

## Research Article

# Optimal Multirobot Coverage Path Planning: Ideal-Shaped Spanning Tree

Chunqing Gao <sup>1</sup>, Yingxin Kou,<sup>1</sup> Zhanwu Li <sup>1</sup>, An Xu,<sup>1,2</sup> You Li <sup>1</sup> and Yizhe Chang <sup>1</sup>

<sup>1</sup>Air Force Engineering University, Baling Road, #1, Xi'an, Shaanxi 710038, China

<sup>2</sup>Northwestern Polytechnical University, West Youyi Road, #127, Xi'an, Shaanxi 710072, China

Correspondence should be addressed to Chunqing Gao; gaochq@163.com

Received 26 February 2018; Accepted 12 September 2018; Published 25 September 2018

Academic Editor: Alessandro Naddeo

Copyright © 2018 Chunqing Gao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The present paper attempts to find the optimal coverage path for multiple robots in a given area including obstacles. For single robot coverage path planning (CPP) problem, an improved ant colony optimization (ACO) algorithm is proposed to construct the best spanning tree and then obtain the optimal path, which contributes to minimizing the energy/time consumption. For the multirobot case, first the DARP (Divide Areas based on Robots Initial Positions) algorithm is utilized to divide the area into separate equal subareas, so much so that it transforms the mCPP problem into several CPP problems, degrading the computation complexity. During the second phase, spanning tree in each subarea is constructed by the aforementioned algorithm. In the last phase, the specific end nodes are exchanged among subareas to achieve ideal-shaped spanning trees, which can also decrease the number of turns in coverage path. And the complete algorithms are proven to be approximately polynomial algorithms. Finally, the simulation confirms the complete algorithms' advantages: complete coverage, nonbacktracks, minimum length, zero preparation time, and the least number of turns.

## 1. Introduction

From the last century, intelligent robots have gradually been penetrating through various industries [1]. Nowadays, massive tasks need collaborations among multiple robots, which brings a plenty of robotic challenges that one of the fundamental problems is path planning for multiple robots. Usually, path planning is to determine an optimal path among "points" (e.g., start point to target points), while avoiding obstacles or no-fly zones. However, when our interest is the whole points of the given area or the reconnaissance points are uncertain, the coverage path planning (CPP), which guarantees robots pass over every point in a given area, should be carried out. In the literature, CPP problem is directly related to a mass of applications, such as search and rescue operations [2], inspecting industrial plants [3], monitoring hazardous environments [4], etc.

From the aforementioned applications, the CPP problem is associated with robots or their sensors and the area of interest. Therefore, one of the most common techniques is area decomposition technique. Our adopted approach is

termed as approximate cellular decomposition, which can separate the area of interest into identical cells (according to the scope of sensors), so that the robots can easily cover every point. Apparently, the union of all cells can approximate any arbitrary shaped target area.

In the literature, the single robot CPP problem has received a lot of attention from the decade. To solve this problem, the primary constraint is to completely cover the area of interest. And the next consideration is the high efficiency. During the varying techniques (see Section 2), the spanning tree coverage (STC) algorithm [5], constructing a spanning tree (ST) for all available cells, can guarantee completely covering the area of interest, while generating nonbacktracking path along the ST. Thus, the STC algorithm is one of the dominant approaches for this problem. Our approach utilizes the idea of STC algorithm and inherits its advantages. Simultaneously, because robots' turns will increase more energy/time consumption, we propose a methodology to find the best ST to reduce the number of turns. For more details, see Section 4 of the paper.

With the development of technology, multiple robots can collaborate to complete the CPP problem, which is referred to as mCPP problem. To determine the optimal path for mCPP problem, the objectives, including completeness, nonbacktracking path, initial position, minimum coverage path, and energy/time consumption, must be taken into consideration. Unfortunately, it is difficult to adequately address the mCPP problem. Particularly the minimal covering time is NP-hard [6]. And the proposed algorithms (see Section 2), which aim to overcome the NP nature of the problem, are mostly focusing only on one of the objectives. Through protracted and unremitting efforts, the authors in [7] make a great progress to propose the DARP (Divide Areas based on Robots Initial Positions) algorithm, which is suitable for the mCPP problem and will guarantee the completeness, nonbacktracking, initial position, and minimum coverage path. To reduce the energy/time consumption, the most direct and simplest method is to obtain “idea-shaped” subareas and decrease the number of turns, where extension investigation only based on the DARP algorithm is bright.

In the present paper, we attempt to solve the mCPP problem without overlooking any of the aforementioned objectives. Our contributions in this paper are threefold. Firstly, the available cells are divided into distinct classes based on the number, capabilities, and initial locations of robots, by improving the DARP algorithm. Secondly, the idea of ST is utilized to generate complete, nonbacktracking, and minimum coverage path. And then, improved ant colony optimization (ACO) algorithm is tried to find the best ST, to minimize the robots’ turns and energy/time consumption. Finally, the target is to further decrease the number of turns by exchanging the specific nodes among subareas.

The paper is organized as follows: Section 2 illustrates the related work on the mCPP problem. Some preparatory works and definitions for mCPP problem are completed in Section 3. In Section 4 improved ACO algorithm is proposed to construct the best ST in given area, taking into account the number of turns. In Section 5 “idea-shaped” subareas are accomplished, with a comprehensive discussion on exchanging cells. And the simulation results and comparison studies are presented in Section 6. Finally, Section 7 concludes the paper together with an outlook on the future work.

## 2. Related Work

*2.1. mCPP Problem in Given Area.* From the previous investigations, mCPP problem has been addressed extensively. The authors in [8] reviewed and evaluated several different methods, such as cellular methods, grids-based methods, and graphs coverage, for known or unknown areas. And this subsection presents the typical and dominant achievements on our problem.

Exact cellular decomposition methods are one representative of classical and popular methods. Typically, it first decomposes the known area into subregions. Next, the robot sweeps the subregions using simple motions. From the literature, trapezoidal decomposition can produce coverage paths only for polygonal spaces. But it creates so many

only convex cells, which increase more sweeping paths [9]. Boustrophedon decomposition, proposed by Choset [10] and Pignon [11], in an attempt to overcome the limitation of the aforementioned method, has shorter coverage paths in the same class of situations. Simultaneously, optimal sweep direction that is aimed at minimizing the number of turns holds less lanes, which has been proved in literature [12, 13]. Furthermore, Morse decomposition allows spiral patterns to simplify the flyable path and to minimize the time [14]. However, it is not flexible enough to handle rectilinear environments, and the path might be retracing. Overall, the exact cellular decomposition methods do not consider the robots’ initial position, and there might be some additional trajectory or backtracks among subregions.

Considering the young research field of mCPP problem, the spanning tree coverage (STC), firstly applied to multiple robots by Hazon, N. et al [15], is one of the dominant approaches. Their algorithm called MSTC can guarantee visiting all cell of interests only once. Unfortunately, the path of each robot is associated with its initial position, and the path of one may cover the whole given area in the worst case. Later, the same authors, trying to alleviate the shortcoming, improved the MSTC to OPT-MSC [16], by restraining the path distance between adjacent robots. However, there is no guarantee for the initial position. In literature [17], an alternative ST method, restraining the number of cells in known area and providing an upper bound on algorithm, was proposed to control each robot’s maximum path length. It performs better than MSTC and OPT-MSTC, though the nonbacktracking constraint may be broken. Worse still, all these methods disregard the number of turns or simple motion, which may result in more energy/time consumption.

*2.2. Area Decomposition.* In [18, 19] a convex polygon decomposition problem is presented. The anchored area is divided into more convex polygon pieces by sweep-line approach. As a result, each subarea contains the robots’ initial positions on its boundary and each of a specified area related to the robots’ capabilities. However, the approach performs well only on unrealistic conditions, such as convex polygon without obstacles and initial position on the boundary.

Concerning the concave polygon, the algorithm described in [13] first divides the concave polygon into separated convex polygons by a principle that the concave decomposition line is in parallel to one edge of the polygon. Then, subregions, who are entirely adjacent or whose widths have the same direction, will be combined to avoid unnecessary back and forth motion. Unfortunately, the technique does not consider the obstacles and initial position again. And there might be many repeated or additional paths when robots traverse to the next subregion.

Many of the state-of-the-art approaches, which seem suitable for the area division problem, have relied on Lloyd’s algorithm [20], Voronoi partitioning [21], K-means [22], some heuristic algorithm [23], and alternate-offer protocol [24]. Although these approaches have their special characteristics, the common ground of direct applying aforementioned approaches to mCPP problem is the suboptimal results or additional paths.

The authors in [7] make a great contribution to fully exploit the robots capabilities based on the advantage features, using a grid-based algorithm referred to as DARP (Divide Areas based on Robots Initial Positions). The proposed algorithm, which performs an area subdivision related to robots and defines paths in distributed manner, can guarantee the completeness (it covers all area), initial position (each subarea includes the robots' initial position, and robots can start at once), nonbacktracking path (the robots visit each cell only once), and minimum coverage path. Actually, the generated path does not consider the number of robots' turns, which increases the difficulty of the robots' implementation and energy/time consumption.

According to the aforementioned analysis, there is room for contributions concerning decreasing the number of turns and the time consumption. To meet these requirements, this work improves the DARP algorithm by exchanging some cells among adjacent subareas. Thus, it simplifies the path and reduces the number of turns so as to inherit the merit of DARP and improve the implementation efficiency of the robots.

### 3. Basic Definitions and Assumptions

Concerning the coverage path planning, the primary task is to organize the area of interest so as to choose suitable approaches which are related to the capabilities of robots.

Usually, the camera or sensor footprint of a robot is usually a trapezium. For simplification, the speed and altitude relative to ground level of the robot are considered to be fixed, and the projection of robots' location is at the center of the footprint which is assumed to be a square about  $l$  on a side.

For ease of understanding, the whole area of interest ( $A$ ) including obstacles ( $O$ ) is discretized into equal cells according to  $l$ . Furthermore, the area of interest to be covered is assumed a rectangle in the  $(x, y)$ -coordinates. Then the set of cells to be covered ( $U$ ) is represented as

$$A = \{(x, y), x \in [1, rows], y \in [1, cols]\} \quad (1)$$

$$O = \{(x, y) \in A, (x, y) \text{ is occupied}\} \quad (2)$$

$$U = A \setminus O \quad (3)$$

where  $rows, cols$  are the number of cells of rows and columns, and the number of all cells is  $n_a = rows \times cols$ . Apparently, the number of cells to be covered is  $n = n_a - n_0$ , where  $n_0$  represents the number of cells of obstacles.

*Definition 1.* Distance between two cells  $(x_i, y_i)$  and  $(x_j, y_j)$

$$D_{ij} = \|x_i - x_j\| + \|y_i - y_j\| \quad (4)$$

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

If  $D_{ij} \leq l$ , the two cells are adjacent; thus the robot can travel from one to the other at each timestamp.

If  $D_{ij} = d_{ij}$ , the two cells are in a line; otherwise there must be at least one turn in the path from one cell to the other.

*Definition 2.* A valid path is considered every sequence of cells

$$P = ((x_1, y_1), \dots, (x_m, y_m)) \quad (5)$$

where  $(x_1, y_1) \in U$ ,  $(x_i, y_i)$ , and  $(x_{i+1}, y_{i+1})$  are adjacent.

Moreover, the path would be closed, if  $(x_m, y_m)$  and  $(x_1, y_1)$  are adjacent.

### 4. Single Robot Coverage Path Planning

To solve the mCPP problem, we can start to cover the area of interest with only one robot. According to the aforementioned analysis, the single robot coverage path planning problem can be transformed to calculate the minimum length of complete coverage path  $P$  with minimum number of turns.

$$\begin{aligned} \min \quad & \|P\| \\ \text{s.t.} \quad & P \supseteq U \end{aligned} \quad (6)$$

where  $\|P\|$  denotes the length of path  $P$ .

*4.1. STC Algorithm.* The STC algorithm imposes a  $2l$ -size grid approximation on the given area of interest, where  $l$  is the sensor size. For the resulting grid, any effective procedure can be used to construct a ST, starting from any initial position. Then, a complete coverage closed path that circumnavigates the ST edges is generated. Figures 1(a)–1(d) illustrate the basic steps to generate a coverage closed path.

Considering the ST for the work-area, the consequences vary from methods or initial positions; thus the complete coverage closed path may be different. Figures 1(d) and 1(f) present two different STs for the same work-area. Apparently, the path in Figure 1(f) has more turns, which requires more energy/time consumption for robots. Also, it does not help the simple motions to alter the direction too frequently. To reduce the number of turns, a better method for ST should be selected.

Most often, minimum-spanning-tree algorithm is referred to together with Kruskal algorithm, Prim algorithm, depth-first search algorithm, and breadth-first search algorithm. In the present formulation, the nodes are at the center of the gathering 4 cells, where the distances between adjacent nodes are the same. And these algorithms all can generate a complete coverage ST for a given area but its shape is completely random and makes a big difference on robot's initial position. Concerning the number of turns, the path resulting from depth-first search algorithm is more likely to have the least compared to others, but still having no optimality guarantee. Fortunately, an improved ACO algorithm is proposed to construct a complete coverage ST with minimum number of turns, so much so that we can benefit from the advantages of ACO algorithm.

*4.2. Improved ACO Algorithm for ST.* Ant colony optimization (ACO) algorithm, firstly proposed by Macro Dorigo [25], is inspired by real life behavior of a colony of ants seeking an optimal path between their colony and food. In their search,

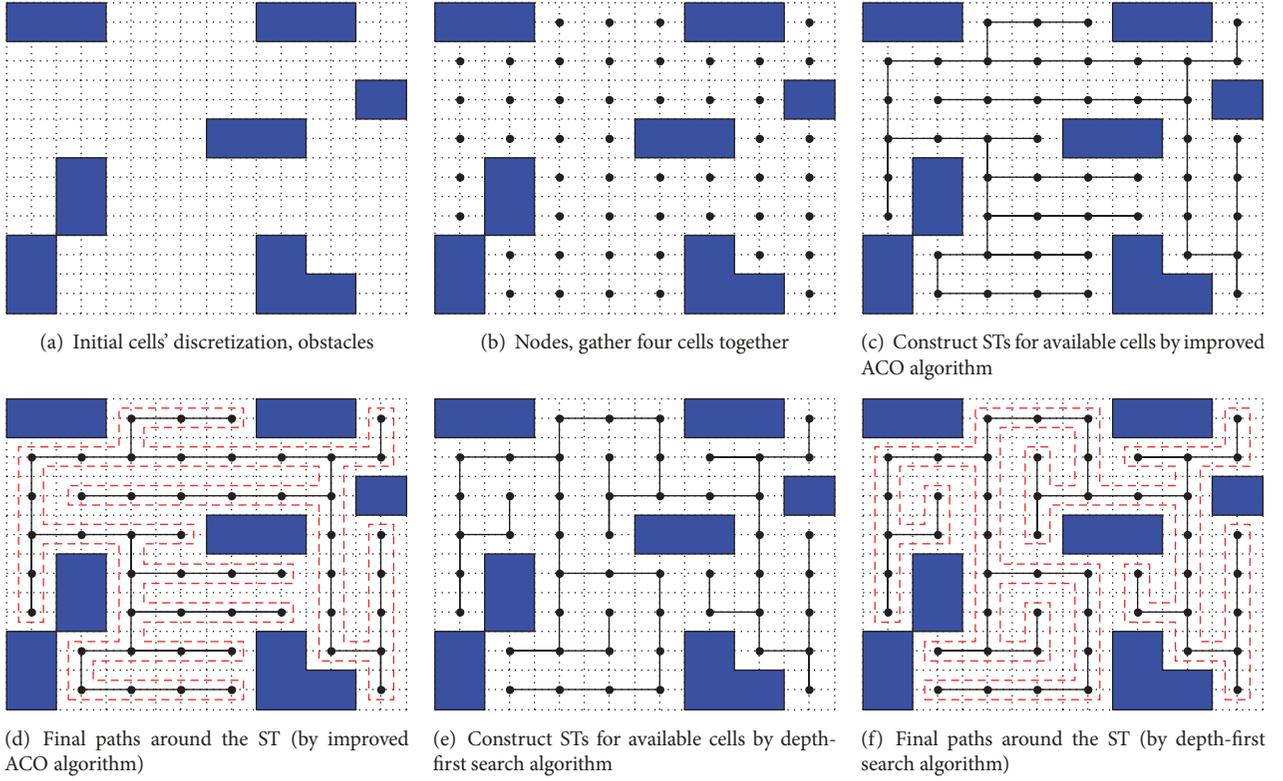


FIGURE 1: Comparison of two ST algorithms.

ants roam randomly and once they find food, they mark the trails by laying pheromone, the amount of which influences other ants to retain random travelling or follow the trails. The more pheromones a path has, the more possibility other ants follow the trail and consequently lay more pheromones. But over time the trails' attraction decreases with the evaporation of pheromone. As a result, the shorter trail that costs less time has more pheromones so that finally it is reinforced. On the contrary, pheromone evaporation and random travelling may contribute to avoiding locally optimal solution. In that case, the optimal solution can be found by exploring only a small part of the solution space.

To accomplish ST with minimum number of turns, the improved ACO algorithm is proposed, whose detailed procedures are as follows.

(i) *Initially.* Initialization of parameter as follows:

$m$ : quantity of the ants

$k$ : index of the ants

$itM$ : maximum iteration

$t$ : index of iteration number

$\alpha$ : influence of the pheromone value

$\beta$ : influence of heuristic information

$\tau_{ij}$ : pheromone information between nodes  $N_i$  and  $N_j$

$\eta_{ij}$ : Heuristic information correlated with the distance between nodes  $N_i$  and  $N_j$

$\rho$ : Evaporation rate of pheromone

All the ants are randomly placed on the nodes

(ii) *Transition Rules.* When the current node is  $N_i$  and the number of iterations is  $t$ , the ant  $k$  ( $k = 1, 2, \dots, m$ ) selects the next node  $N_j$  following the pseudorandom rules as the following formula.

$$N_j = \begin{cases} \arg \max \{ \tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t) \}, & q < q_0 \\ N_j, & q \geq q_0 \end{cases} \quad (7)$$

where  $q_0$  is specified in  $[0, 1]$ ,  $q$  is a random number in  $[0, 1]$ , and  $N_j$  is randomly selected according to the probability:

$$p_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha(t) \cdot \eta_{ij}^\beta(t) \cdot \gamma(t)}{\sum_{l \in adj_{ki}} \tau_{il}^\alpha(t) \cdot \eta_{il}^\beta(t) \cdot \gamma(t)}, & j \in adj_{ki} \\ 0, & j \notin adj_{ki} \end{cases} \quad (8)$$

where  $\eta_{ij} = 1/D_{ij}$ .  $\gamma(t)$  is a parameter to tune the importance of direction; if nodes  $N_{i-1}$ ,  $N_i$ , and  $N_j$  are in a line, it defines  $\gamma(t) = 1$ ; else  $\gamma(t) = 0.5$ . Thus the ants can be firstly guided to the node that is in line with nodes  $N_{i-1}$  and  $N_j$ .  $adj_{ki}$  represents a set of unvisited adjacent nodes when the ant  $k$  is at node  $N_i$ ; if  $adj_{ki} \neq \emptyset$ , and nodes  $N_{i-1}$ ,  $N_i$ , and  $N_j$  are not in a line, then update the number of turns  $r_t$  and the length of backtracks  $r_{lb}$  by  $r_t = r_t + 1$ ,  $r_{lb} = 0$ . If  $adj_{ki} = \emptyset$ , it means there is no unvisited adjacent node for

node  $N_i$ . To improve efficiency, the distance between nodes  $N_i$  and  $N_j$  can be increased temporarily avoiding unwise transition. Simultaneously, update the length of backtracks  $r_{lb}$  and backtracks number  $r_b$  by  $r_{lb} = r_{lb} + 1$ ,  $r_b = r_b + 1$ ; then take node  $N_{i-2r_{lb}+1}$  as the current node, so as to keep finding next node until the next unvisited node; update backtracks number  $r_b$  by  $r_b = r_b + 1$ . Finally, cycle the above procedures until all nodes are visited.

(iii) *Pheromone Actualization*. When all ants have finished a circuit, the pheromone information can be updated according to the optimal path found in the current generation. And considering the evaporation, the pheromone can be updated by the following formula:

$$\begin{aligned} \tau_{ij}(t+1) &= (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \quad 0 < \rho < 1 \\ \Delta\tau_{ij} &= \sum_{k=1}^m \Delta\tau_{ij}^k \\ \Delta\tau_{ij}^k &= \begin{cases} \frac{1}{R_{ak}}, & i, j \in P_o \\ 0, & \text{else} \end{cases} \end{aligned} \quad (9)$$

where  $P_o$  describes a set of nodes included in the optimal path and  $R_{ak}$  represents the number of turns of the optimal path the ant  $k$  ( $k = 1, 2, \dots, m$ ) passed, and it can be calculated from the number of turns  $r_t$  and global backtracks  $r_b$ .

(iv) *Stopping Rules*. The arithmetic stops when there is no improvement on the solution or the set value of iteration is reached, and then the optimal path will be output.

4.3. *Performance Discussion*. This subsection presents the basic steps to generate coverage closed path. Also, a comparison with depth-first search algorithm is made to confirm the performance of improved ACO algorithm.

First, a given terrain including obstacles is discretized as Figure 1(a). Second, every four cells are grouped into a large square-shaped cell, and the center of the large cell is defined as node. Third, the ST is constructed by improved ACO algorithm (the parameters are set as follows:  $m = 30$ ,  $itM = 150$ ,  $\alpha = 1$ ,  $\beta = 1$ ,  $\rho = 0.15$ , and  $\tau_{ij}(0) = 1$ ). Here, depth-first search algorithm is also adopted to be a comparison. Last, a complete coverage closed path that circumnavigates the ST edges is generated.

The coverage path in Figure 1(d) contains 52 90-degree turns (one 180-degree turn can be regarded as two 90-degree turns), which is less than 64 turns of the coverage path in Figure 1(f). The reason is that the ST resulting from improved ACO algorithm is always aimed at minimizing the number of turns, but the ST resulting from depth-first search algorithm tends to be random and restricts the robot's initial position. Apparently, the path resulting from improved ACO algorithm is more likely to have fewer turns so much so that there is less energy/time consumption. Thus, the improved ACO algorithm is more suitable for constructing ST taking into account the number of turns.

## 5. Multirobot Coverage Path Planning

Similar to the single robot CPP problem, the mCPP problem can be defined to calculate the minimum length of paths  $P_i$  so as

$$\begin{aligned} \min \quad & \max_{1 \leq i \leq n_r} \|P_i\| \\ \text{s.t.} \quad & P_1 \cup P_2 \cup \dots \cup P_{n_r} \supseteq U \end{aligned} \quad (10)$$

where  $\|P_i\|$  denotes the length of path  $P_i$ .

5.1. *DARP Algorithm*. The DARP (Divide Areas based on Robots Initial Positions) algorithm, proposed in literature [7], has two phases including dividing the available cells into distinct subareas as many as the number of robots and finding optimal path for each subarea. The latter has been realized from the approaches for single robot CPP problem. In other words, only the problem of dividing available cells, without any concern about the optimal paths, has to be addressed. Thus, the optimal solutions for mCPP problem can be defined as

$$\begin{aligned} \min \quad & \max_{1 \leq i \leq n_r} \|S_i\| \\ \text{s.t.} \quad & S_1 \cup S_2 \cup \dots \cup S_{n_r} = U \\ & S_i \cap S_j = \emptyset, \quad \forall i, j \in 1, \dots, n_r, \quad i \neq j \\ & \|S_1\| \approx \|S_2\| \approx \dots \approx \|S_{n_r}\| \\ & \chi_i \in S_i \\ & S_i \text{ is connected} \quad \forall i \in 1, \dots, n_r \end{aligned} \quad (11)$$

where  $S_i$  denotes the set of subareas but not the strict paths,  $n_r$  denotes the number of robots, and  $\chi_i$  represents the robots' initial position. And these constraints ensure the complete coverage, nonbacktracking path, absolutely fair division, subareas including the related robots' initial position, and continuous subareas, respectively.

To achieve the areas subdivision, we must complete the following twofold.

(i) *Equally Divide the Space*. Each robot's subarea  $S_i$  can be computed by the assignment matrix  $A$ ; they are constructed according to

$$\begin{aligned} S_i &= \{(x, y) \in U : A(x, y) = i\}, \quad \forall i \in 1, \dots, n_r \\ A_{x,y} &= \arg \min_{1 \leq i \leq n_r} E_{i|x,y}, \quad \forall (x, y) \in U \\ E_i &= \alpha_i E_i \\ \alpha_i &= \alpha_i + c(k_i - f) \end{aligned} \quad (12)$$

where  $E_i$  represents evaluation matrix, which expresses the distance between the cells of  $U$  and robots' initial position and  $\alpha_i$  is a scalar correction factor,  $c$  denotes a positive tunable parameter,  $k_i$  is the cardinality of the  $S_i$  set, and  $f$  denotes the global "fair share," which is computed by those available cells divided by the number of robots.

(ii) *Build Spatial Connected Areas.* From the aforementioned analysis, the division cannot guarantee the continuity of the subarea. To deal with such situations, the evaluation matrices are finally updated as

$$\begin{aligned} E_i &= C_i \otimes (\alpha_i E_i) \\ C_{i|x,y} &= \min(\| [x, y] - r \|) - \min(\| [x, y] - q \|) \\ &\quad \forall r \in R_i, q \in Q_i \end{aligned} \quad (13)$$

where  $\otimes$  denotes the element-wise multiplication,  $R_i$  denotes the set of cells connected with the initial position,  $Q_i$  denotes the set of cells assigned to the robot but unconnected with the initial position, and  $C_i$  is a parameter to construct gradually a closed-shaped subarea of each robot. If all subareas of each robot are continuity,  $C_i$  is set to all-one-matrix.

In a nutshell, the division comes out by modifying the related evaluation matrix iteratively.

5.2. *Exchange Nodes.* Although the DARP algorithm, which can equally divide the area of interest and guarantee the continuity of subarea, aims to provide the optimal cell's assignment, the shape of subarea is random without taking simple motion or fewer turns into consideration. As shown in Figure 2, each area has the same number of cells but different shapes, resulting in different coverage paths along with different number of turns.

Dealing with such situations, a method, exchanging nodes among different subareas, is introduced to minimize the number of turns in each subarea. By adopting this method, the number of turns in each subarea should meet the equations as

$$\begin{aligned} T_i &\leq T_{i0}, \quad \forall i \in 1, \dots, n_r \\ \min \sum_{i=1}^{n_r} T_i & \end{aligned} \quad (14)$$

where  $T_i$  denotes the number of turns in  $i$ th subarea and  $T_{i0}$  represents the number of turns before exchanging nodes.

By applying this method, we must achieve the following twofold: confirming the exchangeable nodes and the method to exchange them. A fine-grained analysis is as follows.

(i) *Exchangeable Nodes.* On the basis of DARP algorithm, there are subareas. To obtain the exchangeable nodes, first and foremost, a ST should be constructed for each subarea, where the number of turns in each node can be confirmed.

*Definition 3.* In the ST, one is called the end node, which is only connected with one adjacent node.

Considering the continuity of the ST in subarea, only the end node can be exchanged (given to the adjacent subarea). As shown in Figure 3. The node  $N_j$  is the end node, and it is only connected with node  $N_i$ . Apparently, the end nodes can be divided into three types:

(a) I-shaped end node: there are two turns, and when this node is discarded, there are still two turns.

(b) L-shaped end node: there are four turns, but when this node is discarded, they decrease to two turns.

(c) T-shaped end node: there are four turns, but when this node is discarded, they decrease to none.

(ii) *Exchange Nodes.* Having the above concept in mind, it needs to obey the following rules so that the number of turns will be decreased when exchange nodes.

(a) Each exchange means that one subarea discards an end node and another subarea receives this node.

(b) When discarding an end node, T-shaped end node is top-priority, followed by L-shaped end node. And this discarded node received by an adjacent subarea will connect the new ST and form another shaped end node, where I-shaped end node is the best, L-shaped end node is next, and T-shaped end node is the worst choice to decrease the number of turns. Then, this adjacent subarea will discard an end node to another subarea, and so on. After a circuit, the number of nodes belonging to related subarea is still about  $(n/n_r)$ , and if the number of turns cannot meet (14), then discard the current exchange and restart a new circuit.

(c) All exchangeable nodes in one subarea experience an exchange and then turn to another subarea.

(d) The exchange stops when there is no longer decrease in the total number of turns in the given area of interest or reaching the max iteration times, and then the optimal path will be output.

5.3. *Overview of the Complete Algorithms.* Concluding the above two subsections, the complete algorithms for mCPP problem are summarized. The algorithms in the present paper include three phases: During the first phase, the area of interest is equally divided into  $n_r$  separate subareas by DARP algorithm, so much so that the mCPP problem can be downgraded to  $n_r$  single CPP problems. Next, STs are constructed for subareas, respectively, to confirm the number of turns and exchangeable nodes, utilizing improving ACO algorithm. After the above two phases, the complete coverage, nonbacktracking path, absolutely fair division, subareas including the related robots' initial position, and continuous subareas can be easily satisfied. In the last phase, the target is to decrease the number of turns by exchanging the specific nodes among subareas.

A flowchart of the proposed algorithm is presented in Figure 4.

5.4. *Memory and Computational Complexity Analysis.* As the previous subsection, the complete algorithms include three phases with separate calculations, so that the computational and memory complexity can be analyzed in three parts.

In the first phase, DARP algorithm's memory complexity is  $o(n_r \times n)$ . The maximum iterations (in the worst case scenario) of the algorithm depend on the number of robots, random initial deployments, and the grid size, resulting in being practically infeasible to compute exhaustively. A series of simulations are adopted to approximate the computational complexity, the curve of which is proven strictly bounded

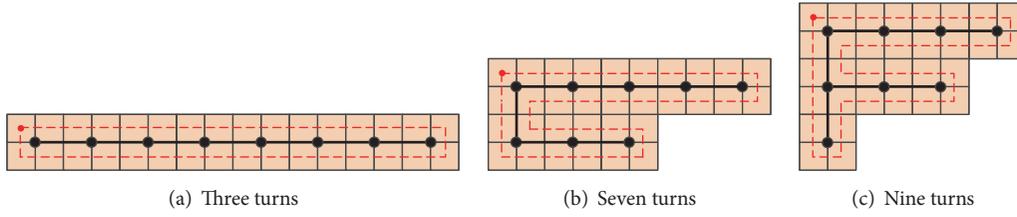


FIGURE 2: All are 32 cells but of different shapes.

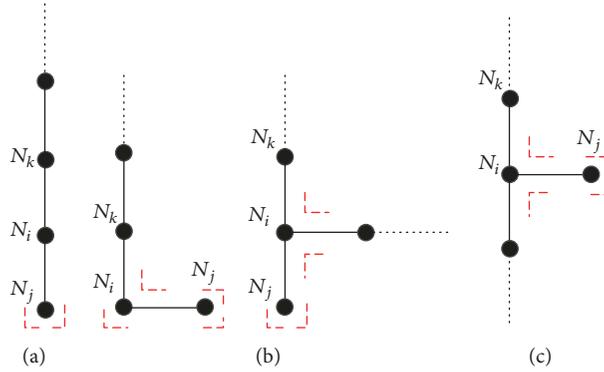


FIGURE 3: Three types of end node.

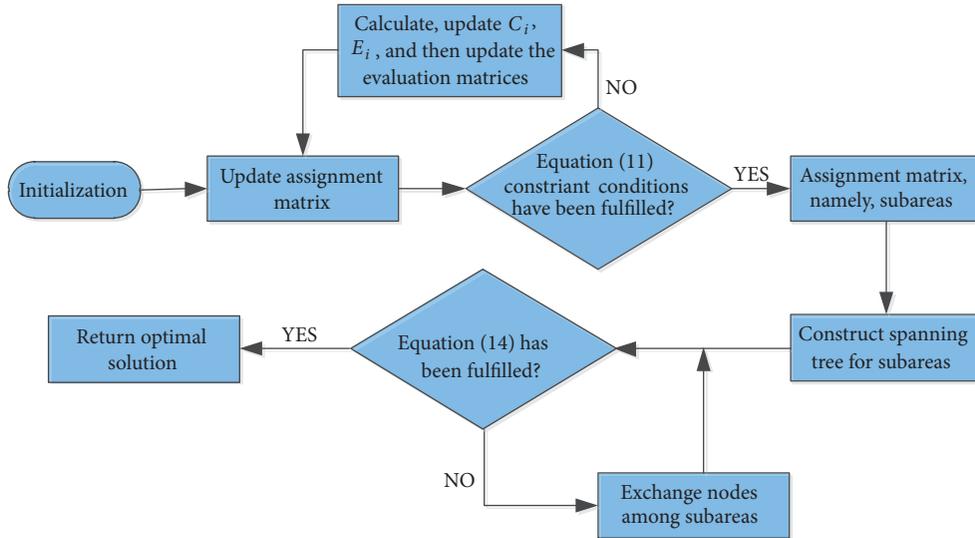


FIGURE 4: Flowchart of the complete algorithms.

under the  $n_r^3 \times n^2$  curve until a practical interesting input ( $n_r \times n$ ).

In the second phase, improved ACO algorithm constructs STs for all subareas, the number of which is equal to the number of robots  $n_r$  and the grid sizes in which are about  $(n/n_r)$ . The memory and computational complexity are  $o((n/n_r)^2 \times n_r)$  and  $o((n/n_r)^4 \times n_r)$ , i.e.,  $o(n^2)$  and  $o(n^4)$ , respectively.

In the last phase, the memory complexity is obviously linear to the size of input ( $n_r \times n$ ), i.e.,  $o(n_r \times n)$ . The computational complexity of exchanging nodes depends on the number of the subareas  $n_r$  and exchange times among

them. There are at most  $n/n_r$  exchangeable nodes in a subarea; thus it needs  $(\alpha \times n/n_r)$  circuits (related to the optimal solution) to complete exchanging these nodes, where  $\alpha$  is a constant number. And in the worst case scenario, each cycle needs to exchange nodes  $n_r$  times. As a result, the computational complexity is defined as

$$\begin{aligned}
 o(T_3) &= n_r \times \left( \alpha \times \frac{n}{n_r} \right) \times n_r = \alpha \times n_r \times n \\
 &= o(n_r \times n)
 \end{aligned}
 \tag{15}$$

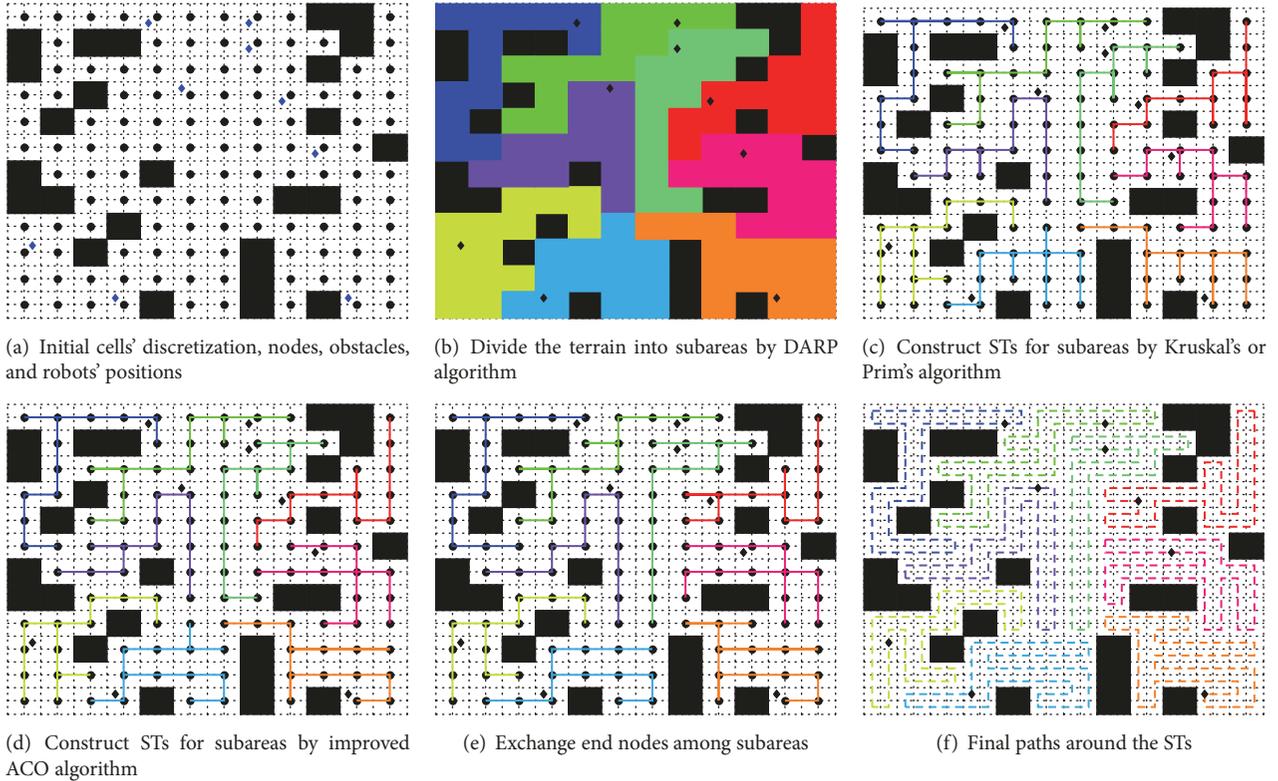


FIGURE 5: Optimal coverage paths for terrain with  $24 \times 24$  grid size, 9 robots, and 100 obstacles.

Since the number of robots  $n_r$  and the grid size  $n$  meet  $1 \leq n_r < n$ , the memory complexity of the complete algorithm can be written as

$$o(M) = o(n_r \times n) + o(n^2) + o(n_r \times n) = o(n^2) \quad (16)$$

And the computational complexity can be written as

$$\begin{aligned} o(T) &= o(n_r^3 \times n^2) + o(n^4) + o(n_r \times n) \\ &= o(n_r \times n^4) \end{aligned} \quad (17)$$

Concluding this section, the complete algorithms are approximately polynomial algorithms until practical input, but may lose their polynomial behavior with numerous inputs ( $n_r \times n$ ).

## 6. Simulation Results

This section presents a simulation of the proposed algorithms. The simulation setup adopted is the same as in [7] for comparable results. More precisely,

- (i) the size of the area of interest is [rows, cols] =  $24 \times 24$
- (ii) the obstacles arrangement follows a random uniform distribution
- (iii) there are 9 robots in this area, and their initial positions are as shown in Figure 5(a)

Figure 5 represents a comparison between the proposed algorithms with DARP+STC algorithms. Figure 5(a) illustrates the initializations. Figure 5(b) represents dividing the terrain into separate subareas (the number of gathered cells is [13 13 13 13 13 13 14 13 14]) by DARP algorithm, each of which contains the related robots' initial position. Then, in Figure 5(c), STs are constructed by algorithm in [7] (Kruskal's or Prim's algorithm), which results in a random tree, not concerning the number of turns. And the total number of turns in paths is 174 ([16 18 18 18 20 22 20 22 20]). In Figure 5(d), taking the number of turns into consideration, improved ACO algorithm is adopted to construct the STs. And the total number of turns in paths is 164 ([16 18 18 16 18 18 20 20 20]), which is less than those in Figure 5(c). To further decrease the number of turns, some specific end nodes are exchanged among subareas, as shown in Figure 5(e). The total number of turns in paths is 148 ([14 18 12 14 18 18 18 16 20]), where the number of turns in each subarea and the whole terrain is obviously the least. Finally, the optimal paths are around the STs in Figure 5(f).

In literature [7], the DARP + STC algorithms are compared with MFC and optimized MSTC algorithms (the state-of-the-art methods) in cover time (in terms of path length) for all robots, which proves that the DARP + STC algorithms have optimality guarantee. In the present paper, the proposed algorithms do not increase the cells of any subarea (namely, they do not increase the path length); at the same time they decrease the number of turns. Apparently, when covering

a given cell, the energy/time consumption covering in a line is less than that covering with turn, so much so that the proposed algorithms have optimality guarantee when concerning the number of turns.

## 7. Conclusions

The proposed algorithms are aimed at finding optimal solution for multirobot coverage path planning problem. During the preliminary analysis, the basic definitions and preparations are made to achieve the optimal performance easily. First, of the proposed algorithms, the area of interest is equally divided into separate subareas utilizing the DARP algorithm, which takes the obstacles and initial positions into consideration. Then, STs are constructed for subareas by improved ACO algorithm, which means avoiding paths with a mass of turns. On the basis of STs, some end nodes are exchanged to modify the shape of STs as “ideal shape,” so as to further decrease the number of turns. With these orchestrated STs, the coverage paths can be calculated. It is worth to point out that the robots can start from their initial positions, and not only are the overall paths complete, nonbacktracking, minimum coverage paths but also there are fewer turns in paths. And these above features never appeared in other methods at the same time.

Regarding the proposed method, it tries to find an optimal solution, but the numbers of turns in paths are still restricted to the basic shapes of the subareas. Thus, one of the future works could be to explore a succinct method which straightforwardly divides the terrain into ideal-shaped subareas, or some other methods can achieve all the aforementioned features.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Disclosure

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

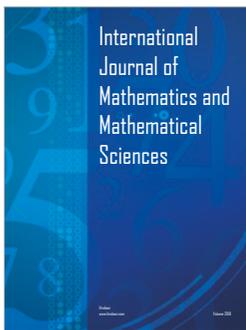
## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] K. Goldberg, “Robotics: Countering singularity sensationalism,” *Nature*, vol. 526, no. 7573, pp. 320–321, 2015.
- [2] S. Waharte and N. Trigoni, “Supporting search and rescue operations with UAVs,” in *Proceedings of the International Conference on Emerging Security Technologies (EST’10)*, pp. 142–147, Canterbury, UK, September 2010.
- [3] J. Nikolic, M. Burri, J. Rehder, S. Leutenegger, C. Huerzeler, and R. Siegwart, “A UAV system for inspection of industrial facilities,” in *Proceedings of the 2013 IEEE Aerospace Conference, AERO 2013*, Big Sky, Mont, USA, March 2013.
- [4] D. W. Casbeer, D. B. Kingston, R. W. Beard, and T. W. McLain, “Cooperative forest fire surveillance using a team of small unmanned air vehicles,” *International Journal of Systems Science*, vol. 37, no. 6, pp. 351–360, 2006.
- [5] Y. Gabriely and E. Rimon, “Spanning-tree based coverage of continuous areas by a mobile robot,” *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1-4, pp. 77–98, 2001.
- [6] X. Zheng, S. Jain, S. Koenig, and D. Kempe, “Multi-robot forest coverage,” in *Proceedings of the IEEE IRS/RSJ International Conference on Intelligent Robots and Systems, IROS 2005*, pp. 2318–2323, Edmonton, Canada, August 2005.
- [7] A. C. Kapoutsis, S. A. Chatzichristofis, and E. B. Kosmatopoulos, “DARP: Divide Areas Algorithm for Optimal Multi-Robot Coverage Path Planning,” *Journal of Intelligent & Robotic Systems*, vol. 86, no. 3-4, pp. 663–680, 2017.
- [8] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [9] H. Choset, K. Lynch, S. Hutchinson et al., *Principles of Robot Motion: Theory Algorithms and Implementation*, The MIT Press, 2005.
- [10] H. Choset, E. Acar, A. A. Rizzi, and J. Luntz, “Exact cellular decompositions in terms of critical points of Morse functions,” in *Proceedings of the ICRA 2000: IEEE International Conference on Robotics and Automation*, pp. 2270–2277, San Francisco, Calif, USA, April 2000.
- [11] H. Choset and P. Pignon, “Coverage path planning: the boustrophedon cellular decomposition,” in *Proceedings of International Conference on Field and Service Robotics*, Canberra, Australia, 1997.
- [12] J. F. Araujo, P. B. Sujit, and J. B. Sousa, “Multiple UAV area decomposition and coverage,” in *Proceedings of the 2013 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2013 - 2013 IEEE Symposium Series on Computational Intelligence, SSCI 2013*, pp. 30–37, Singapore, Singapore, April 2013.
- [13] Y. Li, H. Chen, M. Joo Er, and X. Wang, “Coverage path planning for UAVs based on enhanced exact cellular decomposition method,” *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011.
- [14] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, “Morse decompositions for coverage tasks,” *International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002.
- [15] N. Hazon and G. A. Kaminka, “Redundancy, efficiency and robustness in multi-robot coverage,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 735–741, Barcelona, Spain, April 2005.
- [16] N. Agmon, N. Hazon, and G. A. Kaminka, “Constructing spanning trees for efficient multi-robot coverage,” in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006*, pp. 1698–1703, Orlando, Fla, USA, May 2006.
- [17] Y. Elmaliach, N. Agmon, and G. A. Kaminka, “Multi-robot area patrol under frequency constraints,” *Annals of Mathematics and Artificial Intelligence*, vol. 57, no. 3-4, pp. 293–320, 2009.
- [18] S. Hert and V. Lumelsky, “Polygon area decomposition for multiple-robot workspace division,” *International Journal of Computational Geometry and Applications*, vol. 8, no. 4, pp. 437–466, 1998.
- [19] I. Maza and A. Ollero, “Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms,” in *Distributed Autonomous Robotic Systems*, vol. 6, pp. 221–230, Springer, Tokyo, Japan, 2007.

- [20] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.
- [21] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," *ACM Computing Surveys*, vol. 23, no. 3, pp. 345–405, 1991.
- [22] D. Puig, M. A. Garcia, and L. Wu, "A new global optimization strategy for coordinated multi-robot exploration: development and comparative evaluation," *Robotics and Autonomous Systems*, vol. 59, no. 9, pp. 635–653, 2011.
- [23] H. Bast and S. Hert, "The area partitioning problem," in *Proceedings of the 12th Canadian Conference on Computational Geometry Fredericton*, New Brunswick, Canada, 2000.
- [24] A. Rubinstein, "Perfect equilibrium in a bargaining model," *Econometrica*, vol. 50, no. 1, pp. 97–109, 1982.
- [25] A. Coloni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in *Proceedings of the 1st European Conference on Artificial Life*, pp. 134–142, Paris, France, 1991.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

