# A Novel Graph-based Motion Planner of Multi-Mobile Robot Systems with Formation and Obstacle Constraints

Wenhang Liu, Jiawei Hu, Heng Zhang, Michael Yu Wang, *Fellow, IEEE/ASME,* and
Zhenhua Xiong, *Member, IEEE*

*Abstract*—Multi-mobile robot systems show great advantages over one single robot in many applications. However, the robots are required to form desired task-specified formations, making feasible motions decrease significantly. Thus, it is challenging to determine whether the robots can pass through an obstructed environment under formation constraints, especially in an obstacle-rich environment. Furthermore, is there an optimal path for the robots? To deal with the two problems, a novel graph-based motion planner is proposed in this paper. A mapping between workspace and configuration space of multi-mobile robot systems is first built, where valid configurations can be acquired to satisfy both formation constraints and collision avoidance. Then, an undirected graph is generated by verifying connectivity between valid configurations. The breadth-first search method is employed to answer the question of whether there is a feasible path on the graph. Finally, an optimal path will be planned on the updated graph, considering the cost of path length and formation preference. Simulation results show that the planner can be applied to get optimal motions of robots under formation constraints in obstacle-rich environments. Additionally, different constraints are considered.

*Index Terms*—Multi-mobile robot systems, motion planning, formation constraints, obstacle-rich environment.

## I. INTRODUCTION

IN recent years, multi-mobile robot systems (MMRS) have attracted increasing attention from robotics researchers. Compared with a single robot, the robots can cooperate with each other to achieve better system robustness and flexibility. Through collaboration among robots, complex tasks can be accomplished, such as search and rescue [1], [2], exploring [3], assembling [4] and transporting [5]. Moreover, MMRS can replace the bigger single robot in scenarios with a restricted and cramped environment, where there is not enough space. In many applications of MMRS mentioned above, the robots are usually required to form desired formations to achieve cooperation, such as the object will connect the robots to form a whole system during cooperative transporting. Accordingly,

the flexibility of each robot will be decreased since it has to satisfy formation constraints. In the meanwhile, the robots also need to avoid obstacles when working, which requires them to be as flexible as possible. Conflicts will arise if simultaneously considering formation constraints and obstacle avoidance when planning motions of robots, especially in a dense environment with multiple obstacles. Therefore, it is a challenging problem to determine whether MMRS can pass through an obstructed environment under task-specified formation constraints. Moreover, if it is possible, what is the optimal path for the robots to fulfill the task? Thus, we focus on multi-mobile robot motion planning problems under formation constraints in obstacle-rich environments in this paper. A novel graph-based motion planner is proposed. A mapping between the workspace and configuration space of MMRS is first built, where valid configurations are acquired according to formation constraints and obstacle avoidance. An undirected connected graph with boundary densification is then generated by verifying the connectivity between valid configurations. Vertices and edges on the graph represent valid configurations and the connectivity between configurations, respectively. Finally, the breadth-first search algorithm is adopted on the graph to quickly determine whether there is a feasible path, and the optimal path and motions of robots are computed considering path length and formation preference. Cooperative object transportation is used as a typical example. Simulation results in different obstacle-rich environments demonstrate the effectiveness and generality of the proposed planner. Besides, the planner can be extended if there are additional constraints brought by the tasks.

The main contributions of this paper are threefold. First, a novel mapping between workspace and configuration space of multi-mobile robot systems satisfying formation constraints is proposed. Theoretical proofs show that the problems can be solved on the undirected connected graph generated by mapping. Second, a boundary densification method is proposed on the connected graph to facilitate planning and improve the success rate for obstacle-rich environments. Third, the planner can quickly determine whether it is feasible for MMRS to pass an obstacle-rich environment under formation constraints. Then, the optimal motions of the robots can be planned.

The rest of this paper is organized as follows. In Section II, some related work is discussed. In Section III, we define the basic concepts of multi-robot motion planning problems. In Section IV, the proposed motion planner is explained in

detail. In Section V, the planner is employed for different formation constraints by a case study in object transportation. The simulations and results are given in Section VI. Section VII concludes the paper and outlines the future work.

## II. RELATED WORK

In many studies of multi-robot motion planning, formation constraints among robots are not the key point. For each robot, it regards other robots as dynamic obstacles and collaboration is not considered [6]–[8]. However, as tasks become more complex, the collaboration between multiple robots is inevitable, especially with formation constraints.

Generally, approaches in the previous works on motion planning of MMRS under formation constraints can be divided into two categories. The first category is individually planning each robot while satisfying formation constraints and avoiding obstacles. Under the combination of the two behaviors, the system can hold desired formations [9]. Based on the leader-follower and potential-based methods, desired formations with obstacle avoidance were achieved in [10]–[12]. By combining the virtual structure and obstacle avoidance methods, [13] introduced an approach for multiple robots to maintain formations. Similarly, the combination can be found in [14]. Besides, learning-based methods are also employed for the problems. Formations could be adjusted independently by each robot to pass through obstacle areas based on reinforcement learning [15], [16]. In [17], the task space, including formation constraints and obstacle avoidance, was decomposed into different convex regions through a global planner. Then, configurations were planned through a local planner. In the above research, the systems can all maintain desired formations tightly and move to the goal in environments without obstacles. However, when it comes to obstacle avoidance, some uncontrollable deformation of system formations will appear more or less, which is not acceptable for some cooperative tasks.

The second category considers the multi-robot system as a whole 2D area, which is planned to be strictly separated from obstacles. A system outlined rectangle approach was proposed in [18], regarding the multi-robot system as a virtual rectangle to avoid obstacles. Then, the problem was simplified to the motion planning of a single robot. The idea can also be found in [19], [20]. In [21] and [22], a constrained optimization method was presented for motion planning of multiple robots. The robots could always form the desired formations in the largest convex obstacle-free region computed in the neighborhood of the system. A region-based framework for multi-robot systems was introduced in [23]. Based on virtual structure, robots moved and formed formations inside a changeable circular region always separated from obstacles. In [24], formation control and change were achieved by caging behaviors. The whole system was planned by the rapidly-exploring random tree for obstacle avoidance. Sometimes, the multi-robot systems have to be considered as a whole due to special tasks, as in [25]–[28]. However, for the above research, it is over-constrained since robots can actually be separated. At least, the system could have crossed obstacles instead of merely bypassing obstacles as a whole. Therefore, the system loses part of its obstacle avoidance ability.

It is noted that current research did not explicitly consider the formation constraints. Thus, the deformation of systems during motion cannot be limited to a certain range of tasks. The conflict between forming formations and avoiding obstacles has not been fully handled. In some situations where rigid or less flexible connections among the robots are required, these methods are unsuitable. 2D region-based methods are safe for obstacle avoidance. However, it is only suitable for a few obstacles environment. Therefore, it is worth finding a method for multi-robot motion planning in obstacle-rich environments, at the same time keeping formation constraints.

In multi-robot motion planning studies, graph theory is a powerful tool [29]–[31]. In [7], based on a graph and spanning tree representation, a multi-phase approach was described. Through the relationship between multi-robot planning problems and multi-flow problems, integer linear programming models were proposed to compute optimal paths for multiple robots on connected graphs [32]. Their models were further extended in [33] in connection with different minimization objectives in path planning. The optimal solution or approximate optimal solution could be quickly calculated on the connected graphs. In [34], goal allocation and motion planning were achieved on roadmaps for non-holonomic robots.

In the above studies with graph theory, the robots are more likely to be planned individually, where the confliction between waypoints is the only connection among multiple robots [35]–[38]. However, formation constraints are not considered. Therefore, in order to solve the problem through graph theory, the mapping between configuration space and workspace needs to be studied [39]. Moreover, the configuration space of MMRS should be further investigated, especially in the case of integrated obstacle avoidance and formation constraints.

## III. PROBLEM FORMULATION

In this work, we consider multi-mobile robot motion planning under both obstacle-rich environments and formation constraints. The mobile robots are assumed to be holonomic, and they move on a 2D plane. The typical task is to transport an object from the initial position to goal position.

Some important notations are listed in Table I for convenience. In a workplace $\mathcal{W} \subset \mathbb{R}^3$, the position of the $i$th mobile robot is denoted as $r_i \in \mathbb{R}^2, i = 1, \ldots, n$, and $n$ is used to denote the number of mobile robots. The plane where robots are located is denoted by $\mathcal{P} \subset \mathbb{R}^2$. The center of multi-robot systems is denoted as $p = [x, y] := \sum_{i=1}^{n} \frac{r_i}{n} \in \mathbb{R}^2$, which is used to describe the absolute location of systems in the workplace. Formation $s \in \mathcal{S}$ formed by robots is defined as follows.

$$s := \left\{ d_{ij}^* \right\} \quad i, j = 1, \ldots, n \quad i \neq j$$
$$d_{ij}^* = \|r_i - r_j\|_2 \tag{1}$$

where $s$ is the desired formation shape of multi-robot systems, which consists of relative positions between robots. Since $p$ is not changed by the rotation of the same formation shape, multiple positions of robots may exist. As shown in Fig. 1, it can be seen that the two formation shapes are the same, but $r_i$ is different from each other. Therefore, $\theta \in \mathbb{R}$ is used to denote the shape angle and distinguish positions of robots.

TABLE I
DEFINITION OF SOME IMPORTANT NOTATIONS

| | |
|---|---|
| $r_i$ | The position of $i$th mobile robot. |
| $p$ | The center of multi-mobile robot systems. |
| $\mathcal{P}$ | The plane where mobile robots are located. |
| $\mathcal{S}$ | Formation constraints of multi-robot systems. |
| $\theta$ | The angle of formations. |
| $\mathcal{C}$ | Whole configuration space of a system. |
| $\boldsymbol{c}$ | Configurations of a system, $\boldsymbol{c} = \{x, y, \theta, \mathcal{S}\}$. |
| $\boldsymbol{c}^s$ | Configurations of a system in the same formation shape, $\boldsymbol{c}^s = \{x, y, \theta\}$. |
| $o_i$ | The position of $i$th obstacle. |
| $\mathcal{W}(o_i)$ | Space occupied by obstacle $o_i$. |
| $\mathcal{W}(r_i)$ | Space occupied by robot $r_i$. |
| $\mathcal{C}_{free}$ | A set of valid configurations, $\mathcal{C}_{free} \subset \mathcal{C}$. |
| $\mathcal{C}_{free}^s$ | A set of valid configurations in the same formation shape. |
| $\partial \mathcal{C}_{free}^s$ | The boundary of $\mathcal{C}_{free}^s$. |
| $\mathcal{N}(\boldsymbol{c}^s, \xi)$ | Neighborhood configurations of $\boldsymbol{c}^s$. |
| $\boldsymbol{\Omega}(\boldsymbol{c}^{s,p})$ | Valid angles of the same formation $\mathcal{S}$ with the same $p$. |



(a) Shape with $\theta_1$      (b) Shape with $\theta_2$

Fig. 1. The same formation shape with different angles. $p$ is the same in (a) and (b), while positions of robots are different.

To sum up, the configuration $\boldsymbol{c}$ of multi-mobile robot systems can be described as $\boldsymbol{c} = \{p, \theta, \mathcal{S}\}$, and the whole configuration space is denoted as $\mathcal{C}$. Once $\boldsymbol{c}$ is given, the position of each robot in the workspace can be determined as follows.

$$\{r_1, \ldots, r_n\} = \mathcal{S}^{-1}(p, \theta) \tag{2}$$

where $\mathcal{S}^{-1}$ indicates positions of robots are calculated by the relative position relationship within the formation shape.

Now, let $o_i \subset \mathbb{R}^2, i = 1, \ldots, m$ denotes the position of the $i$th obstacle, and $m$ is the number of obstacles. $\mathcal{W}(o_i) \subset \mathbb{R}^3$ is used to denoted the workspace occupied by the $i$th obstacle. Similarly, $\mathcal{W}(r_i)$ is the workspace occupied by the $i$th robot. The set of valid configurations $\mathcal{C}_{free}$ is defined as follows.

$$\mathcal{C}_{free} := \{\boldsymbol{c} \mid \mathcal{R} \cap \mathcal{O} = \varnothing\}$$
$$\text{where } \mathcal{R} = \mathcal{W}(r_i) \cup \cdots \cup \mathcal{W}(r_n) \tag{3}$$
$$\mathcal{O} = \mathcal{W}(o_i) \cup \cdots \cup \mathcal{W}(o_m)$$

In other words, $\boldsymbol{c} \in \mathcal{C}_{free}$ indicates that the desired formation exists, satisfying that all robots are not in contact with obstacles. Planning configurations in $\mathcal{C}_{free}$ ensures safety during movement. If we directly plan the motion of each robot with formation constraints and obstacle avoidance, it will suffer from the curse of dimensionality. Generally, it is as high as $\mathbb{R}^{2n}$. To avoid this, configurations of systems will be planned first, after which paths of robots can be quickly computed. It can be found that configurations of a system consist of absolute information $\{x, y, \theta\}$ and relative information $\mathcal{S}$. Therefore, planning configurations of a system is divided into two parts. The first part is to plan $\mathcal{S}$, which is more dependent on formation constraints and will be further discussed. $\mathcal{S}$ is strongly related to applications and tasks. The second part is to plan $\{x, y, \theta\}$ for driving the system on $\mathcal{P}$, which is more concerned in this section.

For a multi-mobile system, the planning problem considered in this paper can be formulated as follows.

$$\text{Input} : \mathcal{O}, \mathcal{S}$$
$$\text{Output} : \{\boldsymbol{c}_1, \ldots, \boldsymbol{c}_k\}$$
$$\text{Subject to} \begin{cases} \boldsymbol{c}_i \in \mathcal{C}_{free}, \ \boldsymbol{c}_k \text{ is the goal} \\ \\ \boldsymbol{c}_i \text{ and } \boldsymbol{c}_{i+1} \text{ are connected} \end{cases} \tag{4}$$

Since it is extremely hard to obtain a real continuous space of $\mathcal{C}_{free}$ in the graph, we will solve the problem (4) by translating it using the discretization method. If two configurations can be switched to each other through all available configurations in $\mathcal{C}_{free}$, they are defined as connected. Therefore, the first issue to be studied is the connectivity between configurations after discretizing the workspace.

First, given a formation shape $s$, the configuration space of the same formation is denoted as $\boldsymbol{c}^s = \{x, y, \theta\}$. The set of valid configurations of this formation is defined as $\mathcal{C}_{free}^s$. As shown in Fig. 2, $\boldsymbol{c}_1^s$ and $\boldsymbol{c}_2^s$ are connected because the system can move from $\boldsymbol{c}_1^s$ to $\boldsymbol{c}_2^s$ while keeping the formation continuously unchanged. However, $\boldsymbol{c}_2^s$ and $\boldsymbol{c}_3^s$ are not connected although they are both in $\mathcal{C}_{free}^s$. They can switch to each other only if ignoring the formation constraint. Hence, it is necessary to study the connectivity between different configurations before planning.

**Theorem 1**. *For any $\boldsymbol{c}^s \in \mathcal{C}_{free}^s$, $\boldsymbol{c}^s$ is connected in a neighborhood area $\mathcal{N}(\boldsymbol{c}^s, \xi) := \{\tilde{\boldsymbol{c}}^s\}$, configurations in neighborhood area are defined as follows.*

$$\left\| (\tilde{x} - x)^2 + (\tilde{y} - y^2) \right\|_2 < \xi$$
$$\text{abs}(\tilde{\theta} - \theta) < \xi, \ \xi > 0 \tag{5}$$

*Proof*: $\forall \boldsymbol{c}^s = \{x, y, \theta\} \in \mathcal{C}_{free}^s$, from the definition (3), all robots are separated from obstacles although their distances may be very small. In this case, a specific obstacle-free space $\mathcal{T}_i$ can be found for the $i$th robot, which satisfies

$$\mathcal{W}(r_i) \subset \mathcal{T}_i, \mathcal{T}_i \cap \mathcal{O} = \varnothing \tag{6}$$

$\mathcal{T}_i$ is a bounded open set. Since (6) is available for each robot, the whole formation can move or rotate freely in any direction if $\mathcal{W}(\tilde{r}_i)$ still stays in $\mathcal{T}_i$ after movement, which will generate new valid configurations. Therefore, $\boldsymbol{c}^s$ is connected

Fig. 2. Different valid configurations in $\mathcal{C}^s_{free}$ of a system composed of three mobile robots.



Fig. 3. The change of $\mathbf{\Omega}$ when approaching the boundary.



Fig. 4. The system is forced to change formation in order to pass obstacles.

with these configurations, and the area is denoted as $\mathcal{N}(\boldsymbol{c}^s, \xi)$. Obviously, $\xi$ is strongly related to $\mathcal{T}_i$ and is relatively bigger in an environment without obstacles. It should be pointed out that definition (5) is a little conservative because the whole formation may be able to move bigger than $\xi$ in some directions. However, the conservative definition does not affect overall connectivity. ∎

Theoretically, the real continuous space of $\mathcal{C}_{free}$ can be obtained with infinite discretization. However, the calculation cost is unacceptable. In order to get closer to the real $\mathcal{C}_{free}$ and improve the success rate of subsequent planning, a boundary densification method is introduced. Naturally, we need to look at how to find the boundary $\partial \mathcal{C}^s_{free}$.

Now, we use $\mathbf{\Omega}(\boldsymbol{c}^{s,p})$ to denote all valid angles of the same formation with the same $p$. The maximum range of $\mathbf{\Omega}(\boldsymbol{c}^{s,p})$ is $[0, 2\pi)$. A weak assumption is made for the following statement. When the formation rotates in a fixed $p$, robots and obstacles cannot critically contact all the time. This assumption is quite weak because it occurs only when the shapes of robots and obstacles are always inscribed.

**Theorem 2**. *$\boldsymbol{c}^s$ approaches the boundary $\partial \mathcal{C}^s_{free}$ if and only if $\mathbf{\Omega}(\boldsymbol{c}^{s,p})$ converges to a specific angle. In other words, $\mathbf{\Omega}(\boldsymbol{c}^{s,p}) = \theta^*$ when $\boldsymbol{c}^{s,p} \in \partial \mathcal{C}^s_{free}$.*

*Proof*: Denote the initial position $p = p_0$ and $\boldsymbol{c}^{s,p_0} \in \mathcal{C}^s_{free}$. According to Theorem 1, $\boldsymbol{c}^{s,p_0}$ is connected in $\mathcal{N}(\boldsymbol{c}^{s,p_0}, \xi_0)$, $\boldsymbol{c}^{s,p_1}$ is randomly chosen in $\mathcal{N}(\boldsymbol{c}^{s,p_0}, \xi_0)$, then do

$$(p_{k+1} - p_k) = \lambda_k(p_1 - p_0) \quad \boldsymbol{c}^{s,p_{k+1}} \in \mathcal{N}(\boldsymbol{c}^{s,p_k}, \xi_k) \quad (7)$$

where $\lambda_k$ is a positive coefficient, and this moves the system in one direction. During (7), $\mathcal{N}(\boldsymbol{c}^{s,p_k}, \xi_k)$ may become bigger or smaller depending on the environment. $\boldsymbol{c}^{s,p_{k+1}}$ approaches the boundary $\partial \mathcal{C}^s_{free}$ if and only if $\xi_{k+1} \to 0$. That means, $\mathcal{T}_i \to \mathcal{W}(r_i)$ at least for one robot and the robot is fairly close to obstacles. Due to the assumption, shapes of robots and obstacles are not inscribed. Therefore, the formation cannot rotate around $p_{k+1}$ and $\xi_{k+1} \to 0$. As a result, $\mathbf{\Omega}(\boldsymbol{c}^{s,p_{k+1}})$ will approach the intermittent value $\theta^*$, which indicates that $\mathbf{\Omega}(\boldsymbol{c}^{s,p}) = \theta^*$ when $\boldsymbol{c}^{s,p} \in \partial \mathcal{C}^s_{free}$. A case is shown in Fig. 3, the system forms a rigid formation, where $\mathbf{\Omega}(\boldsymbol{c}^{s,p_0}) = [0, 2\pi)$ and $\mathbf{\Omega}(\boldsymbol{c}^{s,p_k}) \to \theta^*$. It can be seen that $\mathbf{\Omega}$ gradually reduces to a specific angle when approaching the boundary. ∎

Theorem 1-2 are enrolled for unchanged formations. The multi-robot system can move from the initial position to goal position with the desired formation if two configurations are both in $\mathcal{C}^s_{free}$ and connected. But in some situations, the formation is expected or required to change. As shown in Fig. 4, the system can never pass the obstacle if the formation remains unchanged. Therefore, the switch between different formations also needs to be studied. For the sake of brevity but without loss of generality, we assume two formations and the difference between them is small enough (This statement is not rigorous, we will explain further in the following).

**Theorem 3**. *If $\boldsymbol{c}^{s_1,p} \in \mathcal{C}^{s_1}_{free}$, $\boldsymbol{c}^{s_2,p} \in \mathcal{C}^{s_2}_{free}$, and $\exists \theta_0 \in \mathbf{\Omega}(\boldsymbol{c}^{s_1,p}) \cap \mathbf{\Omega}(\boldsymbol{c}^{s_2,p})$, two formations are switchable in p.*

*Proof*: Assume robots form the first formation $s_1$ with $\theta_0$ in $p$. The position of $i$th robot is $r_1^{s_1}$. According to Theorem 1, $\mathcal{W}(r_i^{s_1}) \subset \mathcal{T}_i$. The difference between two formations is small enough means $\mathcal{W}(r_i^{s_1}) \subset \mathcal{T}_i$ and $\mathcal{W}(r_i^{s_2}) \subset \mathcal{T}_i$ for all robots. Then, the formation can switch from $s_1$ to $s_2$. Though Theorem 3 is for two formations with relatively small differences, it can be used to test whether two formations are switchable. This is more practical in later applications. ∎

In the above statement, we adequately describe configurations of multi-mobile robot systems. The connectivity between configurations of the same formation and switching between different formations have been studied in Theorem 1-3. Based on these, our planner is designed, which will be further introduced in the next section.

## IV. THE PROPOSED PLANNER

For a given multi-mobile robot system and an environment with obstacles, we assume that the desired formations $\mathcal{S}$ have been determined. The framework of our planner is shown in Fig. 5. The environment is firstly discretized into cells, which

Fig. 5. The general framework of our planner. The workspace is mapped into the valid configuration space with boundary densification. An undirected connected graph is generated by verifying the connectivity of valid configurations. Whether a path exists is quickly determined on the graph without cost. The optimal path is planned on the updated graph considering path length and formation preference. Then, motions of robots are computed on the path.

are mapped into the valid configuration space of the system. Areas of the configuration space near the boundary are further refined. Then, the connectivity of two configurations is verified to generate an undirected connected graph. Initially, there is no specific cost in the graph, and the breadth-first search method [40] can be applied to quickly decide whether a feasible path exists or not. If the answer is yes, a cost function based on the path lengths of robots and formation preference is assigned to update the graph. Finally, the Dijkstra method [41] is applied to find the optimal path in the updated graph. Our planner is able to be combined with different formation constraints in obstacle-rich environments. Moreover, this planner can be extended in some special situations with additional constraints, which will be discussed in the next section.

### A. Mapping from Workspace to Configuration Space

The first step of our planner is to map the points $p$ on the plane of robots to the configurations $c$ of the multi-mobile robot system. Mapping between workspace and configuration space has been widely used in robot planning. In this paper, full mapping is considered to find the optimal planning path. The mapping $\mathcal{M}$ between $\mathcal{P}$ and $\mathcal{C}$ is built as follows.

$$\begin{aligned}
\mathcal{M}(p) &:= \{c \mid x_c = x_p, y_c = y_p, c \in \mathcal{C}\} \\
\mathcal{M}(c) &:= \{p \mid x_p = x_c, y_p = y_c, p \in \mathcal{P}\}
\end{aligned} \quad (8)$$

In our multi-mobile robot systems, configurations are denoted as $c = \{x, y, \theta, \mathcal{S}\}$. Obviously, $\mathcal{M}(p)$ is a one-to-many mapping, and $\mathcal{M}(c)$ is a many-to-one mapping.

To solve the problem through graph theory, we need to discretize the space. As explained before, $x, y \in \mathcal{P}$ and $\theta \in \Omega$ are used to denote the absolute locations of robots. $\mathcal{P}$ is discretized by $g$, and $\Omega \subset [0, 2\pi)$ is discretized by $\alpha$. Discrete scales $g$ and $\alpha$ are crucial parameters in the planner, determining the distance between adjacent robot waypoints. In some cases, discrete scale is also called resolution or grid size. $g$ is chosen according to the following rules.
(1) $g \in [g_{\min}, g_{\max}]$.

---

**Algorithm 1:** Configurations Mapping

**Input:** Multi-robot systems; Discrete scales $g$, $\alpha$, $g_{\min}$; Formation constraints $\mathcal{S}$; Environment with obstacles $\{o_1, \ldots, o_m\}$.
**Output:** Valid configurations $c \in \mathcal{C}_{free}$.

1  Initializing: $\mathcal{C}_{free} \leftarrow \varnothing$;
2  $p_i \leftarrow \texttt{PlaneDiscretized}(g, \alpha)$;
3  **foreach** $p_i$ **do**
4      $c \leftarrow \mathcal{M}(p_i)$;
5      **if** $\texttt{ValidConfiguration}(c)$ is true **then**
6         $c$ is added to $\mathcal{C}_{free}$;
7      **end**
8  **end**
9  **foreach** $c \in \mathcal{C}_{free}$ **do**
10     **if** $\texttt{BoundaryDetect}(c)$ is true **then**
11        $p_i \leftarrow \mathcal{M}(c)$;
12        $p_{\text{new}} \leftarrow$ $\texttt{BoundaryDiscretized}(p_i, g_{\min}, \alpha)$;
13        **foreach** $p_{\text{new}}$ **do**
14           Repeat 4-7;
15        **end**
16     **end**
17 **end**

---

(2) $g_{\min}$ is related to computational cost and accuracy. The control accuracy of robots is limited and can never reach such a resolution as small as possible. Therefore, $g_{\min}$ cannot be selected too small. Otherwise, it will be impractical or computationally unacceptable.

(3) $g_{\max}$ is related to the connectivity of configurations. This is decided by the size of obstacles and robots. Path planning may fail if $g_{\max}$ is selected too big.

In general, $g$ is set within a reasonable range in order to solve the problem practically and practicably. The choice of $\alpha$ is similar to $g$. It cannot be too small because of the accuracy, nor can it be too big to cross obstacles during rotation. Once discrete scales are determined, for each formation constraint

Fig. 6. A plane with boundary densification. It is firstly discretized by $g$, generating the blue points, which can be mapped into valid configurations. After finding areas near the boundary, it is further discretized by $g_{\min}$, generating the red points.

---

**Algorithm 2:** Planning on Graph

**Input:** Valid configurations $c \in \mathcal{C}_{free}$; Cost function.
**Output:** Robot path $R_1, \ldots, R_n$.

1   Initializing: $R_1, \ldots, R_n \leftarrow \varnothing$, $\mathcal{G}(\mathcal{V}, \mathcal{E}) \leftarrow \varnothing$.
2   **foreach** $c_i \in \mathcal{C}_{free}$ **do**
3     $c_i$ is added to $\mathcal{V}$ as $v_i$;
4   **end**
5   **for** $v_i, v_j \in \mathcal{V}$ **do**
6     **if** `ConnectionDetect`$(v_i, v_j)$ is true **then**
7       $e_{ij}$ is added to $\mathcal{E}$;
8     **end**
9   **end**
10   $\Gamma \leftarrow$ `BFS`$(\mathcal{G})$;
11   **if** $\Gamma$ *exists* **then**
12     $\mathcal{G}' \leftarrow$ `CostFunction`$(\mathcal{G})$;
13     $\Gamma \leftarrow$ `Dijkstra`$(\mathcal{G}')$;
14     $R_1, \ldots, R_n \leftarrow$ `RobotPath`$(\Gamma)$;
15     **return** $R_1, \ldots, R_n$;
16   **else**
17     **return** $R_1, \ldots, R_n$;
18   **end**

---

$s$, the 3D configuration space $\mathcal{C}_{free}^s$ can be built. Valid configurations near the boundary $\partial \mathcal{C}_{free}^s$ are selected. Then, $p$ near the boundary in $\mathcal{P}$ is computed by $\mathcal{M}(c)$. For areas on $\mathcal{P}$ near the boundary, the minimal discrete scale $g_{\min}$ will be used to discretize them further. The newly resulting $p$ is added to the mapping. In this way, the most realistic situation of $\mathcal{C}_{free}^s$ can be approximated to the greatest extent without excessively increasing the computation. An example is given in Fig. 6. The blue points represent $p$ discretized by $g$, and these points can be mapped into valid configurations. After finding the $\mathcal{M}(p)$ near $\partial \mathcal{C}_{free}^s$, the plane is further discretized by $g_{\min}$, generating the red points. Twice discretization can make the boundary denser, thereby increasing the success rate of planning, which will be demonstrated by simulations later. The whole process of mapping is shown in Algorithm 1.

### B. Connected Graph Generator

The second step is to generate an undirected connected graph through valid configurations. Graph theory is usually used to describe a relationship between certain things. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denotes a connected graph, where $\mathcal{V} = \{v_i\}$ is the vertex set and $\mathcal{E} = \{e_{ij} := (v_i, v_j)\}$ is the edge set. Graphs can be divided into directed graphs and undirected graphs, depending on $e_{ij}$ and $e_{ji}$. If $e_{ij} = e_{ji}$ for all edges, the graph is undirected. Otherwise, the graph is directed. In path planning, the points or positions of robots are stored in $\mathcal{V}$, and costs between points are stored in $\mathcal{E}$. The task is to schedule a path from the initial vertex to the goal vertex. Due to previous research, many quick and convenient planning methods can be directly applied.

In our problem, $v_i$ represents a configuration, which has been built by Algorithm 1. Then, it needs to find $e_{ij}$, which represents the cost of configuration switch. To improve planning efficiency, whether two configurations can be transformed

into each other will be determined primarily. The specific cost is not considered at first. The connectivity between configurations can be verified according to Theorem 1 and 3. We exclusively consider the connectivity of two adjacent configurations, which means only one dimension is different in $c$. Although this is incomplete in finding $e_{ij}$, it does not affect the overall connected graph and will significantly decrease the computation. $e_{ij}$ is added to the edge set $\mathcal{E}$ if $v_i$ and $v_j$ are connected. Now, an undirected connected graph $\mathcal{G}$ is obtained. In $\mathcal{V}$, the initial and goal configurations are denoted as $v_{\text{init}}$ and $v_{\text{goal}}$, respectively. For $\mathcal{G}$, we can use the Breadth First Search (BFS) method to quickly search whether there is a path from $v_{\text{init}}$ to $v_{\text{goal}}$. If it exists, we will calculate the specific cost in $\mathcal{E}$ and then find the optimal path. If it does not exist, path planning fails, and all subsequent calculations will be saved.

There are three options for configuration change: movement, rotation, and formation change. A cost function is designed based on the consideration of path lengths of robots and formation preference as follows.

$$e_{ij} = \begin{cases} \sum_{k=1}^{N} \lambda_s \left\| r_k^{v_i} - r_k^{v_j} \right\|_2 & \text{move or rotate} \\ \\ \sum_{k=1}^{N} \left\| r_k^{v_i} - r_k^{v_j} \right\|_2 & \text{change formation} \end{cases} \quad (9)$$

where $\sum_{k=1}^{N} \left\| r_k^{v_i} - r_k^{v_j} \right\|_2$ is the total distance travelled by all robots. $\lambda_s$ is the indicator of preference. The system will try to maintain the desired formation with a smaller $\lambda_s$. As a result, we can change the tendency of formations by controlling the relative relationship between $\lambda_s$ of different formations. After calculating the specific cost, $\mathcal{G}$ is updated to $\mathcal{G}'$. Dijkstra algorithm can be enrolled to find the optimal path in $\mathcal{G}'$. The whole planning process is shown in Algorithm 2. Searching twice by different methods is trying to save time and computational cost.

## V. Case Study in Object Transportation

In this paper, we take multi-robot object transportation as a typical example. There are not only constraints brought by formations but also by objects in transportation [42]. Therefore, it is more challenging than other scenarios. Generally, multi-robot transportation systems are composed of mobile robots and the carrier. Carriers can be mobile robot itself [43]–[45], deformable sheet [46]–[48], or manipulator [49]–[51], which result in different system model and formation constraints. Still and all, they can be solved by our planner, and the first two will be used as examples in this paper.

Now, a little modification is made to the previous definition. $p = [x, y]$ is used to denote the position of object, and $\mathcal{W}(p)$ is used to denote the space occupied by object and carriers. There is a certain relationship as follows.

$$p = \mathfrak{F}(r_1, \ldots, r_n) \tag{10}$$

where $\mathfrak{F}$ is the mapping between positions of robots and the object, which is directly determined by the carrier. As explained before, formation constraints $\mathcal{S}$ are the internal relative information of multi-robot systems. Therefore, if we care about the relative positional relationship within the system, Eq.(10) can be rewritten as follows.

$$\hat{p} = \mathfrak{F}(\mathcal{S}) \tag{11}$$

where $\hat{p}$ is the relative position of objects inside the system. Generally, Eq.(10) is used to describe the absolute locations of the transportation system in the workplace, and Eq.(11) is used for relative positions within the system. Since there are new constraints brought by the object and carriers, the definition of valid configurations (3) is changed as follows.

$$\mathcal{C}_{free} := \{c \mid \mathcal{W}(p) \cap \mathcal{R} \cap \mathcal{O} = \varnothing\} \tag{12}$$

The newly added constraints ensure that the object does not collide with robots and obstacles during transportation. Desired formations are determined according to Eq.(10)-(12). For example, two cases are given later.

### A. Transportation without Carrier

In some systems, there are no special carriers and mobile robots themselves are regarded as carriers. These systems are cheap and easy to control. This is the simplest case because robots are often physically attached to objects, which means the formation constraint $\mathcal{S}$ is fixed. During transportation, relative positions between robots and objects are invariably kept as the initial. Consequently, valid configuration space is reduce to $\mathcal{C}_{free}^{\mathcal{S}} \subset \mathbb{R}^3$. In the fixed $\mathcal{S}$, the whole system can be considered as a rigid body to avoid obstacles.

### B. Transportation with Deformable Sheet

Multi-robot transportation with a deformable sheet as the carrier is shown in Fig. 7. The transported objects are placed on the deformable sheet held by robots. These systems are universal to different shapes of objects and the position of object can be manipulated by adjusting relative positions between robots.



(a) Physical model  (b) Simulation model

Fig. 7.   Multi-robot transportation with a deformable sheet as the carrier.

The model of these systems has been studied in our previous work [47]. A computational approach based on the Virtual Variable Cables Model(VVCM) is presented for calculating Eq.(10), which simplifies the system to a robots-cables-payload system. Then, $\mathfrak{F}$ is replaced by VVCM as follows.

$$p = VVCM(r_1, \ldots, r_n) \tag{13}$$

Eq.(13) assumes that the position of object is at the state of minimum gravitational potential energy solved by convex optimization. It should be pointed out that the minimal system is composed of three robots. For the minimal systems, Eq.(10) can be directly calculated by VVCM. For systems with a higher number of robots, there are multiple local static equilibrium states, resulting in multiple solutions. To deal with the multiple solutions, an optimal formation function is introduced.

$$J(s) = J_{\text{trans}} + J_{\text{pass}} + J_{\text{cross}}$$
$$s^* = \text{argmin}\ J(s) \tag{14}$$

where $J_{\text{trans}}$ ensures the safety and stability of systems during transportation. Valid configurations are included in $J_{\text{pass}}$ and $J_{\text{cross}}$, where the former is to bypass obstacles, the latter is to cross obstacles. For an environment with obstacles, optimal formations are obtained by Eq. (13) and (14). With the generated optimal formations $\mathcal{S}$, the planning problem can be solved by our planner.

## VI. Simulations and Discussions

### A. Simulation Setup

In this section, we have designed three scenarios for multi-mobile robot systems under different formation constraints to test our planner. These systems are placed in an environment containing a large number of randomly generated obstacles. All our simulations are performed on a computer with Intel Core i7-10700 CPU at 2.90 GHz (16 cores) and 64GB RAM. The results are obtained by Gazebo and Rviz.

### B. Simulation Results

#### 1) Maintaining Rigid Formations

In the first part, mobile robots are required to form a rigid formation while traversing an environment with obstacles. Since the formation is rigid, which indicates that $\mathcal{S}$ is given and onefold, the valid configuration of systems is reduced to three-dimension. We set up an environment of 10m×10m. The

(a) A four-robot system



(b) A six-robot system

Fig. 8. Multi-mobile robot systems traverse obstacle-rich environments while maintaining rigid formations. (a) Four robots form a square in a 10m×10m environment. There are 65 obstacles with random positions and radii. (b) Six robots form a hexagon in a 15m×15m environment. There are 90 obstacles with random positions and radii.



Fig. 9. A multi-mobile robot system composed of four robots transports a square object in a 10m×10m environment. There are 30 obstacles with random positions and radii. The system is forced to bypass obstacles like a 2D whole.

initial position of formation center is (1.5,1.5)m, and the target is (8.5,8.5)m. On the premise of ensuring that the initial and target position exists, some obstacles with random positions are generated in the environment, whose radiuses are also randomly selected between 0.05m and 0.1m. The radius of each robot is 0.35m. The results are shown in Fig. 8. In Fig. 8(a), four robots try to maintain a square formation. There are 65 obstacles in the environment. The average path length of each robot is 19.1599m on the optimally planned path. Similarly, in Fig. 8(b), six robots try to maintain a hexagon, and there are 90 obstacles in an environment of 15m×15m. The initial position of formation center is (2.5,2.5)m while the target is (12.5,12.5)m. The average path length of each robot is 24.7397m. It can be seen that the two multi-mobile robot systems are able to traverse the obstacle-rich environments while holding desired formations invariably unchanged. To our best knowledge, no existing planner for multi-robot systems can deal with these environments.

It should be pointed out that our planner may also fail in these environments since all obstacles are randomly generated. Therefore, we repeat lots of times for the system composed of four robots under different numbers of obstacles. Moreover, in order to study the effect of discrete scales in the planner, we also repeat under different scales. The results are given in Table II. For each time, obstacles are randomly generated while the initial and goal positions are fixed, and the environment is

discretized only once by given scales. It can be seen that under the same discrete scales, as the number of obstacles increases, the success rate generally decreases, and the time spent on mapping from workspace to configuration space increases while the time spent on planning on the connected graph decreases. Essentially, the whole configuration space is limited and the same for determined discrete scales. When obstacles increase, the time for validating valid configurations increases, and obviously, the size of valid configuration space decreases. Then, the size of connected graph decreases, resulting in less time for planning. It can also be seen that for the same number of obstacles, the success rate and total time are both larger under smaller discrete scales. However, the success rate cannot be significantly improved by minifying discrete scales when the number of obstacles is more than 80. This means the workplace is too harsh for the system.

As shown in Table II, decreasing the discrete scales will significantly increase the computational cost. Therefore, we try our proposed boundary densification method. The minimal scale $g_{\min} = 0.025$, and the results are given in Table III. The relationship between the time spent on mapping and the number of obstacles is no longer linear since the time on `BoundaryDetect` in Algorithm 1 is irregular. Compared with Table II, for the same environments, the success rate is higher with boundary densification, as indicated by the numbers marked in red and blue. Sometimes the success rate

TABLE II
REPEAT Fig. 8(a) UNDER DIFFERENT CONDITIONS WITHOUT BOUNDARY DENSIFICATION

| Obstacles | Discrete Scale: 0.08m 0.04rad | | | | Discrete Scale: 0.05m 0.04rad | | | | Discrete Scale: 0.025m 0.04rad | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Success Rate | Average Time (s) | | Total | Success Rate | Average Time (s) | | Total | Success Rate | Average Time (s) | | Total |
| | | Mapping | Planning | | | Mapping | Planning | | | Mapping | Planning | |
| 10 | 100/100 | 2.12479 | 0.50741 | 2.63220 | 100/100 | 5.26545 | 1.39201 | 6.65746 | 100/100 | 20.87622 | 5.98285 | 26.85907 |
| 20 | 99/100 | 2.18363 | 0.34327 | 2.52690 | 99/100 | 5.42517 | 0.82054 | 6.24571 | 99/100 | 21.50460 | 3.87619 | 25.38079 |
| 30 | 93/100 | 2.22332 | 0.22335 | 2.44667 | 93/100 | 5.53143 | 0.59595 | 6.12738 | 94/100 | 21.93061 | 2.76713 | 24.69774 |
| 40 | 67/100 | 2.25200 | 0.14657 | 2.39857 | 70/100 | 5.59970 | 0.40005 | 5.99975 | 77/100 | 22.30289 | 1.80580 | 24.10869 |
| 50 | 36/100 | 2.28206 | 0.10722 | 2.38928 | 41/100 | 5.67768 | 0.31582 | 5.99350 | 50/100 | 22.64238 | 1.31906 | 23.96144 |
| 60 | 23/100 | 2.31072 | 0.07745 | 2.38817 | 23/100 | 5.78103 | 0.21294 | 5.99397 | 26/100 | 22.95013 | 0.91771 | 23.86784 |
| 70 | 2/100 | 2.32667 | 0.05235 | 2.37902 | 2/100 | 5.84270 | 0.15498 | 5.99768 | 5/100 | 23.17927 | 0.66619 | 23.84546 |
| 80 | 0/100 | 2.34902 | 0.04201 | 2.39103 | 1/100 | 5.89965 | 0.11418 | 6.01383 | 1/100 | 23.34135 | 0.50529 | 23.84664 |
| 90 | 1/100 | 2.36974 | 0.03097 | 2.40071 | 1/100 | 5.95067 | 0.10586 | 6.05653 | 1/100 | 23.69105 | 0.42129 | 24.11234 |
| 100 | 0/100 | 2.38406 | 0.02386 | 2.40792 | 0/100 | 5.98362 | 0.07432 | 6.05794 | 0/100 | 23.90271 | 0.31573 | 24.21844 |

TABLE III
REPEAT Fig. 8(a) UNDER DIFFERENT CONDITIONS WITH BOUNDARY DENSIFICATION

| Obstacles | Discrete Scale: 0.08m/0.025m 0.04rad | | | | Discrete Scale: 0.05m/0.025m 0.04rad | | | | Discrete Scale: 0.025m 0.04rad | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Success Rate | Average Time (s) | | Total | Success Rate | Average Time (s) | | Total | Success Rate | Average Time (s) | | Total |
| | | Mapping | Planning | | | Mapping | Planning | | | Mapping | Planning | |
| 10 | 100/100 | 2.73023 | 1.19019 | 3.92042 | 100/100 | 6.57826 | 2.66315 | 9.24141 | 100/100 | 20.87622 | 5.98285 | 26.85907 |
| 20 | 99/100 | 2.77770 | 0.91127 | 3.68897 | 99/100 | 6.50419 | 1.97885 | 8.48304 | 99/100 | 21.50460 | 3.87619 | 25.38079 |
| 30 | 93/100 | 2.80989 | 0.74911 | 3.55900 | 94/100 | 6.42345 | 1.46426 | 7.88771 | 94/100 | 21.93061 | 2.76713 | 24.69774 |
| 40 | 69/100 | 2.76320 | 0.53806 | 3.30126 | 72/100 | 6.38639 | 1.06474 | 7.45113 | 77/100 | 22.30289 | 1.80580 | 24.10869 |
| 50 | 37/100 | 2.75822 | 0.40120 | 3.15942 | 47/100 | 6.48182 | 0.84711 | 7.32893 | 50/100 | 22.64238 | 1.31906 | 23.96144 |
| 60 | 23/100 | 2.67300 | 0.29691 | 2.96991 | 24/100 | 6.47801 | 0.64271 | 7.12072 | 26/100 | 22.95013 | 0.91771 | 23.86784 |
| 70 | 2/100 | 2.65702 | 0.22995 | 2.88697 | 3/100 | 6.33030 | 0.45257 | 6.78287 | 5/100 | 23.17927 | 0.66619 | 23.84546 |
| 80 | 0/100 | 2.62448 | 0.17087 | 2.79535 | 1/100 | 6.28102 | 0.34993 | 6.63095 | 1/100 | 23.34135 | 0.50529 | 23.84664 |
| 90 | 1/100 | 2.58503 | 0.12706 | 2.71209 | 1/100 | 6.33555 | 0.29529 | 6.63084 | 1/100 | 23.69105 | 0.42129 | 24.11234 |
| 100 | 0/100 | 2.50583 | 0.07612 | 2.58195 | 0/100 | 6.25361 | 0.22073 | 6.47434 | 0/100 | 23.90271 | 0.31573 | 24.21844 |

is even close to the case with minimal scales. However, the computational cost is far less than discretization directly by minimal scales. Therefore, the proposed boundary densification method can facilitate planning.

*2) Object Transportation without Carrier*

In the second part, mobile robots are required to transport objects without a carrier, which also needs robots to maintain rigid formations. But different from the first part, there are additional constraints brought by objects. Robots and environment are the same as Fig. 8(a). The transported object is a square with a side length of 1.5m. There are 30 obstacles randomly generated, and the results are shown in Fig. 9. The system also can reach the goal position without collision, and the average path length of each robot is 18.4439m. Since obstacles are all higher than the system, constraints of formations and safe transportation can be synthesized, which is that the 2D outline of the whole system does not overlap obstacles. Compared with the first part, the obstacle-crossing ability of this system is observably reduced. It has to merely bypass obstacles and cannot cross them.

*3) Object Transportation with Deformable Sheet*

In the third part, mobile robots are required to transport objects with a deformable sheet, the model of which has been studied in our previous work [47]. Now, formation constraints are no longer rigid, and the system is able to change formations to bypass or cross obstacles depending on the height of transported object. We first test a three-robot system. Each side length of the original deformable sheet is 1.6m, and the height of each contact point between robots and the sheet is 1m. These parameters will be used to calculate the model

as described in Eq.(13). The environment is also 10m×10m, and the radius of each robot is 0.15m. The initial position of formation center is (1.5,1.5)m, and the target is (8.5,8.5)m. There are 80 obstacles randomly generated with random radius between 0.1m and 0.15m, among which 30 are 0.25m in height, 40 are 0.42m in height, and 10 are 1m in height. Limited by formation constraints and the deformable sheet, the system has to bypass some obstacles. $\lambda_s$ is set as the smallest for the formation of a triangle with each side length of 0.7m. The results are shown in Fig. 10, and the deformable sheet is simplified to a robots-cables-payload system as described before. Obstacles of different heights are distinguished by different colors. The system can traverse this obstacle-rich environment without collision. The average path length of each robot is 17.1758m. In Fig. 10(c), it can be seen that the system has to change formations to uplift transported object to cross some obstacles. The height of object during transportation is shown in Fig. 10(d).

We secondly test a four-robot system in a more complex environment including a narrow corridor. The environment is 16m×4m and contains 70 obstacles. $\lambda_s$ is set as the smallest for the formation of a square with each side length of 1.1314m. The results are shown in Fig. 11. The average path length of each robot is 29.1217m. For comparison, we tried the motion planning method in [22]. The strategy is to find the largest obstacle-free space containing the system, inside which motions of robots are planned. Due to ignoring heights, the system is considered on the 2D plane. The result is shown in Fig. 12. The red polygon is the largest obstacle-free space currently containing the robot system. Motions of robots can

(a) Pictures during transportation

(b) Top view

(c) Robot's view

(d) Object's height

Fig. 10. A multi-mobile robot system composed of three robots transport objects with a deformable sheet in a 10m×10m environment. The deformable sheet is simplified to a robots-cables-payload system. There are 80 obstacles with random positions and radii. Obstacles of different heights (0.25m, 0.42m, 1m) are distinguished by different colors. (a) Pictures during transportation. (b) Top view of the whole environment. (c) Robot's view. It can be seen that some obstacles can only be bypassed. (d) The height of object during transportation.



(a) Pictures during transportation

(b) Top view

(c) Robot's view

(d) Object's height

Fig. 11. A multi-mobile robot system composed of four robots transport objects with a deformable sheet in a 16m×6m environment. There are 70 obstacles. (a) Pictures during transportation. (b) Top view of the whole environment. (c) Robot's view. (d) The height of object during transportation.

be planned freely in the red polygon which changes with the movement of the system. It can greatly avoid collisions and maintain desired formations. However, the method failed after several movements because the obstacle-free space could not be found anymore for the system to move toward the destination. This method is not appropriate in such obstacle-rich environments.

## C. Discussion

In the planner, we propose a novel mapping between workspace and configuration space of multi-mobile robot systems, which can be efficiently used in planning. Besides, a boundary densification method is added for facilitating planning. The planner can deal with different formation constraints and additional constraints brought by special tasks. We take object transportation as a typical example to test our planner. Simulation results show that the planner can be applied to obstacle-rich environments. In the second part, the system is regarded as a whole on the 2D plane to avoid obstacles, which has also appeared in other papers. Possibly, the problem may be solved by the methods in [22], [24], [28], [45]. However, these works have not been tested in such a complex environment. As for the first and third parts, to the best of our knowledge, no other existing methods can achieve the same effect. State-of-the-art methods are either unable to maintain formations or cannot be used in these environments. Despite taking some specific examples in the simulations, the planner can be potentially used in many scenarios as long as

Fig. 12.  Motion planning by the method in [22]. The largest obstacle-free space currently containing the robot system is denoted by the red polygon. Robots can move freely in the red polygon. However, it cannot drive the system to the destination if considering the formation constraints.

formation constraints are determined. However, our planner also has some limitations. First, discrete scales are difficult to be chosen in different environments. There is no existing path in the simulations if the scales are set inappropriately. The choice of scales dominates our planner. Moreover, it remains uncertain whether there is a path or scales are inappropriate if planning fails. Instructively, discrete scales are selected according to control accuracy and obstacle size. Scales smaller than the control accuracy are not practically meaningful, and only theoretically feasible solutions exist in this case. Second, the time spent on planning is too heavy, especially when $\mathcal{S}$ contains many formations. Now, the planner cannot be applied in an environment with dynamic obstacles. Since the mapping is full, each valid configuration space of different formations needs to be built. Indeed, sometimes it is unnecessary and wasteful.

## VII. Conclusion and Future Work

This paper proposes a novel multi-mobile robot motion planner with excellent obstacle avoidance ability under formation constraints. The optimal motions of robots can be planned in obstacle-rich environments. Valid configurations of systems are defined to satisfy formation constraints. Then, a mapping between workspace and configuration space is introduced with theoretical analysis. An undirected connected graph is generated to solve the planning problem, on which some existing search methods can be used efficiently. Moreover, a boundary densification method is used for improving the success rate of planning. The generality and effectiveness of the planner are proved by implementing different cases. Despite taking specific examples, the framework of our planner can be potentially employed in other scenarios.

Regarding future work, we will further improve the efficiency of the planner to save time. Since full mapping is unnecessary sometimes, a hierarchical planning structure may be considered. And, a trade-off between planning time and approximate optimal solution can also be enhanced. Obviously, the planner cannot be used in environments with dynamic obstacles. In this regard, a possible direction is to combine the planner as a global planner with some local obstacle avoidance methods. When the multi-mobile robot system approaches a dynamic obstacle, local methods will be employed for real-time obstacle avoidance. More practically, we also will

study the disturbance problem when tracking the planned path. Sensors may be carried by robots in order to react to unforeseen events. Then, physical experiments can be carried out.

## References

[1] J. S. Jennings, G. Whelan, and W. F. Evans, "Cooperative search and rescue with a team of mobile robots," in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR'97*. IEEE, 1997, pp. 193–200.

[2] V. Shree, B. Asfora, R. Zheng, S. Hong, J. Banfi, and M. Campbell, "Exploiting natural language for efficient risk-aware multi-robot sar planning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3152–3159, 2021.

[3] C. Nieto-Granda, J. G. Rogers III, and H. I. Christensen, "Coordination strategies for multi-robot exploration and mapping," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 519–533, 2014.

[4] M. Dogar, R. A. Knepper, A. Spielberg, C. Choi, H. I. Christensen, and D. Rus, "Multi-scale assembly with robot teams," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1645–1659, 2015.

[5] P. Culbertson, J.-J. Slotine, and M. Schwager, "Decentralized adaptive control for collaborative manipulation of rigid bodies," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1906–1920, 2021.

[6] J. P. Van Den Berg and M. H. Overmars, "Roadmap-based motion planning in dynamic environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 885–897, 2005.

[7] M. Peasgood, C. M. Clark, and J. McPhee, "A complete and scalable strategy for coordinating multiple robots within roadmaps," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 283–292, 2008.

[8] S. Kloder and S. Hutchinson, "Path planning for permutation-invariant multirobot formations," *IEEE Transactions on Robotics*, vol. 22, no. 4, pp. 650–665, 2006.

[9] T. Balch and R. C. Arkin, "Behavior-based formation control for multirobot teams," *IEEE transactions on robotics and automation*, vol. 14, no. 6, pp. 926–939, 1998.

[10] W. L. Seng, J. C. Barca, and Y. A. Sekercioglu, "Distributed formation control in cluttered environments," in *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2013, pp. 1387–1392.

[11] T. Chi, C. Zhang, Y. Song, and J. Feng, "A strategy of multi-robot formation and obstacle avoidance in unknown environment," in *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2016, pp. 1455–1460.

[12] G. Wen, C. P. Chen, and Y.-J. Liu, "Formation control with obstacle avoidance for a class of stochastic multiagent systems," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5847–5855, 2017.

[13] H. Rezaee and F. Abdollahi, "A decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 1, pp. 347–354, 2013.

[14] D. Zhou, Z. Wang, and M. Schwager, "Agile coordination and assistive collision avoidance for quadrotor swarms using virtual structures," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 916–923, 2018.

[15] H. M. La, R. Lim, and W. Sheng, "Multirobot cooperative learning for predator avoidance," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 1, pp. 52–63, 2014.

[16] C. Bai, P. Yan, W. Pan, and J. Guo, "Learning-based multi-robot formation control with obstacle avoidance," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[17] X. Zhang, L. Yan, T. L. Lam, and S. Vijayakumar, "Task-space decomposed motion planning framework for multi-robot loco-manipulation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 8158–8164.

[18] Q. Tang, Y. Zhang, F. Yu, and J. Zhang, "An obstacle avoidance approach based on system outlined rectangle for cooperative transportation of multiple mobile manipulators," in *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*. IEEE, 2018, pp. 533–538.

[19] Y. Li and K. Gupta, "Real-time motion planning of multiple formations in virtual environments: Flexible virtual structures and continuum model," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2008, pp. 1902–1907.

[20] J. Jiao, Z. Cao, N. Gu, S. Nahavandi, Y. Yang, and M. Tan, "Transportation by multiple mobile manipulators in unknown environments with obstacles," *IEEE Systems journal*, vol. 11, no. 4, pp. 2894–2904, 2015.

[21] J. Alonso-Mora, R. Knepper, R. Siegwart, and D. Rus, "Local motion planning for collaborative multi-robot manipulation of deformable objects," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 5495–5502.

[22] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control and object transport in dynamic environments via constrained optimization," *The International Journal of Robotics Research*, vol. 36, no. 9, pp. 1000–1021, 2017.

[23] D. Roy, A. Chowdhury, M. Maitra, and S. Bhattacharya, "Multi-robot virtual structure switching and formation changing strategy in an unknown occluded environment," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4854–4861.

[24] H. Song, A. Varava, O. Kravchenko, D. Kragic, M. Y. Wang, F. T. Pokorny, and K. Hang, "Herding by caging: a formation-based motion planning framework for guiding mobile agents," *Autonomous Robots*, vol. 45, no. 5, pp. 613–631, 2021.

[25] J. Fink, M. A. Hsieh, and V. Kumar, "Multi-robot manipulation via caging in environments with obstacles," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 1471–1476.

[26] Z. Wang and M. Schwager, "Force-amplifying n-robot transport system (force-ants) for cooperative planar manipulation without communication," *The International Journal of Robotics Research*, vol. 35, no. 13, pp. 1564–1586, 2016.

[27] T. Machado, T. Malheiro, S. Monteiro, W. Erlhagen, and E. Bicho, "Multi-constrained joint transportation tasks by teams of autonomous mobile robots using a dynamical systems approach," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 3111–3117.

[28] D. Koung, O. Kermorgant, I. Fantoni, and L. Belouaer, "Cooperative multi-robot object transportation system based on hierarchical quadratic programming," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6466–6472, 2021.

[29] K. Solovey and D. Halperin, "k-color multi-robot motion planning," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 82–97, 2014.

[30] J. Yu, "Intractability of optimal multirobot path planning on planar graphs," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 33–40, 2015.

[31] A. Adler, M. de Berg, D. Halperin, and K. Solovey, "Efficient multi-robot motion planning for unlabeled discs in simple polygons," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 4, pp. 1309–1317, 2015.

[32] J. Yu and S. M. LaValle, "Planning optimal paths for multiple robots on graphs," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3612–3617.

[33] J. Yu and S. M. LaValle, "Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics," *IEEE Transactions on Robotics*, vol. 32, no. 5, pp. 1163–1177, 2016.

[34] J. Salvado, M. Mansouri, and F. Pecora, "Combining multi-robot motion planning and goal allocation using roadmaps," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 10 016–10 022.

[41] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, 1987.

[35] W. Wu, S. Bhattacharya, and A. Prorok, "Multi-robot path deconfliction through prioritization by path prospects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9809–9815.

[36] J. Motes, R. Sandström, H. Lee, S. Thomas, and N. M. Amato, "Multi-robot task and motion planning with subtask dependencies," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3338–3345, 2020.

[37] D. Le and E. Plaku, "Multi-robot motion planning with dynamics via coordinated sampling-based expansion guided by multi-agent search," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1868–1875, 2019.

[38] S. D. Han, E. J. Rodriguez, and J. Yu, "Sear: A polynomial-time multi-robot path planning algorithm with expected constant-factor optimality guarantee," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[39] W. Wan and R. Fukui, "Efficient planar caging test using space mapping," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 278–289, 2016.

[40] D. J. Rose, R. E. Tarjan, and G. S. Lueker, "Algorithmic aspects of vertex elimination on graphs," *SIAM Journal on computing*, vol. 5, no. 2, pp. 266–283, 1976.

[42] E. Tuci, M. H. Alkilabi, and O. Akanyeti, "Cooperative object transport in multi-robot systems: A review of the state-of-the-art," *Frontiers in Robotics and AI*, vol. 5, p. 59, 2018.

[43] J. Chen, M. Gauci, W. Li, A. Kolling, and R. Groß, "Occlusion-based cooperative transport with a swarm of miniature mobile robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 307–321, 2015.

[44] A. Yufka and M. Ozkan, "Formation-based control scheme for cooperative transportation by multiple mobile robots," *International Journal of Advanced Robotic Systems*, vol. 12, no. 9, p. 120, 2015.

[45] H. Ebel, W. Luo, F. Yu, Q. Tang, and P. Eberhard, "Design and experimental validation of a distributed cooperative transportation scheme," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1157–1169, 2020.

[46] K. Hunte and J. Yi, "Collaborative object manipulation through indirect control of a deformable sheet by a mobile robotic team," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1463–1468.

[47] J. Hu, W. Liu, H. Zhang, J. Yi, and Z. Xiong, "Multi-robot object transport motion planning with a deformable sheet," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9350–9357, 2022.

[48] K. Hunte and J. Yi, "Collaborative manipulation of spherical-shape objects with a deformable sheet held by a mobile robotic team," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 437–442, 2021.

[49] A. Petitti, A. Franchi, D. Di Paola, and A. Rizzo, "Decentralized motion control for cooperative manipulation with a team of networked mobile manipulators," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 441–446.

[50] Y. Ren, S. Sosnowski, and S. Hirche, "Fully distributed cooperation for networked uncertain mobile manipulators," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 984–1003, 2020.

[51] M. Wu, Y. He, and S. Liu, "Collaboration of multiple mobile manipulators with compliance based leader/follower approach," in *2016 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2016, pp. 48–53.