones be placed in order to connect drones to each other. By solving this problem, the coverage area can be extended and communications can be established even in areas beyond the range of the base station's signal. This is useful for the collection of information after a disaster;
- To solve the connected drone placement problem, we formalized the ILP optimization problem and propose an ILP-based deployment method. As shown in Section 4, the ILP-based deployment method can realize efficient drone deployment;
- We propose a heuristic algorithm named the adjacent deployment method for large-scale networks. The computation time for our ILP optimization problem is generally huge because it is NP-hard. Therefore, the adjacent deployment method can be used to obtain the solution for large-scale networks for which it would be difficult for the ILP-based deployment method to obtain the solution.

The rest of this paper proceeds as follows. In Section 2, the problem addressed in this paper is described. In Section 3, we describe the ILP-based deployment method and the adjacent deployment method. In Section 4, we evaluate the proposed methods through simulation experiments. Finally, in Section 5, we conclude this paper.

2. Problem Formulation

There are N_U users and N_D drones in area A. Let $\mathcal{U}$ and $\mathcal{D}$ denote the sets of users and drones, respectively. $\mathcal{U} = \{1, 2, \ldots, N_U\}$ and $\mathcal{D} = \{1, 2, \ldots, N_D\}$. The area A is divided into L disjoint subareas A_l ($l \in \mathcal{L}$), where $\mathcal{L} = \{1, 2, \ldots, L\}$. By definition, $\bigcup_{l \in \mathcal{L}} A_l = A$ and $A_i \cap A_j = \phi$ ($i, j \in \mathcal{L}, i \neq j$). The set of subareas is defined as $\mathcal{A} = \{A_l \mid l \in \mathcal{L}\}$.

In this paper, we assume that each user has a connection request destined for the sink node, as shown in Figure 1, where a base station is regarded as the sink node. Through the sink node, requests for connection to the Internet can be fulfilled; i.e., to the servers outside of area A. We aim to establish connections for all the requests, but it is hard to simultaneously cover all the requests because the number of drones is supposed to be much smaller than the number of users (i.e., the number of connection requests). Therefore, we can achieve this by iteratively establishing connections. Specifically, we first establish connections for some users by deploying drones to ensure the connectivity to the sink node. Then, we reposition the drones to a different set of subareas to cover the connection requests of other users. Until we have covered all the connection requests, we repeat this procedure.

To do so, time is divided into time slots with the same duration. In the problem, for each time slot, to maintain the connectivity to the base station, the deployment subareas for the drones are selected in such a way that a connected graph can be constructed with the drones as nodes (Figure 2). Then, the drones stay in the selected areas until the end of the time slot. Here, let $P_d(t)$ denote the subarea of drone d ($d \in \mathcal{D}$) at time slot t ($t = 1, 2, \ldots$). Note that $P_d(t) \in \mathcal{A}$. Drone d is equipped with an omnidirectional antenna and can communicate with ground users in area $P_d(t)$ at time t. The size of a subarea depends on the communication radius of the drones' antennas, and we thus assume that the subarea size is fixed to fit within the communication radius of an antenna.

At time 0, there are N_U connection requests in the network. In each connection request, an application-layer message is sent, and thus its data size is supposed to be relatively large. Moreover, in each time slot, each drone is supposed to accommodate at most M connection requests. The number M of served connection requests depends on the length of the time slot. As the length of a time slot increases, the value of M increases. However, for a long

time slot, even if all users in the area have sent all the data that they want to send, the drone cannot move to another area until the end of the time slot, and this waiting is wasteful.

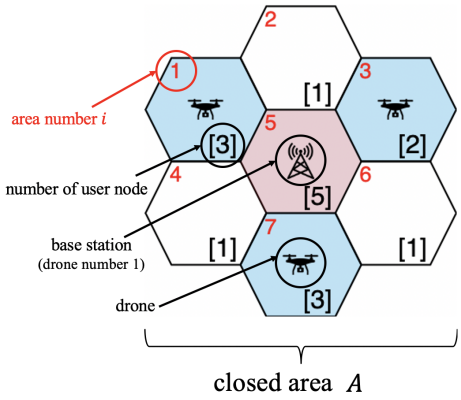

Figure 2. System model.

Note that the value of M depends on not only the length of the time slot but also several other factors, such as radio interference. For example, when the impact of the radio interference is strong, the value of M decreases. This is because the throughput of each connection decreases due to the radio interference, and thus the number of connection requests that the drone can complete for the data transmission during the time slot decreases. We can indirectly consider such situations by appropriately setting the value of M. In this paper, the value of M is given as a parameter without discussing how the length of the time slot is determined and how much radio interference there is because we aim to discuss a common way to construct connected drone networks that can accommodate all the connection requests for various situations.

We also assume that the drones have directional antennas. The transmission distance of a directional antenna is long compared to the transmission distance of an omnidirectional antenna [29]. Therefore, when the drones are deployed in the neighboring areas (Figure 1), they can communicate with each other.

Under the above assumptions, we consider the connected drone placement problem of minimizing the completion time T required to respond to the connection requests from all users by moving drones. Formally, the completion time T is defined as

$$T = \inf_{t \in \mathbb{N}} \{t \mid \mathcal{R}(t) = \phi\},$$

where $\mathcal{R}(t)$ denotes the unsatisfied connection requests at time t. Note that $\mathcal{R}(0) = \mathcal{U}$. The completion time T depends on the drone deployment, and we consider good drone deployment as having a short completion time T in this paper. Moreover, for the communication with the cloud servers via a base station, for each time t, we consider only the drone deployment where all drones are connected at each time slot. The desirable solution for the connected drone placement problem can be summarized as follows:

- The completion time T is short;
- For each time slot, a connected graph is constructed with drones as nodes.

In the problem formulation, we assume that users $\mathcal{U}$ and their positions are known in advance. In practice, the information needs to be collected before our proposed methods are applied. One way of doing this is to fly drones over areas and investigate the number of users in each area. If there are user devices in each area that have been able to communicate with the drone, the drone counts the number of these devices. Our proposed methods are supposed to be used in situations some time after a disaster. Immediately after a disaster, drones fly around to find and rescue victims, but after some time has passed, users with devices remain in fixed positions, allowing their locations to be collected by drones.

Moreover, although solving the problem allows us to obtain the solution regarding which area to place the drones in to maintain connectivity to the base station, it does not indicate where the best place to collect information from the ground user is. The best drone placements for collecting information from ground users have been studied [24,25,27,28], and these methods can be applied. Although our proposed method can combine these methods, we leave the combination problem for future work.

3. Proposed Method

We explain the ILP-based method and adjacent placement method in Sections 3.1 and 3.2, respectively.

3.1. ILP-Based Deployment

In the ILP-based method, until all connection requests are satisfied, the placements $P_d(t)$ ($d \in \mathcal{D}$) are determined for each time t. The placements $P_d(t)$ at time t are determined by solving the ILP problem explained later. Table 1 summarizes the symbols used in the ILP-based method. The procedure of the ILP-based method is as follows:

Step 1. $t := 1$ and $\mathcal{R}(1) := \mathcal{U}$;

Step 2. $k := 1$, $n := 1$, and $\mathcal{W}(t) := \phi$;

Step 3. The maximum number N_G of drones in each group is set to $N_G := n$ and the number of groups K is set to $K := \lceil |\mathcal{D}|/N_G \rceil$. Therefore, the drones are divided into K groups $\mathcal{G}_\ell$ ($\ell = 1, 2, \ldots, K$) as follows:

$$\mathcal{G}_\ell = \begin{cases} \{(\ell-1)N_G + 1,\ (\ell-1)N_G + 2,\ \ldots,\ \ell N_G\} & (\ell = 1, 2, \ldots, K-1), \\ \{(\ell-1)N_G + 1,\ (\ell-1)N_G + 2,\ \ldots,\ |\mathcal{D}|\} & (\ell = K); \end{cases}$$

Step 4. By solving the ILP problem, the placements $P_d(t)$ ($d \in \mathcal{G}_k$) can be determined;

Step 5. If $k = K$, the procedure goes to step 6. Otherwise, k is updated to $k := k + 1$ and the procedure returns to step 4;

Step 6. $\mathcal{W}_n(t)$, which is the set of users with satisfied connection requests according to the placements $\mathcal{P}_d(t)$ ($d \in \mathcal{D}$) when $N_G = n$, is calculated. If $|\mathcal{W}_n(t)| > |\mathcal{W}(t)|$, $\mathcal{W}(t)$ is updated to $\mathcal{W}(t) := \mathcal{W}_n(t)$;

Step 7. If $n = |\mathcal{D}|$, the procedure goes to step 8. Otherwise, $k := 1$, $n := n + 1$, and the procedure returns to step 3;

Step 8. t is updated to $t := t + 1$. $\mathcal{R}(t)$ is updated to $\mathcal{R}(t) := \mathcal{R}(t-1) \setminus \mathcal{W}(t-1)$. If $\mathcal{R}(t) = \phi$, the algorithm is stopped. Otherwise, the procedure returns to step 2.

In step 3, the drones in the kth group $\mathcal{G}_k$ are deployed through the ILP. To obtain the connected drone deployment for each time t, instead of determining the deployment of all drones at once, the drones are divided into K groups. The formulation of the ILP is difficult when all drones are assigned at once, as a topology with branches cannot be obtained. We overcome this problem by dividing the drones into groups.

In the ILP, the placement of the kth group $\mathcal{G}_k$ is considered under the assumption that the drones in $\mathcal{G}_\ell$ ($\ell = 1, 2, \ldots, k-1$) have already been placed. Let $\overline{\mathcal{D}}_k$ denote the set of the drones already deployed when the deployment of drones in $\mathcal{G}_k$ is considered: $\overline{\mathcal{D}}_k = \cup_{\ell=1}^{k-1} \mathcal{G}_\ell$. Furthermore, $\overline{\mathcal{L}}_k$ denotes the set of the areas where drones have already been deployed: $\overline{\mathcal{L}}_k = \cup_{\ell=1}^{k-1} \cup_{d \in \mathcal{G}_\ell} \{P_d(t)\}$. The drones in $\mathcal{G}_k$ can be placed in subareas $\mathcal{L} \setminus \overline{\mathcal{L}}_k$. For these subareas, we determine the placement of the drones in $\mathcal{G}_k$ by solving the following optimization problem.

$$\text{Maximize} \quad \sum_{u \in \mathcal{R}(t)} \sum_{d \in \mathcal{G}_k} v_{u,d} \tag{1}$$

$$\begin{aligned}
\text{Subject to} \quad & \forall d, e \in \overline{\mathcal{D}}_k \cup \mathcal{G}_k, \forall i, j \in \mathcal{L}, \\
& a_{i,j} + x_{d,i} + x_{e,j} - 2 \leq y_{i,d}^{j,e} \\
& a_{i,j} \geq y_{i,d}^{j,e} \\
& x_{d,i} \geq y_{i,d}^{j,e} \\
& x_{e,j} \geq y_{i,d}^{j,e} \qquad (2) \\
& \forall d, e \in \overline{\mathcal{D}}_k \cup \mathcal{G}_k, \\
& \sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{L}} y_{i,d}^{j,e} = z_{d,e} \qquad (3) \\
& d = l_k, l_k + 1, \ldots, l_k + |\mathcal{G}_k| - 2, \\
& z_{d,d+1} = z_{d+1,d} = 1 \qquad (4) \\
& \sum_{\bar{d} \in \overline{\mathcal{D}}_k} z_{l_k,\bar{d}} \geq 1 \qquad (5) \\
& \forall d \in \mathcal{G}_k, \\
& \sum_{i \in \mathcal{L}} x_{d,i} = 1 \qquad (6) \\
& \forall d \in \overline{\mathcal{D}}_k, \\
& x_{d,m(d)} = 1, \sum_{i \in \mathcal{L}} x_{d,i} = 1 \qquad (7) \\
& \forall d \in \overline{\mathcal{D}}_k \cup \mathcal{G}_k, \forall i \in \mathcal{L}, \forall u \in \mathcal{R}(t), \\
& x_{d,i} + c_{u,i} - 1 \leq w_{u,d,i} \\
& x_{d,i} \geq w_{u,d,i} \\
& c_{u,i} \geq w_{u,d,i} \qquad (8) \\
& \forall d \in \overline{\mathcal{D}}_k \cup \mathcal{G}_k, \forall u \in \mathcal{R}(t), \\
& \sum_{i \in \mathcal{L}} w_{u,d,i} = v_{u,d} \qquad (9) \\
& \forall i \in \mathcal{L}, \\
& \begin{cases} \sum_{d \in \overline{\mathcal{D}}_k \cup \mathcal{G}_k} x_{d,i} \leq \lceil \frac{\gamma_i}{M} \rceil & (\gamma_i \geq 1) \\ \sum_{d \in \overline{\mathcal{D}}_k \cup \mathcal{G}_k} x_{d,i} \leq 1 & (\gamma_i = 0) \end{cases} \qquad (10) \\
& d = 1, \\
& x_{d,B} = 1 \qquad (11)
\end{aligned}$$

Equation (1) is the objective function that represents the number of users that can communicate with the drones in $\mathcal{G}_k$. Equation (2) is the constraint that the drones are placed in subareas adjacent to each other. Equation (3) eliminates the area information i, j from $y_{i,d}^{j,e}$, and $z_{d,e}$ represents whether drones d and e can communicate with each other. Equation (4) is the constraint that the drones with successive IDs are placed in adjacent subareas and can communicate with each other. Equation (5) is the constraint that the drone with the smallest index l_k in $\mathcal{G}_k$ is able to communicate with and be adjacent to one of the drones in $\overline{\mathcal{D}}_k$. Equation (6) is the constraint that each drone is deployed in only one subarea among all subareas. Equation (7) is the constraint that the already deployed drone $d \in \overline{\mathcal{D}}_k$ is deployed in subarea $m(d)$ and is not deployed in other subareas $\mathcal{A} \setminus \{m(d)\}$. In Equation (8), $w_{u,d,i}$ represents whether user u stays in area A_i and communicates with drone d. In Equation (9), $v_{u,d}$ is the transformation of $w_{u,d,i}$ through the elimination of the information in area i and

represents whether user u can communicate with drone d. Equation (10) is the constraint for the maximum number of drones in each area A_i. Note that γ_i is constant when solving the ILP. Equation (11) is the constraint that drone one is deployed in the area containing the base station.

Table 1. List of symbols.

Symbol	Meaning
$\mathcal{A}$	Set of subareas
$\overline{\mathcal{D}}_k$	Set of drones in groups $\mathcal{G}_i$ ($i = 1, 2, \ldots, k-1$); i.e., $\cup_{i=1}^{k-1}\mathcal{G}_i$
$\mathcal{G}_k$	Set of drones in group k
$\mathcal{R}(t)$	Set of users for which no data have been collected by time t
l_k	Minimum identification number of the drone included in $\mathcal{G}_k$
$m(d)$	Placement of drone d ($d \in \overline{\mathcal{D}}_k$)
$v_{u,d}$	Binary variable that is equal to 1 if user u is communicating with drone d; otherwise, 0
$a_{i,j}$	Variable that is equal to 1 if areas i and j are adjacent; otherwise, 0
$x_{d,i}$	Binary variable that is equal to 1 if drone d is deployed in area i; otherwise, 0
$y_{i,d}^{j,e}$	Binary variable that is equal to 1 if drones d and e deployed in areas i and j are communicating; otherwise, 0
$z_{d,e}$	Binary variable that is equal to 1 if drones d and e are communicating; otherwise, 0
$c_{u,i}$	Binary variable that is equal to 1 if user u is in area i; otherwise, 0
$w_{u,d,i}$	Binary variable that is equal to 1 if user u communicates with drone d and is in area i; otherwise, 0
γ_i	$\sum_{u \in \mathcal{R}(t)} c_{u,i}$
M	Maximum number of users that a drone can communicate with
N_G	Maximum number of drones in group
B	Area number of the base station

Figure 3 shows an example of the drone deployment using the ILP-based deployment method. In the example, the number of drones $|\mathcal{D}| = 6$ and the number of subareas $L = 7$. We set the number of groups K to 3. Therefore, $N_G = 2$, $\mathcal{G}_1 = \{1,2\}, \mathcal{G}_2 = \{3,4\}$, and $\mathcal{G}_3 = \{5,6\}$. The drones in each group are connected (e.g., for $\mathcal{G}_1$, drones one and two are connected). First, the deployment of the drones in $\mathcal{G}_1$ is determined through the ILP. Drones one and two are deployed in subareas five and one, respectively. The drones in $\mathcal{G}_2$ and drones three and four are then deployed to subareas seven and four, respectively. By connecting drone three to drone one, the drones in $\mathcal{G}_1$ and $\mathcal{G}_2$ are connected with each other. In the same way, the drones in $\mathcal{G}_3$ are deployed to subareas two and three. Although the deployment of drones at $t = 1$ is determined, no connection request is satisfied (e.g., the connection request in subarea six is not satisfied). Therefore, t is updated to $t := 2$, and then the drones are deployed again using the ILP.

Note that, even though our system model described in Section 2 simplifies the radio interference, the computation time of our ILP optimization problem is generally huge because it is NP-hard. Therefore, our system model is suitable for solving the connected drone placement problem, as the computation time for solving the ILP optimization problem is reduced compared to a system model with a complex radio propagation model.

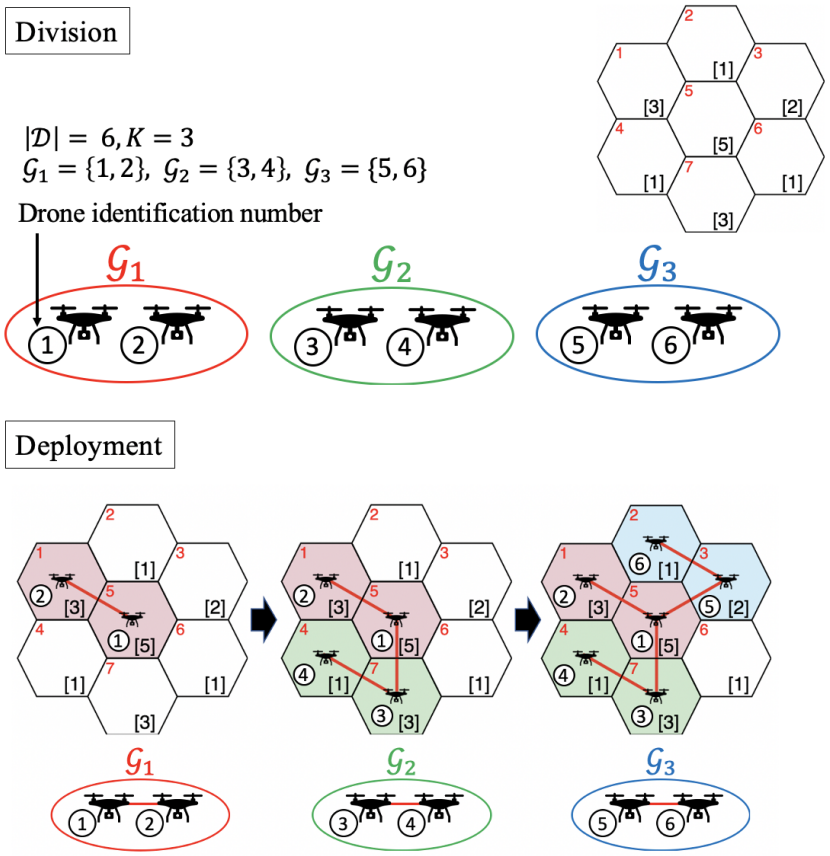

Figure 3. Example of group division.

3.2. Adjacent Placement Method

In the adjacent placement method, to obtain a connected network, the deployment of drone d is chosen from the subareas $f_{\mathrm{NA}}(\mathcal{P}(t))$ that are adjacent to the subareas where the drones are deployed already in $\mathcal{P}(t)$. Note that $f_{\mathrm{NA}}(\mathcal{P}(t)) \subset \mathcal{A}$. When there are no users in the neighboring subareas $f_{\mathrm{NA}}(\mathcal{P}(t))$, the neighboring subarea in $f_{\mathrm{NA}}(\mathcal{P}(t))$ is expanded outward to find a subarea with a user. If such a subarea is found, it is connected to the subarea in $\mathcal{P}(t)$ to make the connected deployment. The detailed procedure for the adjacent placement method is as follows:

Step 1. $t := 1$. For $A \in \mathcal{A}$, $\mathcal{R}_A(1) := \mathcal{U}_A$, where $\mathcal{U}_A$ denotes the initial set of users with unsatisfied connection requests in subarea A;

Step 2. $d := 2$, $P_1(t) = A_B$, and $\mathcal{P}(t) = \{P_1(t)\}$;

Step 3. The set $\mathcal{A}_{\mathrm{UC}}$ of adjacent subareas with users with unsatisfied connection requests is updated to $\mathcal{A}_{\mathrm{UC}} := \{A \mid |\mathcal{R}_A(t)| > 0, A \in f_{\mathrm{NA}}(\mathcal{P}(t))\}$. If $|\mathcal{A}_{\mathrm{UC}}| > 0$, subarea A is randomly chosen from $\mathcal{A}_{\mathrm{UC}}$ with probability ϵ. In contrast, A is set to area A_{large} with the largest number of users in $\mathcal{A}_{\mathrm{UC}}$ with probability $1-\epsilon$. When there are multiple subareas with the maximum number of users in $\mathcal{A}_{\mathrm{UC}}$, A is randomly chosen from among those areas. After the setting of A, drone d is assigned to $P_d(t) = A$ and $\mathcal{P}(t)$ is updated to $\mathcal{P}(t) := \mathcal{P}(t) \cup \{A\}$. Otherwise, the following steps are undertaken:

Step 3.1. $l = 1$. $\mathcal{A}^{(0)} := \mathcal{P}(t) \cup f_{\mathrm{NA}}(\mathcal{P}(t))$ and $\mathcal{A}^{(1)} := f_{\mathrm{NA}}(\mathcal{A}^{(0)})$;

Step 3.2. $\mathcal{A}^{(l)}_{\mathrm{UC}} = \{A \mid |\mathcal{R}_A(t)| > 0, A \in \mathcal{A}^{(l)}\}$ is calculated;

Step 3.3. If $|A^{(l)}_{\mathrm{UC}}| > 0$, the procedure goes to step 3.4. Otherwise, $l := l+1$, $\mathcal{A}^{(l)}$ is set to $f_{\mathrm{NA}}(\mathcal{A}^{(l-1)}) \setminus \bigcup_{k=0}^{l-1} \mathcal{A}^{(k)}$, and the procedure returns to step 3.2;

Step 3.4. If $|\mathcal{D}| < d + l + 1$, subarea A is randomly chosen from $A^{(l)}_{\mathrm{UC}}$. We select the set of subareas $\mathcal{A}_{\mathrm{short}}$, where subarea A is connected to one of the

subareas in $\mathcal{P}(t)$ with the smallest number of hops. $d := d + |\mathcal{A}_{\text{short}}|$, and $\mathcal{P}(t) := \mathcal{P}(t) \cup \mathcal{A}_{\text{short}} \cup \{A\}$. Otherwise, subarea A is randomly chosen from $f_{\text{NA}}(\mathcal{P}(t))$. Drone d is assigned to $P_d(t) = A$ and $\mathcal{P}(t)$ is updated to $\mathcal{P}(t) := \mathcal{P}(t) \cup \{A\}$;

Step 4. If the number $|\mathcal{R}_A(t)|$ of users in subarea A is greater than M, multiple drones are deployed in subarea A. Specifically, if $\lceil |\mathcal{R}_A(t)|/M \rceil \leq |\mathcal{D}| - d + 1$, $\lceil |\mathcal{R}_A(t)|/M \rceil$ drones are deployed to subarea A. In other words, drones $d, d+1, \ldots, d-1+\lceil |\mathcal{R}_A(t)|/M \rceil$ are deployed to subarea A. Otherwise, $|\mathcal{D}| - d + 1$ drones are deployed to subarea A. d is updated to $d := |\mathcal{D}|$;

Step 5. If $d = |\mathcal{D}|$, the procedure goes to step 6. Otherwise, $d := d + 1$, and then the procedure returns to step 3;

Step 6. Based on $P_d(t)$ $(d \in \mathcal{D})$, $\mathcal{R}_A(t+1)$ $(A \in \mathcal{A})$ is calculated, and $\mathcal{R}(t+1)$ is set to $\cup_{A \in \mathcal{A}} \mathcal{R}_A(t+1)$. If $\mathcal{R}(t+1) = \phi$, the algorithm is stopped. Moreover, if $\mathcal{R}(t) = \mathcal{R}(t+1)$, $d := 2$, and then the procedure returns to step 3. Otherwise, t is updated to $t := t + 1$, and then the procedure returns to step 2.

Steps 3.1–3.4 are the process of finding the subarea where users exist and connecting that subarea to the subarea where the drones are already deployed. Step 4 is the process of deploying multiple drones in a subarea.

Figure 4 shows an example of drone placement using the adjacent placement method. In Figure 4, $|\mathcal{D}|$ is 6 and M is 5. First, drone one is deployed in subarea five. Drone two is placed in subarea five because the number of users in subarea five is larger than M. Next, drone three is placed in subarea seven with probability $1 - \epsilon$. Drone four is deployed in subarea seven because the number of users in subarea seven is greater than $M = 5$. Drones three and four are placed in the small area adjacent to the small area where drones one and two are placed. Similarly, drones five and six are deployed in subareas four and three, respectively.

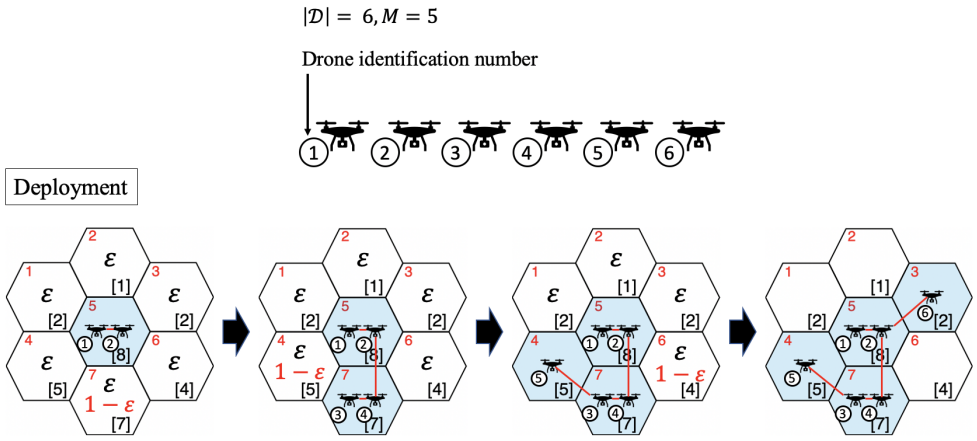

Figure 4. Example of implementing the adjacent deployment method.

4. Performance Evaluation

4.1. Evaluation Model

Numerical experiments were conducted to verify the effectiveness of the proposed methods. Table 2 shows the parameter values used in the numerical experiments. There are 400 users ($|\mathcal{U}| = 400$) in a closed area A. The area A is divided into $L \in \{25, 49, 81, 121, 169, 225\}$ subareas, and users are deployed in these subareas. The base station is located in subarea $B = \lceil L/2 \rceil$. The number of drones $|\mathcal{D}|$ is set to 14, and each drone can communicate with $M = 5$ users at each time. In this paper, we consider three scenarios with different user arrangements: random, uneven, and all-equal scenarios. In the random scenario, the number of users in each subarea is randomly chosen according to a uniform distribution on [0, 10]. Figure 5 shows an example of the deployment in the random scenario. In the uneven

scenario, users only exist in some areas. Figure 6 shows an example of the deployment in the uneven scenario. Users exist only in the upper-right and lower-left subareas. The number of users in the upper-right and lower-left subareas is chosen according to a uniform distribution on [20, 30].

In the all-equal scenario, as shown in Figure 7, the number of users is the same for all subareas; i.e., $|\mathcal{U}|/L$.

Table 2. Parameter settings.

Parameter	Value
Number of users $\|\mathcal{U}\|$	400
Number of subareas L	25, 49, 81, 121, 169, 225
Base station deployment area B	$\lceil L/2 \rceil$
Number of drones $\|\mathcal{D}\|$	14
Maximum number of users from which one drone can collect data M	5

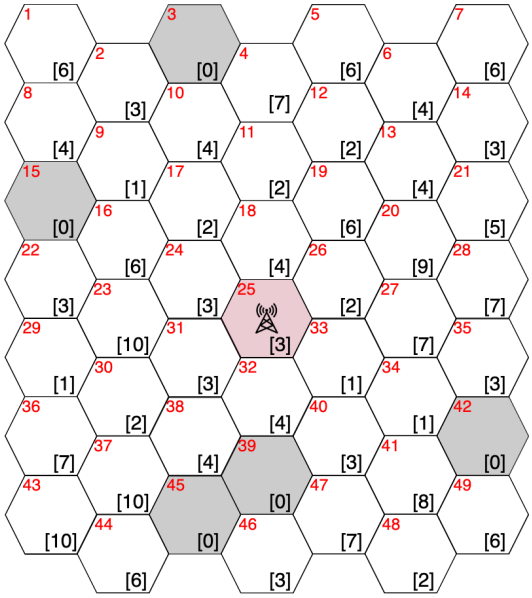

Figure 5. Random scenario.

In the ILP-based deployment method, to solve the ILP problem described in Section 3.1, we use the optimization solver CPLEX [30]. In the adjacent deployment method, the probability ϵ is set to 0.05, unless otherwise stated.

The completion time T was used as the performance metric. For the random and uneven scenarios, 10 trials were conducted with different user arrangements, and the average over these trials was taken as the completion time.

Results for three other methods (random, rolling, and spiral methods) are presented for performance comparison. These are straightforward methods and thus regarded as the baseline methods. The random method is consistent with the adjacent placement method when $\epsilon = 1$ and determines the drone deployment area randomly from among the neighboring areas. The random method was run 1.0×10^3 times, and the results are shown for the most efficient data collection among the runs. In the rolling method, drones are repeatedly deployed so that they rotate to the right, as shown in Figure 8. The drones are deployed in a circular pattern until they have completed a full circle. In the spiral method, the drones are repeatedly deployed in a spiral pattern around the base station, as shown in Figure 9.

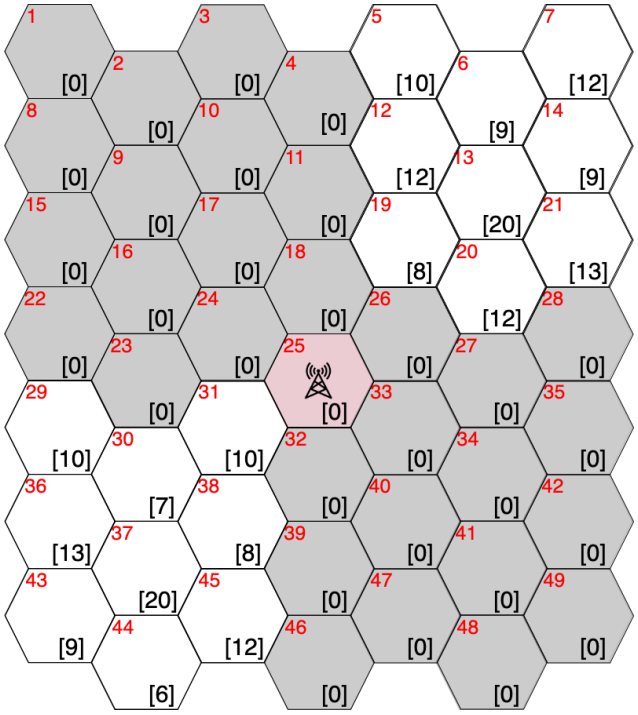

Figure 6. Uneven scenario.

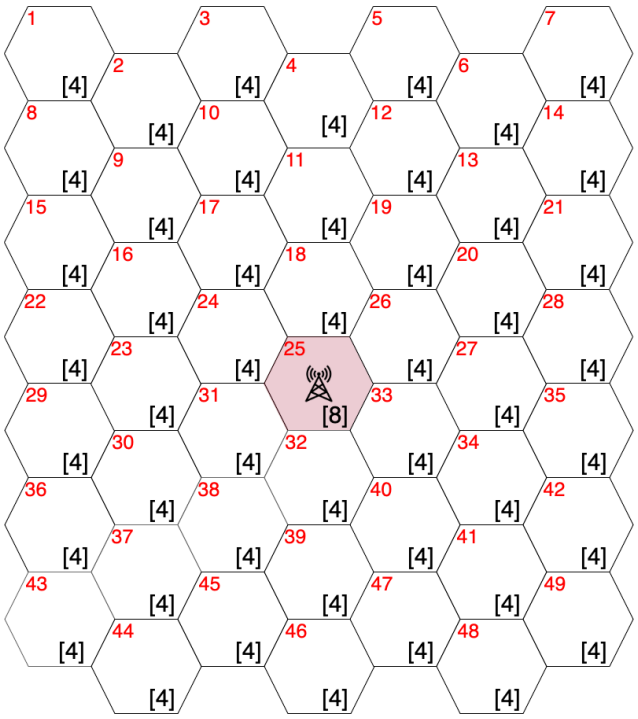

Figure 7. All-equal scenario.

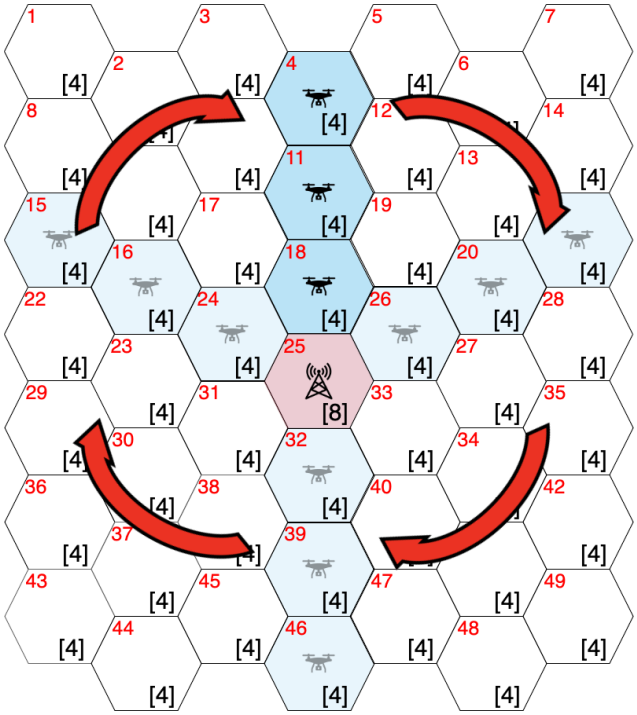

Figure 8. Rolling method.

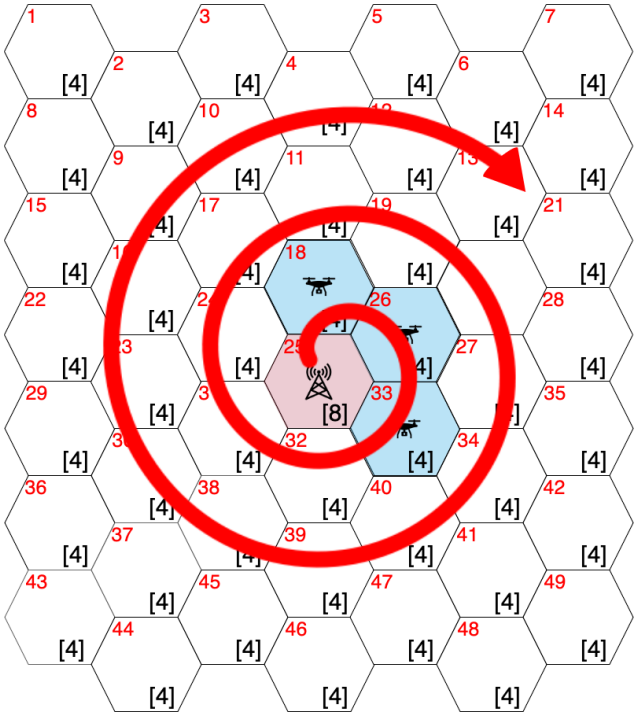

Figure 9. Spiral method.

4.2. Results

Figure 10 shows the completion time T as a function of the number L of subareas. From Figure 10, we can observe that the proposed methods outperformed the comparison methods in all scenarios. As L increased, the difference between the proposed methods and the other methods increased. These results indicate that our proposed method performs well compared with the comparison methods. In particular, for a large L, the use of our proposed method is effective. Moreover, for a small L, the completion time T of the adjacent placement method is comparable with that of the ILP-based deployment method.

For $L = 225$, the computation complexity of the ILP-based deployment is high, and we cannot obtain the solution in a reasonable time. The adjacent placement method performs best and is useful for a large number of subareas.

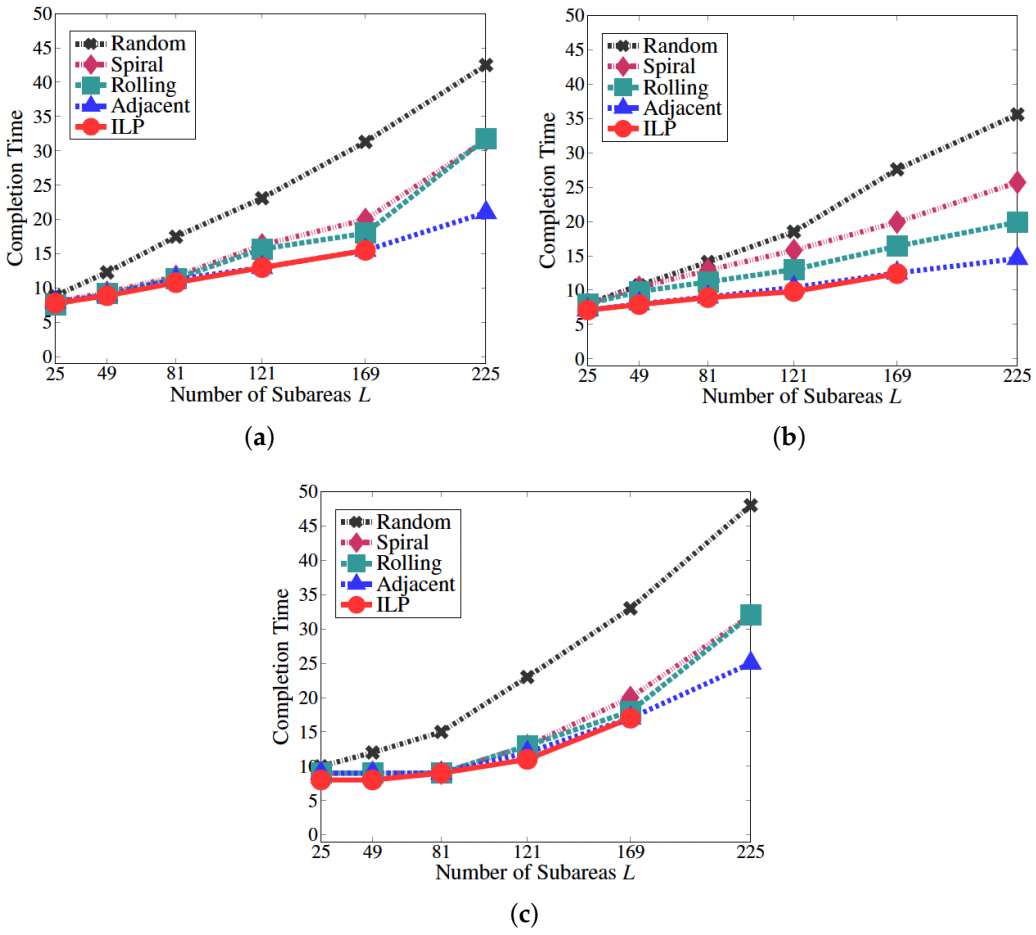

Figure 10. Completion time T as a function of the number of subareas. (**a**) Random scenario; (**b**) uneven scenario; (**c**) all-equal scenario.

Figure 11 shows the completion time T as a function of the number $|\mathcal{U}|$ of users. For any $|\mathcal{U}|$, the ILP-based deployment achieved the best performance. For a large $|\mathcal{U}|$, compared with the spiral and rolling deployments, the adjacent placement method and random deployments demonstrated short completion times T, and their performances were comparable. Note that the computation time of the random deployment was longer than that of the adjacent placement method because the random deployment performs the drone placement many times and then selects the best placement.

We next confirmed the connections of the drones when using the proposed methods. Figures 12 and 13 show the drone deployment at time $t = 4$ for the ILP-based and the adjacent placement methods, respectively, where $L = 49$, $|\mathcal{U}| = 200$, and $|\mathcal{D}| = 6$. In both proposed methods, the drones were assigned to the connected positions. These results indicate that our proposed method can establish a communication path to the base station. In the ILP-based deployment method, the drones are deployed to the subareas with the highest total numbers of users. Moreover, from Figure 12, we can observe that there were branches in the topology of the drones because of the drone grouping. For the adjacent placement method, although the drones were not assigned to the subareas with the largest total numbers of users, we observed that the drones were assigned to subareas with many users.

Finally, we evaluated the effects of the probability ϵ in the adjacent placement method. Figure 14 shows the completion time T as a function of ϵ, where $L = 49$, $|\mathcal{U}| = 400$, and $|\mathcal{D}| = 6$. In all scenarios, for $\epsilon = 0.05$, the completion time T was the minimum and comparable to that of the random method. For the uneven scenario, the completion time was comparable to that of the ILP deployment method. In contrast, for large ϵ, the completion time T was long and the performance of the proposed method was almost the same as that of the spiral or rolling methods. These results indicate that the proposed method works well when ϵ is set as sufficiently small.

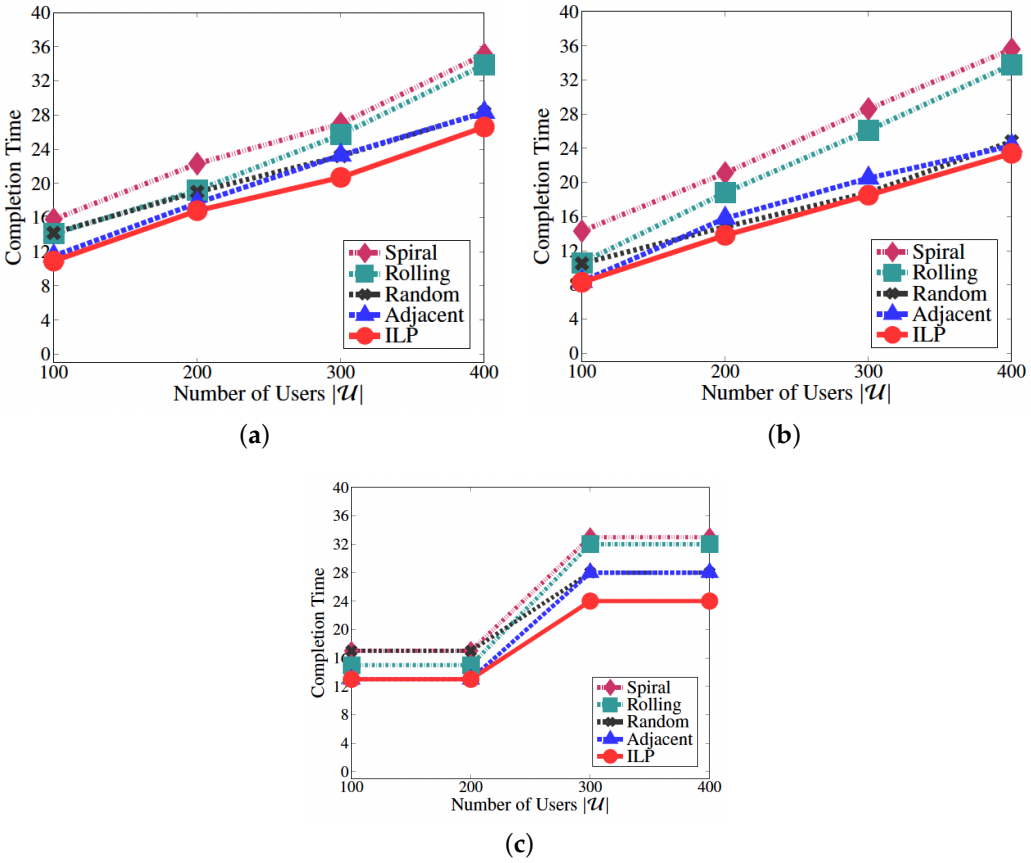

Figure 11. Completion time T as a function of the number of users. (**a**) Random scenario; (**b**) uneven scenario; (**c**) all-equal scenario.

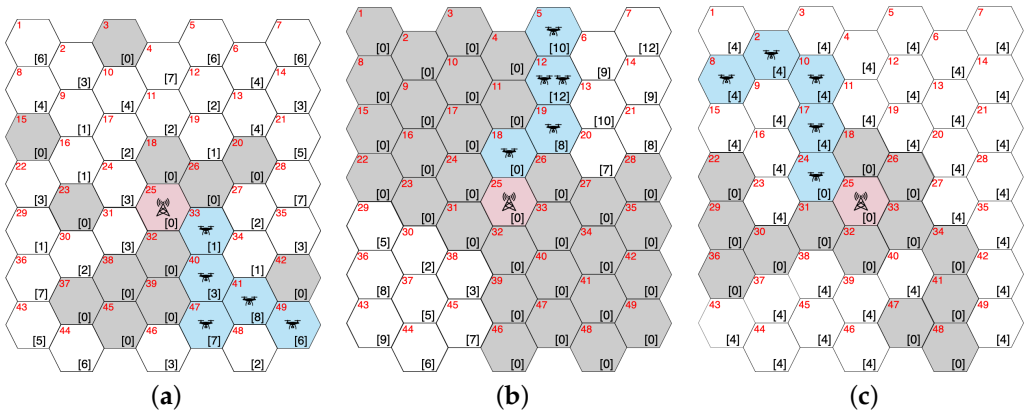

Figure 12. Drone deployment at time $t = 4$ using the ILP-based deployment method. (**a**) Random scenario; (**b**) uneven scenario; (**c**) all-equal scenario.

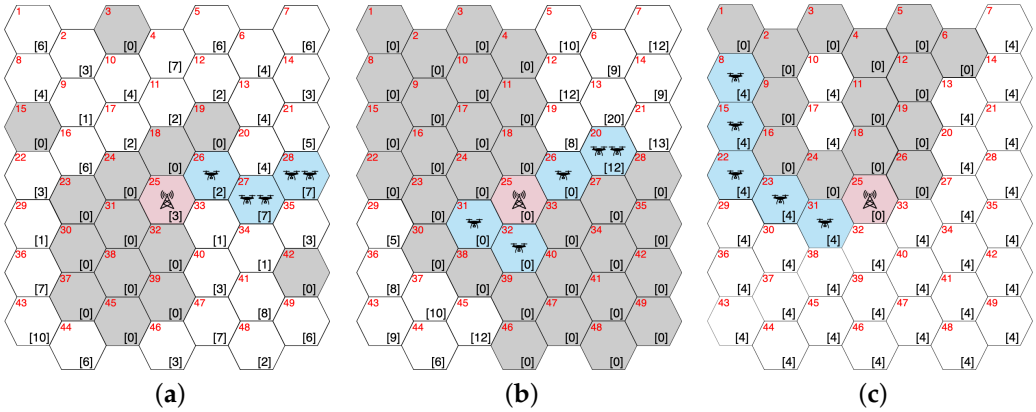

Figure 13. Drone deployment at time $t = 4$ using the adjacent deployment method. (**a**) Random scenario; (**b**) uneven scenario; (**c**) all-equal scenario.

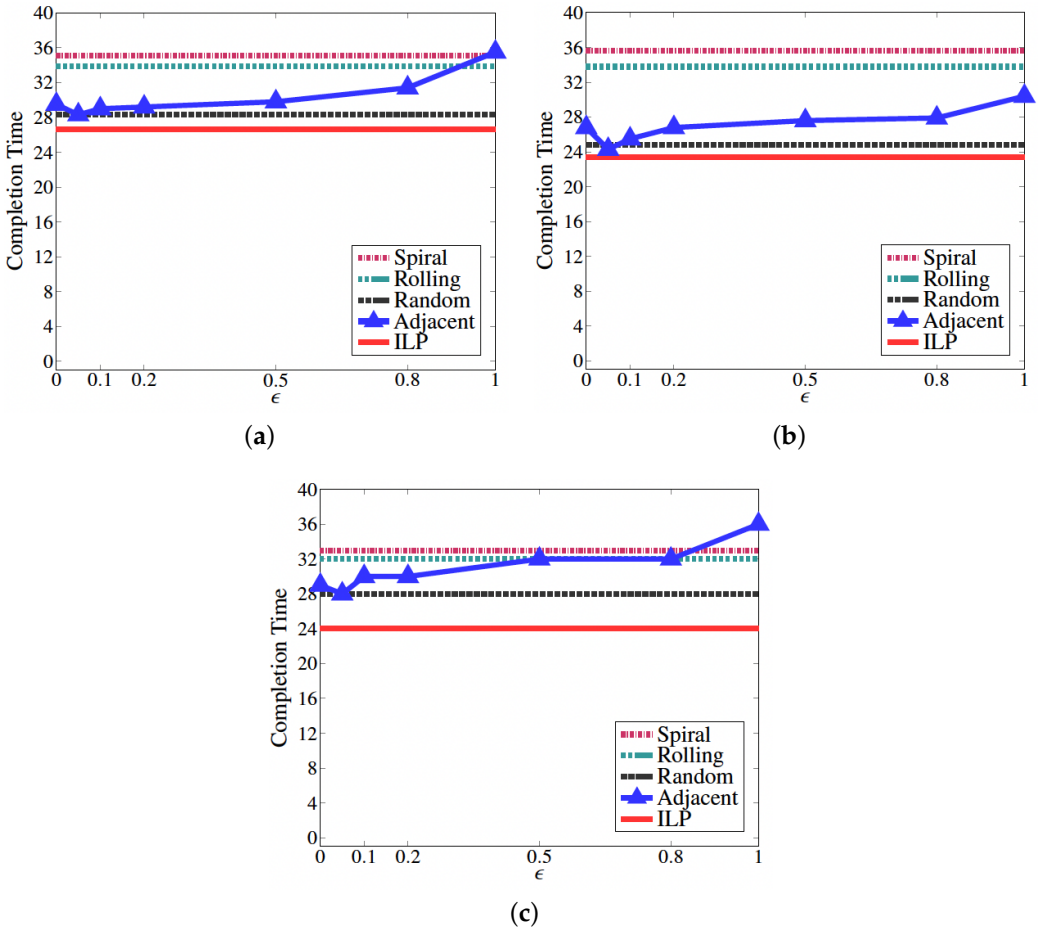

Figure 14. Completion time T as a function of ϵ. (**a**) Random scenario; (**b**) uneven scenario; (**c**) all-equal scenario.

5. Conclusions

In this paper, we addressed the drone deployment problem considering the connectivity of drone networks and proposed a deployment method based on ILP and the adjacent placement method. In the ILP-based deployment method, the position of drones is determined by solving an ILP problem, the objective function of which is to maximize the total number of users from whom data can be collected at each point in time. In the adjacent placement method, the largest area is selected with a high probability from the areas adja-

cent to where the drone has already been deployed, and a random selection is made with a low probability. Through numerical experiments, we evaluated the performance of the proposed methods and showed that they can be used to efficiently collect data. In this paper, we adopted the simple system model because it was sufficient to determine which areas drones should be deployed to construct connected drone networks. However, the collection of data from users on the ground could be improved by positioning the drones in a more suitable location. To address this problem, we can first identify the areas where a connected drone network can be constructed with our proposed method and then consider more complex propagation models to identify the best locations to collect data from users on the ground. We leave this combination problem for future work.

Author Contributions: Conceptualization, T.K., K.H. and C.P.; methodology, H.O. and T.K.; writing—original draft preparation, H.O.; writing—review and editing, T.K. and K.H.; visualization, H.O.; supervision, C.P. and J.C.; funding acquisition, T.K. and K.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by JSPS KAKENHI 21K11857 and 23K11077.

Data Availability Statement: Data are available from the authors upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, X.; Savkin, A.V. Networked Unmanned Aerial Vehicles for Surveillance and Monitoring: A Survey. *Future Internet* **2021**, *13*, 174. [CrossRef]
2. Yanmaz, E.; Yahyanejad, S.; Rinner, B.; Hellwagner, H.; Bettstetter, C. Drone networks: Communications, coordination, and sensing. *Ad Hoc Netw.* **2018**, *68*, 1–15. [CrossRef]
3. Lim, S.; Chae, S.H.; Lee, H. RE-ORA: Residual Energy-Aware Online Random Access for Improving the Lifetime of Slotted ALOHA-Based Swarming Drone Networks. *IEEE Access* **2021**, *9*, 45504–45511. [CrossRef]
4. Zhang, H.; Wang, G.; Lei, Z.; Hwang, J.N. Eye in the sky: Drone-based object tracking and 3d localization. In Proceedings of the 27th ACM International Conference on Multimedia, Nice, France, 21–25 October 2019; pp. 899–907.
5. Eichleay, M.; Evens, E.; Stankevitz, K.; Parker, C. Using the unmanned aerial vehicle delivery decision tool to consider transporting medical supplies via drone. *Glob. Health Sci. Pract.* **2019**, *7*, 500–506. [CrossRef]
6. Aurambout, J.P.; Gkoumas, K.; Ciuffo, B. Last mile delivery by drones: An estimation of viable market potential and access to citizens across European cities. *Eur. Transp. Res. Rev.* **2019**, *11*, 30. [CrossRef]
7. Lee, J.; Zhong, Z.; Kim, K.; Dimitrijevic, B.; Du, B.; Gutesa, S. Examining the applicability of small quadcopter drone for traffic surveillance and roadway incident monitoring. In Proceedings of the Transportation Research Board 94th Annual Meeting, Washington, DC, USA, 11–15 January 2015; Number 15-4184; p. 15.
8. Bae, M.; Yoo, S.; Jung, J.; Park, S.; Kim, K.; Lee, J.Y.; Kim, H. Devising mobile sensing and actuation infrastructure with drones. *Sensors* **2018**, *18*, 624. [CrossRef]
9. Restas, A. Drone applications for supporting disaster management. *World J. Eng. Technol.* **2015**, *3*, 316. [CrossRef]
10. Erdelj, M.; Natalizio, E.; Chowdhury, K.R.; Akyildiz, I.F. Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Comput.* **2017**, *16*, 24–32. [CrossRef]
11. Rashid, M.T.; Zhang, D.Y.; Wang, D. Socialdrone: An integrated social media and drone sensing system for reliable disaster response. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Toronto, ON, Canada, 6–9 July 2020, pp. 218–227.
12. Mishra, B.; Garg, D.; Narang, P.; Mishra, V. Drone-surveillance for search and rescue in natural disaster. *Comput. Commun.* **2020**, *156*, 1–10. [CrossRef]
13. Alshabtat, A.I.; Dong, L. Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks. *Int. J. Electr. Comput. Eng.* **2011**, *5*, 989–995.
14. Forsmann, J.H.; Hiromoto, R.E.; Svoboda, J. A time-slotted on-demand routing protocol for mobile ad hoc unmanned vehicle systems. *Unmanned Syst. Technol. IX SPIE* **2007**, *6561*, 530–540.
15. Lee, W.; Lee, J.Y.; Lee, J.; Kim, K.; Yoo, S.; Park, S.; Kim, H. Ground control system based routing for reliable and efficient multi-drone control system. *Appl. Sci.* **2018**, *8*, 2027. [CrossRef]
16. Park, S.; Kim, H.T.; Kim, H. Energy-efficient topology control for UAV networks. *Energies* **2019**, *12*, 4523. [CrossRef]
17. Shang, B.; Liu, L.; Ma, J.; Fan, P. Unmanned aerial vehicle meets vehicle-to-everything in secure communications. *IEEE Commun. Mag.* **2019**, *57*, 98–103. [CrossRef]
18. Chen, Y.; Zhang, H.; Xu, M. The coverage problem in UAV network: A survey. In Proceedings of the Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT), Hefei, China, 11–13 July 2014; pp. 1–5.

19. Ahmed, A.; Awais, M.; Akram, T.; Kulac, S.; Alhussein, M.; Aurangzeb, K. Joint placement and device association of UAV base stations in IoT networks. *Sensors* **2019**, *19*, 2157. [CrossRef] [PubMed]
20. Tuba, E.; Capor-Hrosik, R.; Alihodzic, A.; Tuba, M. Drone placement for optimal coverage by brain storm optimization algorithm. In *International Conference on Hybrid Intelligent Systems, Proceedings of the 17th International Conference on Hybrid Intelligent Systems (HIS 2017), Delhi, India, 14–16 December 2017* ; Springer: Cham, Switzerland, 2017; pp. 167–176.
21. Sabino, S.; Horta, N.; Grilo, A. Centralized unmanned aerial vehicle mesh network placement scheme: A multi-objective evolutionary algorithm approach. *Sensors* **2018**, *18*, 4387. [CrossRef]
22. Sabino, S.; Grilo, A. NSGA-II based Joint Topology and Routing Optimization of Mesh Networks with Flying Access Points. *Procedia Comput. Sci.* **2019**, *160*, 165–172. [CrossRef]
23. Al-Hourani, A.; Kandeepan, S.; Lardner, S. Optimal LAP altitude for maximum coverage. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 569–572. [CrossRef]
24. Song, H.Y. A method of mobile base station placement for high altitude platform based network with geographical clustering of mobile ground nodes. In Proceedings of the 2008 International Multiconference on Computer Science and Information Technology, Wisla, Poland, 20–22 October 2008; pp. 869–876.
25. Lyu, J.; Zeng, Y.; Zhang, R.; Lim, T.J. Placement optimization of UAV-mounted mobile base stations. *IEEE Commun. Lett.* **2016**, *21*, 604–607. [CrossRef]
26. Hu, Y.; Zhang, F.; Tian, T.; Ma, D. Placement optimisation method for multi-UAV relay communication. *IET Commun.* **2020**, *14*, 1005–1015. [CrossRef]
27. Iellamo, S.; Lehtomaki, J.J.; Khan, Z. Placement of 5G drone base stations by data field clustering. In Proceedings of the 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), Sydney, Australia, 4–7 June 2017; pp. 1–5.
28. Iranmanesh, S.; Abkenar, F.S.; Raad, R.; Jamalipour, A. Improving throughput of 5G cellular networks via 3D placement optimization of logistics drones. *IEEE Trans. Veh. Technol.* **2021**, *70*, 1448–1460. [CrossRef]
29. Matsuda, T.; Kaneko, M.; Hiraguri, T.; Nishimori, K.; Kimura, T.; Nakao, A. Adaptive direction control for UAV full-duplex relay networks using multiple directional antennas. *IEEE Access* **2020**, *8*, 85083–85093. [CrossRef]
30. IBM. CPLEX Optimizer. Available online: https://www.ibm.com/analytics/cplex-optimizer (accessed on 30 April 2023).