Towards Self-Organizing UAV Ad-Hoc Networks Through Collaborative Sensing and Deployment

Narangerelt Batsoyol and HyungJune Lee
Department of Computer Science and Engineering
Ewha Womans University, Seoul, South Korea
Email: hyungjune.lee@ewha.ac.kr

***Abstract*—In this paper, we consider an aerial ad-hoc network construction problem using UAVs in a disaster scenario. We aim to reconnect the communication-wise isolated urban area with the outside communication infrastructure. Our main goal is to perform both network exploration and relay deployment tasks at the same time by taking a progressive optimization toward a self-organizing network construction. We propose a novel UAV exploration-and-deployment algorithm that gradually explores the region of interests and achieves full network coverage in a fast manner. Then, we present an effective network refinement algorithm based on clustering that minimizes the number of UAVs for deployment by finding out essential UAVs, while keeping the similar network coverage performance. Simulation results demonstrate that our proposed scheme significantly reduces the execution time for network exploration and deployment compared to a baseline counterpart. Also, our cluster-based network refinement algorithm provides a very lightweight yet effective solution, well-balancing between UAV resource and computation overhead.**

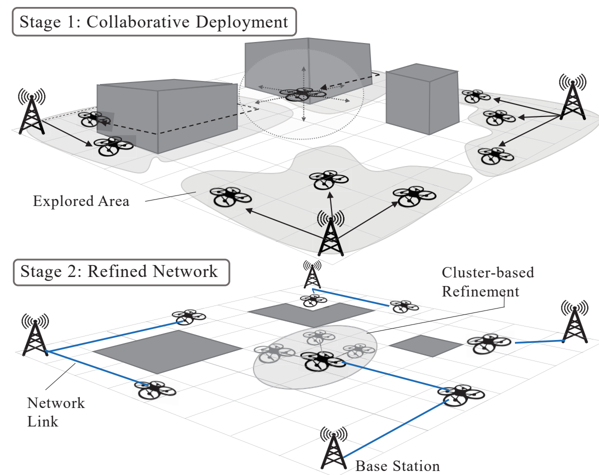


Fig. 1. Self-organizing UAV network construction with collaborative deployment and refined network construction procedures

I. INTRODUCTION

In disaster situations, a large-scale urban area occupied with various types of complex obstacles may suffer from pervasive communication failures due to severely collapsed infrastructure network. In this situation, deploying unmanned aerial vehicles (UAVs) to form an aerial ad-hoc network by reconnecting the geographically secluded area to its outside communication infrastructure (via nearby base stations) would be an effective solution for search and rescue tasks.

UAVs can be utilized as flexible networking and sensing devices suitable for information sharing and environment monitoring [9]. In a post catastrophic disaster, the affected area may be filled with numerous debris and collapsed building structures, and their geographical distribution and status are unknown. To construct a totally new ad-hoc network using only UAVs that connect to a nearby base station, it is essential to explore the cluttered region to perceive obstacle-free spaces for the UAV ad-hoc network.

Regarding efficient area exploration, various algorithms have been proposed in robotics field using multiple robots [5], [6], [11]. Bio-inspired algorithms simulating insect behaviors as well as swarm algorithms have been proposed for efficient exploration over region of interests (RoI). Although most of them work well for a relatively small-scale environment, their large-scale implementation is limited to some extent from the perspective of collaborative space exploration.

The problem of finding the optimal UAV placement for maximizing network coverage with the minimum usage of UAVs is similar to the antenna positioning problem or the radio network design problem [1], [2], [10]. These are combinatorial problems suffering from high computation complexity. Prior works on optimization algorithms for the antenna positioning problem are not practically feasible in general environments since they are mostly oriented towards small-scale indoor systems involving only few antennas. The existing studies of the large scale radio networks use evolutionary algorithm or genetic algorithms. However, these algorithms have high computation overhead because they try to find all possible combinations of antenna locations for obtaining an optimal solution, not well-feasible to real-world network environments.

Our previous work [3] has tackled this problem in three dimensions (3D) by first exploring the unknown urban area prior to network construction. This work completes to obtain the global knowledge of obstacles' geographical distribution and the height map of them. Then, it finds the optimal placement for UAVs in the 3D space to obtain the full network coverage with the minimum number of UAVs. However, due to its inherent two-phase framework, it takes some amount of time for UAVs to start the actual network operation.

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2015R1D1A1A01057902 and NRF-2018R1A2B6004006).

In this paper, we focus on constructing an ad-hoc network purely based on UAVs that can extend the network coverage over RoI and also connect to the outside communication infrastructure via base stations. UAVs perform both environment exploration and relay deployment tasks at the same time by taking a progressive optimization toward a self-organizing network construction. UAVs move progressively towards unexplored areas, while maintaining their intra-communication among connected UAVs, and also their inter-communication with at least one base station. We aim to solve the problems of 1) how to disperse UAVs throughout the unknown cluttered environment so that the network can effectively extend its network coverage over RoI as much as possible while staying within communication range; and 2) how to minimize the number of deployed UAVs while still not critically affecting its network coverage performance.

We propose a novel UAV exploration-and-deployment algorithm that gradually explores the RoI while maintaining the connection with the base station until an almost full network coverage is achieved in a fast manner. Our algorithm disperses UAVs from base stations with a gradual expansion toward their maximum coverage while keeping the minimal network connection with at least one base station. In the middle of this procedure, UAVs can perform an obstacle exploration for understanding its size and geographical distribution. After this procedure, UAVs become aware of all the accessible areas where a UAV can be deployed to form its self-organizing ad-hoc network with other UAVs and base stations.

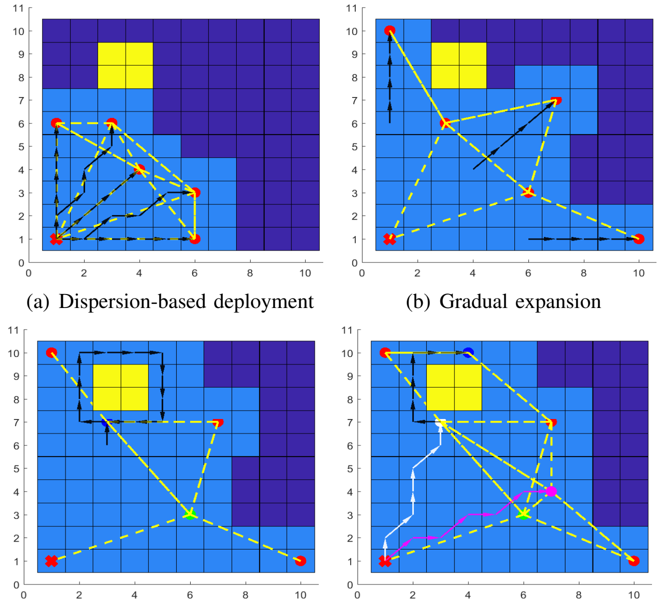
We further present an optimal network refinement process that minimizes the number of deployed UAVs by cutting out some redundant UAVs with the coverage overlapped. We devise a simple yet efficient clustering-based refinement algorithm for UAVs, achieving the maximum coverage even after discarding the least essential UAVs.

Our paper is organized as follows: After presenting our system model in Sec. II, we present our collaborative deployment in Sec. III and network refinement in Sec. IV. We validate our algorithms in Sec. V and finally conclude our work in Sec. VI.

II. System Model

We consider the problem of constructing a self-organizing ad-hoc network with almost full-coverage using UAVs in a large-scale unknown urban environments. Our goal is to perform an efficient gradual exploration for UAVs to cover the entire RoI in a fast manner, while maintaining the connectivity with the base station through each other and dispatch UAVs for extending wireless coverage toward terrestrial areas. We aim to find out a converged point of the minimum number of UAVs to deploy and their final deployed positions.

We assume that UAVs can communicate with other UAVs and base stations using a wireless radio such as 802.11 or 802.15.4 within its effective radio range R. For simplicity, we focus on two-dimensional (2D) flight space with minimal space exploration by deploying all UAVs at the same height. UAVs are allowed to traverse over virtual grid cells, where the Cartesian coordinate system with xy axis portraying a cross-sectional 2D representation from the above is used. Depending on an obstacle's occupancy at a grid cell, each grid cell is specified by 0 for the obstacle-free case or 1 for the obstacle case.



(a) Dispersion-based deployment (b) Gradual expansion (c) Obstacle exploration (d) Additional UAV deployment

Fig. 2. Collaborative deployment on 10×10 grid size with the communication range of $R = 5$, where 5 UAVs are deployed at initial state. Circles represent deployed UAVs, and dashed lines for valid communication links where one base station marked with black cross is located at the left bottom corner of RoI. Arrows represent UAV navigation paths. Yellow cells represent obstacle cells, while light blue cells are already-sensed cells, and dark blue cells are unexplored cells. Red circles are initially deployed UAVs; blue circle: obstacle exploring UAV, white circle: additional successor UAV, and purple circle: additional UAV for space exploration.

It is assumed that base stations are located at the four corners of a rectangular RoI area, and can communicate with UAVs with the same radio. As long as an information data is deliverable to a base station, all other network devices that have connected to a base station can share the information via the outside base station network. In case that the line-of-sight communication link between two UAVs is interfered with any obstacle, we consider it as a communication-incapable link. For any obstacle-free transmission link, its maximum communication range is given by R. We also assume that UAVs can sense an obstacle's occupancy in its directly adjacent cells using radar sensors or laser scanners [8].

The problem of constructing almost full-coverage aerial ad-hoc networks can be divided into two sub-problems: 1) collaborative sensing-and-deployment of UAVs for exploration and construction of an entire new self-organizing network and 2) network refinement for reducing the number of deployed UAVs, as illustrated in Fig. 1.

III. Collaborative UAV Deployment

After a disaster scenario, it is important to reconnect a totally secluded urban area with the outside environment. UAVs can be used to construct a new ad-hoc network by extending wireless coverage toward parts of the area, and connecting with other UAVs and possibly with a base station directly or indirectly (via multi-hop connection). To find out affected sub-regions where previous and new obstacles are relocated, the network exploration procedure is usually a prerequisite for the subsequent network construction. However, it may result in losing the golden hour for emergency rescue tasks.

Instead, we would rather have a combined approach for both exploration and network construction at the same time, even if its ongoing network construction is not perfect at that point. In this way, we can achieve a more responsive network coverage over RoI.

To rapidly launch a network construction procedure, UAVs are dispersed from their corresponding base stations to each different predefined angle direction, as shown in Fig. 2(a). By letting UAVs equally cover their RoI from a base station, we divide the RoI into a set of sub-areas where a base station is located within one sub-area. Once some UAVs are dispersed from one base station, they need to only explore their home sub-area that the base station belongs to.

Algorithm 1 Dispersion-Based Gradual Deployment

Input: *$UAVlist$, Dispersion angles θ, Cartesian coordinate*
Output: *Global map H, final UAV locations*
// If the cell at xy is an obstacle, then $C_{xy} = 1$, else $C_{xy} = 0$
$CircleBorder$ cells found by Midpoint Circle Algo with radius R
$DestCells =$ cells among $CircleBorder$ where $(y/x) \approx tan(\theta)$
Send $UAVlist$ to $DestCells$, by finding the path by Bresenham's
while (H is not completed up to in-border cells) **do**
  // I. Dispersion-based exploration
  while (UAVs connected to base station && no obstacle cell detected && $UAVgoing$ connected to its $fixedNeighbor$)) **do**
    $UAVgoing$ follows the path in θ;
    Update *sensedCellList* and $H_{xy} = 0$
    if (can communicate with other UAVs) **then**
      For the cells along the communication path, do $H_{xy} = 0$
    else if (cannot communicate within R distance) **then**
      Call for additional UAV to assign the obstacle exploration
    end if
    if (no UAVs can move farther) **then**
      if (on θ, $R/2$ path is still left to be explored) **then**
        Call for additional UAV
      end if
    end if
  end while
  // II. Gradual expansion
  while (All UAVs' roles not decided) **do**
    For each UAV, find its space expansion by triangulation.
    Select the UAVs with the greatest sensed cell expansion as $UAVgoing$, the selected neighbors as $fixedNeighbor$
  end while
  // III. Obstacle exploration
  if (At least one $C_{xy} = 1$ in sensor range) **then**
    Encircle the obstacle base by following $C_{xy} = 1$ in the clockwise direction
    Update *sensedCellList*
    Save in $H_{xy} = 1$, where $C_{xy} = 1$, else 0;
    Call for additional UAV
    Go to the farthest explored cell on θ
  end if
  // IV. Additional UAV deployment: i) Called for successor
  if (Obstacle Exploring UAV called for its successor) **then**
    Go to the calling UAV's location by finding the shortest path
  end if // ii) Called for indirect detection of obstacle
  if (Called by UAVs at θa and θb) **then**
    Send a UAV to $\theta c = min(\theta a, \theta b) + |\theta b - \theta a|/2$
    Perform obstacle exploration
  end if // iii) Called for space exploration
  if (called on θ) **then**
    Go to the farthest explored cell on θ
  end if
end while

A group of UAVs gradually expand their network formation as long as their multi-hop wireless link is connected to their base station, as in Fig. 2(b). Whenever a UAV visits a cell, it records the exploration event into its *visitedCellList*. In case that the UAV detects an obstacle at its surrounding cells, we assign one dedicated UAV for the obstacle exploration task by diagnosing how large and where the obstacle is located as in Fig. 2(c). Each UAV continues to perform its gradual expansion, while recording the surrounding obstacle's occupancy information of its surrounding cells into its *sensedCellList*, until it reaches out to the border of its undertaken sub-area. If some more UAVs are necessary to maintain connectivity or expand space exploration, supplementary UAVs can be dispatched from base stations to the spots in need, as in Fig. 2(d).

When two UAVs encounter within their radio range, they exchange their own *visitedCellList* and *sensedCellList*. Since our work operates on top of a grid cell topology focusing on the cell level rather than the normal point level, we use the Bresenham's line algorithm [4] and the Midpoint circle algorithm [7] for converting from lines and curves to their corresponding pixel-like cells.

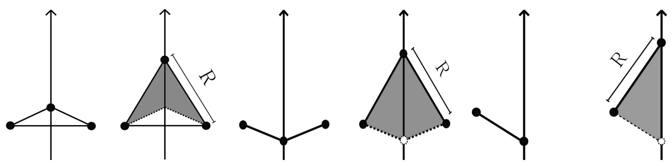
A. Dispersion-Based Exploration

We let UAVs be dispersed from a base station to different predefined angle directions. For example, 5 UAVs are dispersed from a base station located at the left bottom of RoI to 5 different angle directions $\theta = 0°$, 30°, 45°, 60°, and 90°, as in Fig. 2(a). They continue to follow their own predefined direction until they reach the farthest cell where UAV can keep the connectivity with the base station. On the grid cell topology, we find the circle border cells within the radio range R from the origin of the base station based on the Midpoint circle algorithm. Among these border cells, we choose one with (x, y) that lies on the predefined angle, i.e., $\tan\theta \approx y/x$. Once they determine their farthest cells as the destination cells, the navigation path from the base station to their respective destination cell is obtained by using Bresenham's line algorithm. When UAVs reach their destination cells, they start the gradual expansion process.

B. Gradual Expansion

Once all of UAVs within a sub-area reach their farthest cells, we check if some UAVs can be further expanded, as long as its connection to the base station stays via either one-hop connection or multi-hop connection, as in Figs. 2(a) and 2(b). To find out an optimal set of UAVs that can be used for their maximum expansion, we may attempt to list up all combinations of single UAV expansion. However, this approach suffers from tremendous computation complexity, making it infeasible in practice.

Our approach is based on an iterative gradual expansion of a single UAV at a time. The goal of this process is to find out a set of UAVs that perform gradual expansion and explore as many sensed cells as possible in a fast manner. To quantify how much one gradual expansion can extend its sensing area, we use a triangulation method as illustrated in Fig. 3. This



(a) Subtractive Triangulation (b) Additive Triangulation (c) Single Triangulation

Fig. 3. Triangulations for three different cases. The arrow represents a UAV's dispersion angle; the circle is a UAV; a solid line stands for a valid communication link. The grey area is the approximate expanded sensing area.

method provides some important rules of thumb for expansion, while satisfying the distance constraint.

We first choose one UAV that will provide the largest sensing area expansion while other UAVs remain at the same positions, and its expansion does not break other UAVs' connection to its base station. If other UAVs get disconnected from the base station due to the UAV's expansion, that UAV can not be used for gradual expansion and stays at the current cell for the rest of the period. For example, the green UAV as in Fig. 2(d) does not move for further expansion from Fig. 2(a) to Fig. 2(d) since the UAV operates as the only relay node for connecting other UAVs with the base station. After the first UAV has been expanded for its coverage, we select the next UAV for gradual expansion based on the sensing area expansion size. We continue this process until all the UAVs are considered for the expansion and complete one round of expansion. In case that there are multiple UAVs with the same sensing area expansion, we randomly choose one among them.

We provide some more detailed procedures for the area expansion based on a triangulation method.

1) Triangulation for Two Neighbors: In case of a UAV, UAV_k that is directly connected to two or more UAVs, we select its two neighbor UAVs that are connected each other. This selection provides the most effective sensing area expansion to UAV_k. In this case, we apply the subtractive triangulation method, as in Fig. 3(a). A triangle is formed with two selected neighbor UAVs, and UAV_k calculates its original triangle size $Area_{origin}$ based on the following formula: $Area = \sqrt{p(p-a)(p-b)(p-c)}$, where $p = (a+b+c)/2$ and a, b, c are each side length of the triangle, respectively.

Now that UAV_k moves as far as possible toward its dispersion angle until it keeps its connection with two neighbor UAVs (within the radio range R), we compute the surface area of the newly created triangle as $Area_{dest}$. Therefore, the new explorable area estimate by the expansion of UAV_k is given by $S_k = Area_{dest} - Area_{origin}$, as illustrated with the grey area in Fig. 3(a).

In case that two closest neighbors of UAV_k are not connected each other, we apply the additive triangulation method, as in Fig. 3(b). We calculate the new explorable area estimate as $S_k = Area_{origin} + Area_{dest}$, as illustrated with the grey area in Fig. 3(b).

We compute S_k for all possible neighbor UAV pairs of UAV_k and choose one pair of two neighbor UAVs with the largest S_k.

2) Triangulation for One Neighbor: For UAVs that have only one neighbor UAV or have its dispersion angle of $\theta = 0°$ or $90°$, we select one best neighbor UAV that can sense more new cells. UAV_k moves as far as possible toward its dispersion angle, while keeping the connection with its neighbor UAV. Three vertexes of the new cell position, the original cell position of UAV_k, and the current cell position of its neighbor UAV form a triangle, and we compute the area size as the new explorable area estimate, S_k.

We compute S_k for all neighbor UAVs and choose one with the largest value.

C. Obstacle Exploration

In the middle of dispersion or expansion process, if a UAV senses an obstacle located at its adjacent cells within the sensing range, it stops at the current cell, as the blue UAV in Fig. 2(c). Now the UAV changes its mode to the obstacle exploration and fully encircles around the obstacle in the clockwise direction, while keeping updating its *sensedCellList*. The UAV stops if it comes back to its original starting cell.

Once the UAV obtains the knowledge of how large and where an obstacle is geographically distributed at its *sensedCellList*, it is shared with other UAVs as well as its connected base station. Finally, the UAV prepares to leave for its original farthest obstacle-free cell with its original dispersion angle by following the shortest path. Before flying to the cell, the UAV (marked with blue UAV in Fig. 2(c)) calls for its successor UAV (marked with white UAV in Fig. 2(d)) from a base station to replace its location to cover the current area. The original UAV now flies to the destination cell, while its successor UAV stays for the rest of time.

If multiple UAVs turn out to encircle the same obstacle and request their successor UAVs to extend the wireless coverage toward the obstacle area, only one successor UAV is dispatched.

D. Additional UAV Dispatch

In case that no UAVs can make further progress for their expansion, and more than $R/2$ distance path is still left to be explored as in Fig. 2(c), an additional UAV (marked with purple color in Fig. 2(d)) is dispatched to its originally calculated farthest cell toward the corresponding dispersion angle for letting it complete the maximum expansion.

We repeat all previous steps until all the cells including in-border cells, which are cells with two cell distance from the center border of its sub-area, are explored. As in Fig. 2(d), all the cells with light blue color are explored except two-cell wide boundary cells with dark blue color. To the end, we complete the collaborative exploration-and-deployment algorithm with total network coverage of 100%, as in Algorithm 1.

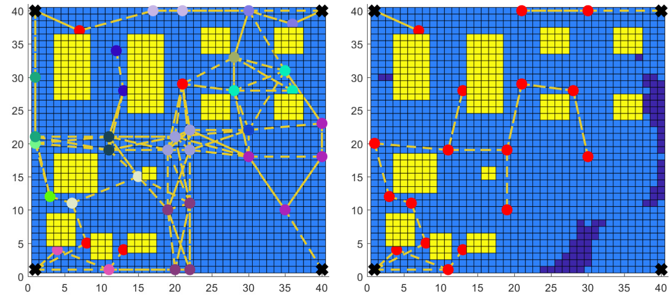
IV. Network Refinement Procedure

Once the exploration-and-deployment process takes place in each sub-area where one base station is located at its corner, we achieve 100% full network coverage Fig. 4(a). Since many UAVs have been leveraged for various roles of the initial dispersion, the obstacle exploration, and additional UAV supplement, there are inevitable overlapped network coverage to some parts of terrestrial cells.

Algorithm 2 Cluster-Based Refinement

Input: $UAVlist$, *Network Information*, R
Output: $refinedUAVlist$, *Refined Network Information*
// P: coverage overlap ratio, M: intra-cluster minimum coverage ratio
// I. Overlap-based clustering
for (k in $UAVlist$, starting from the UAV with the largest number of one-hop UAVs) **do**
  for ($clusterList$) **do**
    Populate $cluster$ with UAV_k
    Find $CoveredList_{UAV_k}$, covered cells by UAV_k
    for (Directly connected neighbors of UAV_k) **do**
      Starting from closest located neighbor:
      Find $CoveredList_{neighbor}$
      Find $OverlapCells$ between $CoveredList_{UAV_k}$ and $CoveredList_{neighbor}$
      if ($|OverlapCells| >= P \times |CoveredList_{UAV_k}|$) **then**
        Populate $clusterList(k)$ with this neighbor
      end if
    end for
  end for
end for// Clear up process for cluster, eliminating duplicate ones and any cluster which is sub-member of another cluster
// II. Intra-cluster refinement
for (G in $clusterList$) **do**
  //Compute the coverage of cluster G
  $Coverage_G = \cup coveredCellList_{UAV_k}$, where $UAV_k \subset G$
  $flagStop = empty$
  while ($len(G)$ >=1) **do**
    If the first element of G is in $flagStop$, then break
    Starting from the first element UAV_i of G:
    $Coverage_{tempG} = \cup coveredCellList_{UAV_k}$, where $\{G \setminus UAV_i, UAV_k \subset G\}$
    if ($|Coverage_{tempG}| < M \times |Coverage_G|$) **then**
      break;
    end if
    if (all UAVs connected to base station) **then**
      Remove UAV_i from G, $UAVlist$ and $clusterList$
    else
      Put UAV_i at the end of G and in $flagStop$
    end if
  end while
end for
// III. Inter-cluster refinement
//Compute total coverage $Coverage_{Total} = \cup coveredCellList_{UAV_k}$, where $UAV_k \subset UAVlist$
$flagStop = empty$
while ($len(UAVlist)$ >=1) **do**
  If the first element of $UAVlist$ is in $flagStop$, then break
  Starting from the first element UAV_i of $UAVlist$:
  $Coverage_{tempTotal} = \cup coveredCellList_{UAV_k}$, where $\{UAVlist \setminus UAV_i, UAV_k \subset UAVlist\}$
  if ($|Coverage_{tempTotal}| < 0.9 \times |Coverage_{Total}|$) **then**
    break;
  end if
  if (all UAVs connected to base station) **then**
    Remove UAV_i from $UAVlist$
  else
    Put UAV_i at the end of $UAVlist$ and in $flagStop$
  end if
end while

We want to find out some essential UAVs that play more effective roles of network coverage and UAV network connection toward base stations. The challenge is to find and remove ineffective UAVs, isolated UAVs, or network holes without significant coverage degradation. Instead of listing up all possible combinations for the removal of some selected UAVs from the current deployment state, requiring too much complexity, we propose a lightweight cluster-based refinement algorithm. Our algorithm suggests a set of the least essential UAVs and removes them from the current network, while the overall network coverage performance is fairly maintained.



(a) Full-deployment state (b) After-refinement state

Fig. 4. Full deployment state and after refinement state on 40×40 grid size test with communication range $R = 10$ where base stations are located at four corners represented by black crosses. Circles represent deployed UAVs that belong to the same cluster with the same color, and dashed lines are valid communication links toward at least one base station. Yellow cells represent obstacle cells, while dark blue cells are unexplored cells by any UAVs.

Our network refinement process is divided into three steps: 1) overlap-based clustering such that each cluster is formed based on the criterion of wireless cell coverage overlap, 2) intra-cluster refinement that removes ineffective UAVs within a cluster, and 3) inter-cluster refinement across clusters, as described in Algorithm 2.

A. Overlap-Based Clustering

Given the deployment state of all the UAVs, we let the UAVs form clusters based on how many cells are redundantly overlapped with wireless coverage. First, each UAV finds its directly connected one-hop neighbor UAVs. We prioritize UAVs with respect to the number of one-hop neighbor UAVs to determine the clustering order. The first UAV with the largest number of one-hop neighbor UAVs starts clustering.

Suppose that UAV_k turns out to cover nearby cells within the radio range, in $coveredCellList_{UAV_k}$. The UAV checks if how much its own covered cell is overlapped with its one-hop neighbor UAV's. If the overlap ratio is larger than overlap $P\%$ (e.g., 60%, 70%, etc.), UAV_k, which is a cluster head, selects the corresponding one-hop neighbor UAV as its cluster member. Once the first cluster is formed, the next UAV with the second largest number of one-hop neighbor UAVs continues this process as a cluster head. It should be noted that even after completing this process, UAVs that do not belong to any cluster will never be removed during the refinement.

B. Intra-Cluster Refinement

Once the clustering procedure is completed, we perform the first refinement procedure within each cluster. We start from the largest cluster by removing one UAV at a time with the priority of hop distance from the cluster head, as long as a cluster still keeps at least $M\%$, *intra-cluster minimum coverage ratio* (e.g., 80%) of its original coverage cells. A UAV can not be removed if its removal makes other UAVs disconnected from base stations. If a UAV belonging to multiple clusters is removed from a certain cluster, it is also removed from all the belonging clusters. All the removed UAVs go back to their closest base station.

C. Inter-Cluster Refinement

We perform the last refinement procedure across clusters. Now we form one higher-level cluster which all the remaining UAVs belong to. We try to randomly choose one UAV and remove it at a time as long as its removal still satisfies the *inter-cluster minimum coverage ratio* (e.g., 90%) of the original total coverage. The final UAV deployment after all the refinement process is illustrated in Fig. 4(b).

V. Evaluation

We evaluate our proposed algorithms in a simulated territory environment of $120{\times}120$ m^2. To simulate the urban territory environment, we randomly generate obstacles over the RoI area, as in Fig. 4. We use a grid cell topology where the size of a virtual cell is $3{\times}3$ m^2, and the total number of cells is $40{\times}40$. The RoI area is divided into 4 sub-areas where a base station is located at the corner of each sub-area. We use a unit disk radio model with the radio range R of 30 m. The flying speed of UAVs is simulated with 3 $meter/sec$ (or 1 $cell/sec$), and the height for UAV exploration and deployment is fixed to $R/2$ from the ground.

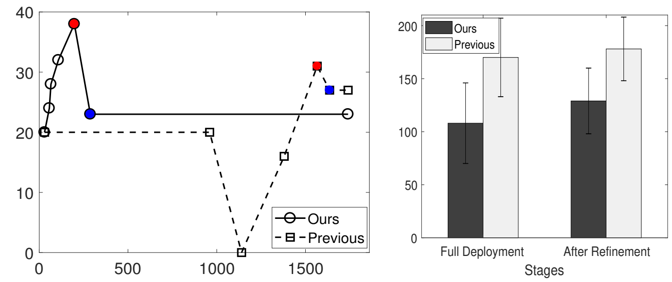
To obtain statistically meaningful evaluation results, we use 100 randomly generated test dataset with each different obstacle distribution and provide the average and the standard deviation with error bars, if appropriate. Our evaluation is divided into two parts of collaborative UAV deployment and network refinement performance. To validate our collaborative UAV deployment algorithm, we measure the required deployment time before the actual operation, the number of UAVs over time, and exploration efficiency in travel distance. We compare the proposed algorithm against our previous work [3]. For network refinement, we first show how our cluster-based refinement algorithm behaves by varying tuning parameters. Then, the performance is compared with a previous evolution-based optimal approach [2], in terms of the number of UAVs, computation overhead with respect to total network coverage ratio.

We have used the parameters of overlap threshold $P = 60\%$ and intra-cluster minimum coverage ratio $M = 80\%$ for our refinement algorithm, unless otherwise noted.

A. Collaborative Deployment Performance

We compare our proposed algorithm against our previous work [3]. Our previous algorithm tackles the same problem in 3D with two separate phases. In the former phase, UAVs completely explore every cell over the whole RoI area and construct the height map over all the explored cells. In the latter phase, it finds effective deployment positions of UAVs in 3D with a hierarchical way. This paper focuses on constructing one-phase quick UAV networks in the relatively simple 2D space over RoI with minimal obstacle exploration. We compare them in terms of the efficiency of UAV exploration and UAV resource under the same resulting coverage performance (98% used in Fig. 5).

We have initially deployed 5 UAVs with the dispersion angles $\theta = 0°$, $30°$, $45°$, $60°$, and $90°$ from each base



(a) Average number of UAVs currently in RoI as of the execution time

(b) Average travel distance per UAV

Fig. 5. Efficiency of UAV exploration and UAV resource for our work vs. previous work, under the same coverage performance condition. The red points represent the end of the deployment process, while the blue points do right after the refinement process.

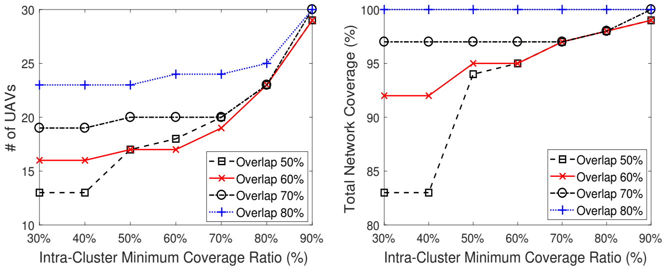
stations, and the total number of UAVs is 20 at the beginning of our experiment. As in Fig. 5, we show the dynamics of the average number of UAVs over execution time for our work and the previous work. The red points represent the end of the deployment process, while the blue points do right after the refinement process. Although both algorithms have achieved 100% full network coverage at the end of the deployment process, the previous work has spent 1636 sec in total, consisting of 960 sec for the full exploration, 180 sec for returning back to base stations, 426 sec for the full deployment, and 70 sec for the refinement. On the other hand, our work has spent 288 sec in total, consisting of 198 sec for exploration-and-full-deployment and 90 sec for the refinement. This means that this work outperforms the previous work with a factor of 5.7 in terms of responsiveness until reaching the steady-state deployment performance.

Regarding the usage of UAV resource, our algorithm has required 7 more UAVs than the previous work up to the full deployment. However, after the refinement process, our algorithm has reduced 15 UAVs by filtering out some ineffective UAVs and has used an even smaller number of UAVs than the previous work. This implies that our refinement process has effectively been embedded with our dispersion-based deployment.

From the perspective of travel distance of UAVs, we measure the average travel distance per UAV for all the UAVs including removed UAVs after refinement. Since the previous work takes some amount of time for 3D obstacle exploration to obtain the height, the 3D obstacle exploration has been replaced by its 2D obstacle exploration by just encircling it at the ground, as done in this work for fair comparison. Fig. 5(b) shows that UAVs using our algorithm has traveled much less compared to the previous work for both full-deployment and after-refinement cases, with a factor of up to 1.6. This result shows the efficiency of gradual exploration and deployment by UAVs in one phase.

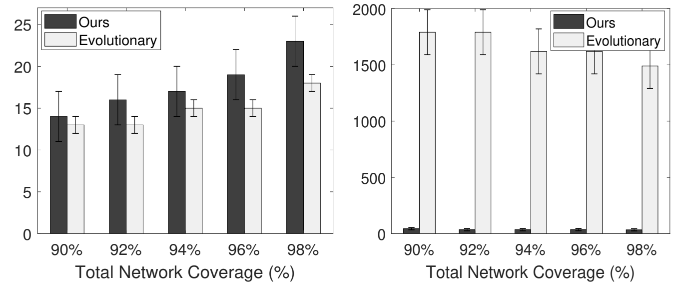
B. Network Refinement Performance

First, we show how our cluster-based network refinement algorithm works with respect to our internal parameters, i.e., overlap $P\%$ and *intra-cluster minimum coverage ratio* $M\%$



(a) Average # of UAVs (b) Total network coverage ratio

Fig. 6. Average number of UAVs and total network coverage ratio at the end by varying the overlap threshold for clustering, and the intra-cluster minimum coverage ratio for refinement under the scenario of 38 deployed UAVs on average before the refinement.



(a) Average # of UAVs (b) Average running time

Fig. 7. Average number of UAVs and average running time for our refinement algorithm vs. evolution-based optimal algorithm under the scenario of 38 deployed UAVs on average before the refinement.

in terms of the number of UAVs and total network coverage, as in Fig. 6. We use the same test scenario as in Fig. 5 where the number of deployed UAVs is 38 at the full-deployment, before the refinement. From the perspective of UAV resource, as the overlap $P\%$ threshold increases, we need more UAVs to meet a higher clustering requirement (as in Fig. 6(a)), leading to higher total network coverage performance in return (as in Fig. 6(b)). Once the clustering process is completed, as the criterion on whether a certain UAV can be discarded becomes more stringent (i.e., as *intra-cluster minimum coverage ratio* $M\%$ increases), we end up with more UAVs to deploy and its resulting higher total network coverage performance. In our algorithm, the combination of using the coverage overlap of 60% and the intra-cluster minimum coverage ratio of 50% provides a practical trade-off point where the total network coverage of 95% can be achieved using only 17 UAVs in the end.

Lastly, we compare our algorithm with an evolution-based optimal placement algorithm [2] to validate the refinement performance in Fig. 7. We have adapted the evolutionary algorithm to our refinement algorithm so that we effectively find and remove only one UAV per each iteration that turns out to be the one with the minimal contribution to the entire network coverage. On the other hand, the evolutionary algorithm finds all possible combinations of UAVs that can lead to the largest total network coverage. Based on this, it removes relatively ineffective multiple UAVs per each iteration. As shown in Fig. 7(a), under the total network coverage requirement, the evolutionary algorithm reduces more unnecessary UAVs than ours, resulting in more efficient usage of UAV resource. However, the evolutionary algorithm suffers from tremendously high computation overhead compared to ours in Fig. 7(b). This result implies that our refinement algorithm provides a much more lightweight practical impact.

VI. Conclusion

We have presented a self-organizing aerial ad-hoc network construction using UAVs with a necessary full-coverage requirement in a large-scale unknown urban environment. We have proposed an iterative one-phase exploration-and-deployment algorithm to construct a new ad-hoc network over RoI that can reconnect with the existing outside network infrastructure in a more responsive fashion. Also, we have incorporated an effective cluster-based network refinement algorithm that filters out unnecessary deployed UAVs with significant coverage overlap with other essential UAVs, without critically hindering the overall network coverage performance. Our experimental results show that our proposed algorithms significantly outperform recent works in terms of the number of UAVs for deployment, execution time, and computation complexity.

For future work, we may come up with an adaptive dispersion scheme for UAVs to continuously determine the next visiting cell based on the learned environment dynamics. In this way, we can more evenly distribute only the essential UAVs into an even more complicated obstacle environment.

References

[1] K. Al Nuaimi and H. Kamel. A survey of indoor positioning systems and algorithms. In *Innovations in information technology (IIT), 2011 international conference on*, pages 185–190. IEEE, 2011.

[2] E. Alba. Evolutionary algorithms for optimal placement of antennae in radio network design. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 168. IEEE, 2004.

[3] N. Batsoyol, Y. Jin, and H. Lee. Constructing full-coverage 3D UAV ad-hoc networks through collaborative exploration in unknown urban environments. In *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018.

[4] J. E. Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[5] M. S. Couceiro, D. Portugal, and R. P. Rocha. A collective robotic architecture in search and rescue scenarios. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, pages 64–69. ACM, 2013.

[6] M. S. Couceiro, R. P. Rocha, and N. M. Ferreira. A novel multi-robot exploration approach based on particle swarm optimization algorithms. In *Safety, Security, and Rescue Robotics (SSRR), 2011 IEEE International Symposium on*, pages 327–332. 2011.

[7] J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes, and R. L. Phillips. *Introduction to computer graphics*, volume 55. Addison-Wesley Reading, 1994.

[8] N. Gageik, P. Benz, and S. Montenegro. Obstacle detection and collision avoidance for a UAV with complementary low-cost sensors. *IEEE Access*, 3:599–609, 2015.

[9] F. Nex and F. Remondino. UAV for 3D mapping applications: a review. *Applied geomatics*, 6(1):1–15, 2014.

[10] M. Vasquez and J.-K. Hao. A heuristic approach for antenna positioning in cellular networks. *Journal of Heuristics*, 7(5):443–472, 2001.

[11] J. Yuan, Y. Huang, T. Tao, and F. Sun. A cooperative approach for multi-robot area exploration. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1390–1395. IEEE, 2010.