Grouping-Based Optimization Method for Multirobot System Pattern Formation

Tingting Wang, Fangfang Zhang, Jianbin Xin, and Yanhong Liu

***Abstract*—This article presents a novel optimization method for multirobot formation in an obstacle environment. For this challenge, we proposed an iterative optimization approach in previous work, in which all robots as a whole obtain the optimal goal pattern, and then, move to the goal without collision to form the pattern. However, this approach results in high consumption in terms of time and the path that robots travel as the number of robots increases. To improve efficiency, we propose a grouping-based optimization method. First, a specific grouping strategy (the number of groups and the number of data points in each group are fixed) is utilized to group the multiple robots. And then in an obstacle environment, each group of robots completes its optimal pattern formation in parallel without collision through coordination within and between groups. The simulation results of the multiletter pattern formation validate the effectiveness of the method proposed in this article compared to the method without grouping.**

***Index Terms*—Grouping, mixed integer convex quadratic programming, multirobot system, pattern formation.**

I. Introduction

For multirobot systems, many research problems are focused on to be addressed, such as formation [1], [2], avoidance [3], task allocation [4], etc. Among them, pattern formation is gaining increasing attention in various scenarios. In military or disaster relief scenarios, multirobots form specific patterns to complete tasks such as path searching and area coverage [5]. For entertainment purposes, multirobots equipped with colorful LEDs can present pleasing visual effects by changing patterns [6]. Furthermore, the idea of pattern formation can also be applied to chip design by modeling droplets as robots to coordinate the droplets automatically. The goal formations are determined to fit within the specified regions of the chip [7], [8].

Many articles on the pattern formation study how to effectively and stably control multiple robots to form a predefined pattern [9]–[14]. A self-assembly algorithm proposed in [9] that links three primitive collective behaviors: edge-following, gradient formation, and localization to ensure that thousands of robots form user-specified patterns. In [10], a scaled rotation gain matrix is designed to ensure that unmanned aerial vehicles (UAVs) converge to the desired pattern. The risk of collision is eliminated via the distributed task assignment of UAVs to desired formation points. The distributed self-assembly approach presented in [11] in which two motion chains are iteratively generated in parallel to fill the desired pattern. In [11], a decentralized method is proposed that the repeating pattern formation is divided into two phase. In the first phase, the multirobots are grouped into multiple basic patterns, and in the second phase, the final pattern is assembled by the basic pattern. In [12], swarm robot with highly limited cognition to form desired global patterns on a discrete grid according to the generated probabilistic local state-action graph. In [13], a pattern formation algorithm with minimum shape for nonholonomic robots by combining the idea of the virtual structure method for formation control with a behavior-based approach is proposed. In [14], the swarm robots are divided into multiple groups, and then, an adaptive mechanism based on social groups is adopted to dynamically adjust members of different groups through cooperation to form space patterns in parallel.

In practical applications, patterns often need to be optimized according to the task requirements. The limited existing methods solve the optimized pattern [15]–[19]. In [15] and [16], the coverage method is adopted to optimize the deployment of multiple robots in a given pattern area to achieve a visually appealing final pattern. In [17] and [18], the convex programming techniques are used to minimize the total travel distance for getting the optimal pattern in the obstacle-free environment. In [19], an algorithm is proposed to find the optimal goal pattern position for the UAV cluster, and then, adopted the Hungarian algorithm to allocate targets. The deep Q learning the method is used to find the optimal flight parameters of the noncollision the trajectory for the UAV to the target position under the condition of ensuring the minimum total flight distance of a single UAV so that the UAV reaches the goal point to form the pattern.

Although these works are helpful to pattern formation for multirobot systems, it is observed that the optimal pattern generation in these works does not consider the obstacle constraints, so these methods for pattern formation cannot efficiently optimize the pattern formation in the obstacles environment. In order to achieve the optimization for the pattern formation in an environment with obstacle, we propose an iterative optimization approach in previous work [20] where the optimal pattern parameters are obtained by solving the system model established in

Manuscript received 6 January 2021; revised 8 July 2021; accepted 19 October 2021. Date of publication 12 November 2021; date of current version 26 August 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 61603345, Grant 61703372, and Grant 61773351; in part by Outstanding Foreign Scientist Support Project in Henan Province under Grant GZS2019008; in part by Young Talent Lift Project in Henan Province under Grant 2020hytp006 to conduct this research investigation. (*Corresponding author: Fangfang Zhang*.)

The authors are with the School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China (e-mail: tingwang951126@163.com; zhangfangfang@zzu.edu.cn; j.xin@zzu.edu.cn; liuyh@zzu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSYST.2021.3122548, provided by the authors.

Digital Object Identifier 10.1109/JSYST.2021.3122548

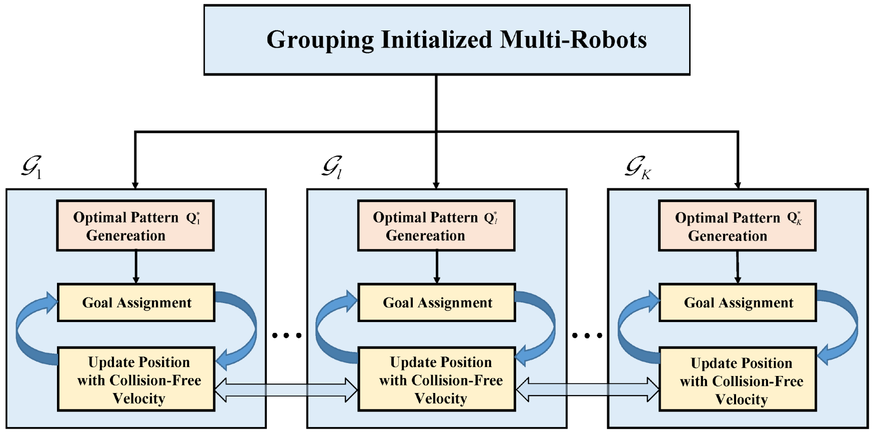

Fig. 1. Overview of the proposed method.

an obstacle environment, then an iterative controller is designed to assign the goals and plan a collision-free path for each robot to reach the goal. However, in this approach, all robots as a whole obtain the optimal goal pattern, and then, move together. It has a high cost in time and path that robots travel as the number of robots expands.

To address these limitations, this article proposes a grouping-based optimization method for multirobot pattern formation in an obstacle environment. The main contributions of this article can be summarized as follows.

1) A specific grouping method is adopted to group randomly deployed multirobots.
2) Each group of robots realizes its optimal pattern formation in parallel without collision via coordination within and between groups in an obstacle environment.
3) Simulation results in scenarios of letter pattern formation verify the effectiveness of our algorithm in improving the time efficiency and reducing the path compared with the algorithm without grouping.

The structure of this article are arranged as follows. In Section II, we outline the problem description. The specific grouping method is introduced in Section III. The generation of the optimal pattern and an iterative controller are presented in Section IV. In Section V, we demonstrate the simulation experiment and analyze the results. Finally, Section VI concludes this article.

II. Problem Description

In the plane region, given N disk robots with a radius r_i, $i \in \{1, \ldots N\}$ and M static disk obstacles with a radius r_{o_m}, $m \in \{1, \ldots M\}$, we define $\mathcal{R}_i$ as robot i, the symbol contains the attributes of the robot, such as its position, radius, and velocity. The initial position vector of the N robots is expressed as $\mathbf{P} = [\mathbf{p}_1; \ldots; \mathbf{p}_N]_{N\times 2}$, where $\mathbf{p}_i = [x_i, y_i]_{1\times 2}$ represents the initial position of $\mathcal{R}_i$. The velocity of $\mathcal{R}_i$ is $\mathbf{v}_i = [v_{ix}, v_{iy}]_{1\times 2}$. The position vector of the M static obstacles is represented by $\mathbf{O} = [\mathbf{o}_1; \ldots; \mathbf{o}_M]_{M\times 2}$, where $\mathbf{o}_m = [x_{\mathbf{o}_m}, y_{\mathbf{o}_m}]_{1\times 2}$ represents the position of obstacle m. Let there be K desired patterns. The desired pattern l is given by $\mathbf{S}_l = [\mathbf{s}_1^l; \ldots; \mathbf{s}_{N_l}^l]_{N_l\times 2}$, where $\mathbf{s}_j^l = [x_{\mathbf{s}_j}^l, y_{\mathbf{s}_j}^l]_{1\times 2}$ represents the position of point j of the desired pattern l, $l \in \{1, \ldots K\}$, $j \in \{1, \ldots N_l\}$. N_l is the number of robots required to form the desired pattern l, $\sum_{l=1}^{K} N_l = N$. The randomly initialized multirobots are grouped into K groups $\mathcal{G}=\{\mathcal{G}_l\}$, where $\mathcal{G}_l=\{\mathcal{R}_i^l\}$, $l \in \{1, \ldots, K\}$, $i \in \{1, \ldots, N_l\}$. $\mathbf{P}_l = [\mathbf{p}_1^l; \ldots; \mathbf{p}_{N_l}^l]_{N_l\times 2}$ is denoted as the initial position vector of $\mathcal{G}_l$, where $\mathbf{p}_i^l = [x_i^l, y_i^l]_{1\times 2}$ represents the initial position of $\mathcal{R}_i^l$. The optimal pattern l of the desired pattern l is expressed as $\mathbf{Q}_l^*=[\mathbf{q}_1^l; \ldots; \mathbf{q}_{N_l}^l]_{N_l\times 2}$, which is transformed from $\mathbf{S}_l$ through scale and translation, where $\mathbf{q}_j^l = [x_{\mathbf{q}_j^l}, y_{\mathbf{q}_j^l}]_{1\times 2}$ represents the position of point j of optimal pattern l, $j \in \{1, \ldots N_l\}$. For this problem, important assumptions are given as follows.

1) Failure of the robot during movement is not considered.
2) All robots can accurately obtain position and velocity information between each other.
3) The kinematics model of all robots is a holonomic model referenced from [21].

An overview of the method is given in Fig. 1. Based on the traditional K-means method [22], given the number of groups and the number of samples in each group, we present a specific clustering method to group randomly initialized multirobots into K groups $\mathcal{G}=\{\mathcal{G}_l\}$, $l \in \{1, \ldots, K\}$. Then, $\mathcal{G}_l$ completes its optimal pattern formation in parallel. In the parallel optimal pattern formation process, first, $\mathcal{G}_l$ obtains the optimal pattern $\mathbf{Q}_l^*$ via the optimal pattern generation model established in an obstacle environment; second, an iterative controller obtains the optimal assignment between $\mathcal{G}_l$ and $\mathbf{Q}_l^*$, and then, $\mathcal{R}_i^l$ in $\mathcal{G}_l$ reaches the assigned goal without collision through coordination within and between groups, after which the pattern l is formed. Once all patterns are formed, the whole ends.

Next, we introduce the implementation of the proposed algorithm in detail.

III. Grouping Initialized Multirobots

This section presents the grouping strategy developed from the traditional K-means method [22]. This strategy sets the fixed number of groups and the fixed size of each group to group the randomly initialized multirobots.

A. Traditional K-Means Method

The K-means algorithm clusters the point set $X=\{\chi_i\}$, $i \in \{1,\ldots,N\}$, into K clusters $\mathcal{C}=\{\mathcal{C}_j\}$, $j \in \{1,\ldots,K\}$. Let μ_j be the centroid of the cluster $\mathcal{C}_j$. The squared error between μ_j and the point χ_i in the cluster $\mathcal{C}_j$ is defined as follows:

$$J\left(\mathcal{C}_j\right) = \sum_{\chi_i \in \mathcal{C}_j} \left\|\chi_i - \mu_j\right\|^2 \tag{1}$$

where $\mu_j = \frac{1}{|\mathcal{C}_j|}\sum_{\chi_i \in \mathcal{C}_j} \chi_i$, and $|\mathcal{C}_j|$ is the number of points in $\mathcal{C}_j$.

The goal of K-means is to minimize the sum of the squared error of all K clusters

$$\min J\left(\mathcal{C}\right) = \sum_{j=1}^{K}\sum_{\chi_i \in \mathcal{C}_j} \left\|\chi_i - \mu_j\right\|^2. \tag{2}$$

The main steps of the K-means algorithm are as follows [23].

1) Choose K points as scattered as possible as the initial centroids of clusters.
2) Assign the remaining points to the nearest centroids, forming K clusters.
3) Recalculate the centroid of each cluster to determine whether the new centroids are the same as the old centroids. If the same, the clustering is complete; otherwise, update the centroids and perform step 2).

The traditional K-means method does not specify the number of points contained in each cluster. In this article, the number of robots contained in each group is fixed according to the number of robots required to form each desired pattern.

B. Proposed Grouping Method

With reference to the traditional K-means method, we present the developed clustering method based on the initial position vector $\mathbf{P} = [\mathbf{p}_1;\ldots;\mathbf{p}_N]_{N\times 2}$, in which the multirobots are clustered into K groups $\mathcal{G}=\{\mathcal{G}_j\}$, $j \in \{1,\ldots,K\}$, where K is the number of desired patterns. The number N_j of robots required to form each desired pattern is used to constrain the size of each group. We update the group centroid iteratively until the centroid converges, after which the grouping is complete. This method consists of three steps, where steps 1) and 3) are the same as those in the traditional method. In step 1), we choose the K dispersed positions of the robots as the initial centroids of the groups. Next, we detail step 2), i.e., assigning the robots to the centroids.

Considering the fixed size of the group, we minimize the sum of the squared distance between the robots and the centroids to obtain the optimal allocation between the robots and the centroids. The assignment model is as follows:

$$\min Z\left(\mathcal{G},\boldsymbol{\sigma}\right) = \sum_{i=1}^{N}\sum_{j=1}^{K} z_{ij}x_{ij} \tag{3}$$

Subject to

$$\begin{cases} \sum_{i=1}^{N} x_{ij} = N_j, & j = 1,\ldots,K \quad (4a) \\ \sum_{j=1}^{K} x_{ij} = 1, & i = 1,\ldots,N \quad (4b) \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \mathcal{R}_i \text{ assigned to } \boldsymbol{\mu}_j \\ 0, & \mathcal{R}_i \text{ not assigned to } \boldsymbol{\mu}_j \end{cases} \tag{5}$$

where $z_{ij}=\|\mathbf{p}_i-\boldsymbol{\mu}_j\|^2$ is the squared distance between the position $\mathbf{p}_i$ of $\mathcal{R}_i$ and the centroid position $\boldsymbol{\mu}_j$ of the group $\mathcal{G}_j$, and $\boldsymbol{\mu}_j = \frac{1}{|\mathcal{G}_j|}\sum_{\mathcal{R}_i \in \mathcal{G}_j} \mathbf{p}_i$, $j \in \{1,\ldots,K\}$. $\boldsymbol{\sigma}(x_{ij})_{N\times K}$ is the assignment between the robots and the centroids. Equation (4 a) means that group $\mathcal{G}_j$ contains N_j robots. Equation (4 b) expresses that a robot can be assigned to only one group.

Solving the function (3) satisfying the constraints (4) and (5) is the 0–1 integer programming problem [24]. The problem can be solved utilizing the CPLEX optimization software [25] to obtain the optimal assignment.

In step 3), we recalculate the centroid of each group to determine whether the centroids have changed; if not, the grouping is complete. Otherwise, we update the centroids and perform step 2).

As with the traditional clustering methods, different initial centroids lead to different clusterings because the K-means method converges to the local minima. We select centroids that are as dispersed as possible; this process is detailed in [23]. Fig. 2 shows an illustration of the specific grouping algorithm for three groups of fixed size.

IV. Optimal Pattern Formation

After completing the grouping, each group completes its optimal pattern formation in parallel without collision through coordination within and between groups. Without loss of generality, we present in detail the process undertaken by a group of robots $\mathcal{G}_l=\{\mathcal{R}_i^l\}$ in an obstacle environment to achieve the optimal pattern formation. The process consists of two phases [20]: the optimal pattern $\mathbf{Q}_l^*$ generation and the collision-free path planning using an iterative controller.

A. Optimal Pattern $\mathbf{Q}_l^*$ Generation

In an obstacle environment, $\mathcal{G}_l$ minimizes the objective function (the sum of the squared distance between the robot positions and the goal positions) to obtain the optimal assignment $\boldsymbol{\sigma}_l^*$ between the robots and the pattern goals, the optimal scale and the translation parameters α_l^*, $\mathbf{d}_l^* = [d_1^l, d_2^l]_{1\times 2}^*$. Then, the optimal pattern $\mathbf{Q}_l^*=\alpha_l^*\mathbf{S}_l + \mathbf{d}_l^*$.

The objective function of α_l, $\mathbf{d}_l$, and $\boldsymbol{\sigma}_l$ is given as follows:

$$\min H\left(\alpha_l,\mathbf{d}_l,\boldsymbol{\sigma}_l\right) = \sum_{i=1}^{N_l}\sum_{j=1}^{N_l} h_{ij}x_{ij} \tag{6}$$

subject to

$$\begin{cases} \sum\limits_{i=1}^{N_l} x_{ij} = 1, & j = 1,\ldots,N_l \quad (7a) \\ \sum\limits_{j=1}^{N_l} x_{ij} = 1, & i = 1,\ldots,N_l \quad (7b) \end{cases}$$

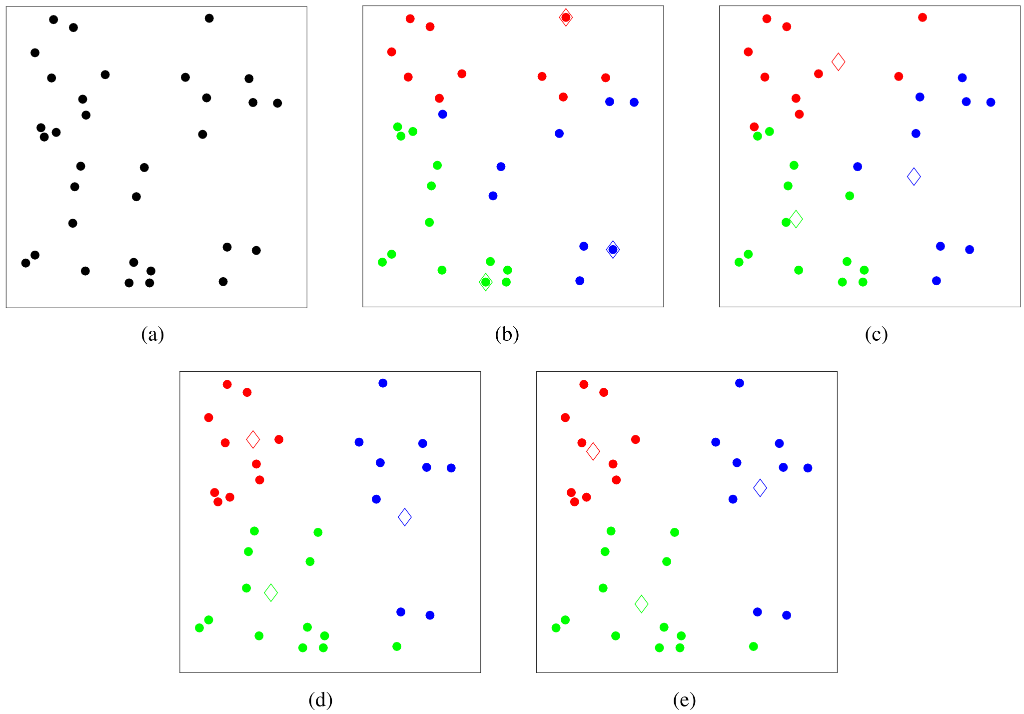

Fig. 2. Illustration of the specific grouping algorithm. The groups labeled red, green, and blue have fixed sizes of 10,13, and 9, respectively. (a) Initial positions of multiple robots as the input. (b) Three positions selected as initial group centroid and initial assignment of the positions to groups. (c) and (d) Intermediate iterations updating group labels and their centroids. (e) Final grouping obtained by the specific grouping algorithm at convergence.

$$x_{ij} = \begin{cases} 1, & \mathcal{R}_i^l \text{ assigned to } \mathbf{q}_j^l \\ 0, & \mathcal{R}_i^l \text{ not assigned to } \mathbf{q}_j^l \end{cases} \tag{8}$$

where $h_{ij} = \|\mathbf{p}_i^l - \mathbf{q}_j^l\|^2$ is the squared distance between the initial position $\mathbf{p}_i^l$ of $\mathcal{R}_i^l$ and the position $\mathbf{q}_j^l$ of the point j of the optimal pattern l. $\boldsymbol{\sigma}_l = (x_{ij})_{N_l \times N_l}$ is the assignment matrix between the optimal pattern goal and the robot. Formulas (7 a) and (7 b), respectively, indicate that a goal can be assigned to only one robot and that a robot can be assigned to only one goal. The optimal pattern $\mathbf{Q}_l^*$ is obtained from the desired pattern $\mathbf{S}_l$ by scale and translation.

$$\mathbf{q}_j^l = \alpha_l \mathbf{s}_j^l + \mathbf{d}_l \Rightarrow$$
$$\begin{cases} x_{\mathbf{q}_j^l} = \alpha_l y_{\mathbf{s}_j^l} + d_1^l \\ y_{\mathbf{q}_j^l} = \alpha_l y_{\mathbf{s}_j^l} + d_2^l \end{cases} \tag{9}$$

where $\mathbf{q}_j^l = [x_{\mathbf{q}_j^l}, y_{\mathbf{q}_j^l}]_{1\times 2}$ is the position of the point j of the optimal pattern l and $\mathbf{s}_j^l = [x^l_{\mathbf{s}_j}, y^l_{\mathbf{s}_j}]_{1\times 2}$ represents the position of the point j in the desired pattern l, $j \in \{1, \ldots N_l\}$.

$$\begin{aligned} h_{ij} &= \left(x_i^l\right)^2 + \left(y_i^l\right)^2 + \alpha_l^2 \left(\left(x^l_{s_j}\right)^2 + \left(y^l_{s_j}\right)^2\right) \\ &\quad - 2\alpha_l \left(x_i^l x^l_{s_j} + y_i^l y^l_{s_j}\right) + 2\alpha_l \left(x^l_{s_j} d_1^l + y^l_{s_j} d_2^l\right) \\ &\quad - 2\left(x_i^l d_1^l + y_i^l d_2^l\right) + \left(d_1^l\right)^2 + \left(d_2^l\right)^2 \\ &= \alpha_l^2 \mathbf{s}_j^l \left(\mathbf{s}_j^l\right)^\top - 2\alpha_l \mathbf{p}_i^l \left(\mathbf{s}_j^l\right)^\top + 2\alpha_l \mathbf{s}_j^l \mathbf{d}_l^\top \\ &\quad - 2\mathbf{p}_i^l \mathbf{d}_l^\top + \mathbf{d}_l \mathbf{d}_l^\top + \mathbf{p}_i^l \left(\mathbf{p}_i^l\right)^\top. \end{aligned} \tag{10}$$

Substituting formula (10) into $H(\alpha_l, \mathbf{d}_l, \sigma_l)$, we obtain

$$\begin{aligned} H(\alpha_l, \mathbf{d}_l, \sigma_l) &= \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \Big(\alpha_l^2 \mathbf{s}_j^l \left(\mathbf{s}_j^l\right)^\top x_{ij} - 2\alpha_l \mathbf{p}_i^l \left(\mathbf{s}_j^l\right)^\top x_{ij} \\ &\quad + 2\alpha_l \mathbf{s}_j^l \mathbf{d}_l^\top x_{ij} - 2\mathbf{p}_i^l \mathbf{d}_l^\top x_{ij} + \mathbf{d}_l \mathbf{d}_l^\top x_{ij} + \mathbf{p}_i^l \left(\mathbf{p}_i^l\right)^\top x_{ij}\Big) \\ &= \alpha_l^2 \sum_{i=1}^{N_l} \mathbf{s}_j^l \left(\mathbf{s}_j^l\right)^\top - 2\alpha_l \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \mathbf{p}_i^l \left(\mathbf{s}_j^l\right)^\top x_{ij} \\ &\quad + 2\alpha_l \sum_{i=1}^{N_l} \mathbf{s}_j^l \mathbf{d}_l^\top - 2 \sum_{i=1}^{N_l} \mathbf{p}_i^l \mathbf{d}_l^\top \\ &\quad + \sum_{i=1}^{N_l} \mathbf{d}_l \mathbf{d}_l^\top + \sum_{i=1}^{N_l} \mathbf{p}_i^l \left(\mathbf{p}_i^l\right)^\top \\ &= A\alpha_l^2 - 2\alpha_l \sum_{i=1}^{N_l} \sum_{j=1}^{N_l} \mathbf{p}_i^l \left(\mathbf{s}_j^l\right)^\top x_{ij} + 2\alpha_l \mathbf{s}_l \mathbf{d}_l^\top \\ &\quad - 2\mathbf{p}_l \mathbf{d}_l^\top + N_l \mathbf{d}_l \mathbf{d}_l^\top + B \end{aligned} \tag{11}$$

where $A = \sum_{j=1}^{N_l} \mathbf{s}_j^l (\mathbf{s}_j^l)^\top$, $B = \sum_{i=1}^{N_l} \mathbf{p}_i^l (\mathbf{p}_i^l)^\top$, $\mathbf{s}_l = \sum_{j=1}^{N_l} \mathbf{s}_j^l$, and $\mathbf{p}_l = \sum_{i=1}^{N_l} \mathbf{p}_i^l$ are independent of α_l, $\mathbf{d}_l$, and $\boldsymbol{\sigma}_l$.

If the assignment $\boldsymbol{\sigma}_l^*(x_{ij}^*)$ is optimal at some values of $\alpha_l \in (0, \infty)$, $\mathbf{d}_l$, then it is also optimal at any $\alpha_l \in (0, \infty)$, $\mathbf{d}_l$ [17], [20]. Thus, from formula (11), let a pseudocost $k_{ij} = -\mathbf{p}_i^l (\mathbf{s}_j^l)^\top$;

then, the following assignment model is established:

$$\min K = \sum_{i=1}^{N_l}\sum_{j=1}^{N_l} k_{ij}x_{ij} = \sum_{i=1}^{N_l}\sum_{j=1}^{N_l}\left(-\mathbf{p}_i^l\left(\mathbf{s}_j^l\right)^\top\right)x_{ij} \tag{12}$$

subject to the constraints (7) and (8).

We can solve this assignment model to obtain the optimal assignment $\boldsymbol{\sigma}_l^*(x_{ij}^*)$ and the minimum function value K^* by adopting the auction algorithm [26].

Based on $\boldsymbol{\sigma}_l^*(x_{ij}^*)$, formula (11) can be written as

$$H\left(\alpha_l, \mathbf{d}_l\right) = A\alpha_l^2 + 2\alpha_l K^* + 2\alpha_l \mathbf{s}_l \mathbf{d}_l^\top - 2\mathbf{p}_l\mathbf{d}_l^\top + N_l\mathbf{d}_l\mathbf{d}_l^\top + B. \tag{13}$$

The function (13) is a convex quadratic function of α_l, $\mathbf{d}_l$ [17], [20], which serves as the final objective function needed to minimize for obtaining the optimal pattern parameters α_l^*, $\mathbf{d}_l^*$.

Considering that the optimal pattern generation model is built in the obstacle environment, the final objective function (13) needs to meet the following constraints.

1) The obstacle constraints should be considered to ensure that the optimal pattern $\mathbf{Q}_l^*$ is not generated in static obstacles; thus, the following conditions must be met:

$$\left\|\mathbf{q}_j^l - \mathbf{o}_i\right\| \geq r_j + r_{o_i} \tag{14}$$

where $\mathbf{o}_i = [x_{\mathbf{o}_i}, y_{\mathbf{o}_i}]$ is the position of the static obstacle i, $i \in \{1, \ldots, M\}$; r_j is the robot radius; and r_{o_i} is the static obstacle radius.

According to (9), the constraint (14) can be further written as follows:

$$\sqrt{\left[\left(\alpha_l x_{\mathbf{s}_j}^l + d_1^l\right) - x_{\mathbf{o}_i}\right]^2 + \left[\left(\alpha_l y_{\mathbf{s}_j}^l + d_2^l\right) - y_{\mathbf{o}_i}\right]^2} \geq r_j + r_{o_i}. \tag{15}$$

The aforementioned equation is a nonconvex constraint. Minimizing the objective function (13) under the constraint (15) is a nonconvex optimization problem. Nonconvex optimization problems can have a global optimal solution. However, it can have multiple local optimal solutions, making the problem difficult to solve. Therefore, the binary variables are introduced to approximate the constraint (15) into a linear constraint [20], [27], and then, transform the nonconvex optimization problem into a simple mixed integer convex quadratic programming problem [28], [29] with a global optimal solution

$$\begin{cases} \left|\alpha_l x_{\mathbf{s}_j^l} + d_1^l - x_{\mathbf{o}_i}\right| \geq r_j + r_{o_i} \\ \left|\alpha_l y_{\mathbf{s}_j^l} + d_2^l - y_{\mathbf{o}_i}\right| \geq r_j + r_{o_i} \end{cases}$$
$$\Rightarrow \begin{cases} \alpha_l x_{\mathbf{s}_j^l} + d_1^l - x_{\mathbf{o}_i} \geq r_j + r_{o_i} - FW_{ij}(1) \\ x_{\mathbf{o}_i} - \left(\alpha_l x_{\mathbf{s}_j^l} + d_1^l\right) \geq r_j + r_{o_i} - FW_{ij}(2) \\ \alpha_l y_{\mathbf{s}_j^l} + d_2^l - y_{\mathbf{o}_i} \geq r_j + r_{o_i} - FW_{ij}(3) \\ y_{\mathbf{o}_i} - \left(\alpha_l y_{\mathbf{s}_j^l} + d_2^l\right) \geq r_j + r_{o_i} - FW_{ij}(4) \\ \sum_{\omega=1}^{4} W_{ij}(\omega) \leq 3 \end{cases} \tag{16}$$

where $W_{ij}(\omega)$ is the introduced binary variable and has a value of 1 or 0, $\omega \in \{1, 2, 3, 4\}$, and F is a positive value and is much larger than $r_j + r_{o_i}$.

2) The following separation conditions between the optimal pattern goals should be satisfied to ensure that the robot can finally reach the assigned goals without overlapping

$$\left\|\mathbf{q}_i^l - \mathbf{q}_j^l\right\| \geq r_i + r_j \Rightarrow$$
$$\left[\left(\alpha_l x_{\mathbf{s}_i^l} - \alpha_l x_{\mathbf{s}_j^l}\right)^2 + \left(\alpha_l y_{\mathbf{s}_i^l} - \alpha_l y_{\mathbf{s}_j^l}\right)^2\right] \geq (r_i + r_j)^2$$
$$\Rightarrow \alpha_l \min\left\|\mathbf{s}_i^l - \mathbf{s}_j^l\right\| \geq r_i + r_j \tag{17}$$

where $i \neq j; i, j \in \{1, \ldots, N_l\}$.

3) In general, practical applications require the pattern goals to not lie outside the workspace. The following constraints can be specified:

$$\begin{cases} \alpha_l \max\left(x_{\mathbf{s}_j^l}\right) + d_1^l \leq X_{\max} - r_j \\ \alpha_l \min\left(x_{\mathbf{s}_j^l}\right) + d_1^l \geq X_{\min} - r_j \\ \alpha_l \max\left(y_{\mathbf{s}_j^l}\right) + d_2^l \leq Y_{\max} - r_j \\ \alpha_l \min\left(y_{\mathbf{s}_j^l}\right) + d_2^l \geq Y_{\min} - r_j \end{cases} \tag{18}$$

where $X_{\max}$, $X_{\min}$, $Y_{\max}$, and $Y_{\min}$ are the maximum and minimum boundaries of the entire application area along the X- and Y-axes, respectively.

Under the affine constraints (16)–(18), minimizing the objective convex quadratic function (13) is a mixed integer convex quadratic programming problem [28], [29]. The problem can be solved utilizing the CPLEX optimization software to obtain the global optimal pattern parameters α_l^*, $\mathbf{d}_l^*$. Then, the optimal pattern $\mathbf{Q}_l^* = \alpha_l^*\mathbf{S}_l + \mathbf{d}_l^*$ is obtained. The pseudocode is provided in Algorithm 1.

B. Collision-Free Path Planning Using an Iterative Controller

In the kth iteration ($k \in \mathbb{N}^+$, the time-step index of the controller), $\mathcal{G}_l(k)$ and $\mathbf{Q}_l^*$ are assigned first, and then, through coordination with the robots in $\mathcal{G}_l(k)$, and the robots in other groups, the robot $\mathcal{R}_i^l(k)$ obtains the collision-free velocity and updates its position. Repeat the iterative process until all $\mathcal{R}_i^l(k)$ reach the assigned goals, and the pattern l is formed.

1) Goal Assignment: Although the allocation $\boldsymbol{\sigma}_l^*$ between $\mathcal{G}_l$ and $\mathbf{Q}_l^*$ at the initial moment was achieved in Section IV-A, $\mathcal{R}_i^l(k)$ needs to avoid collisions during its movement, which

Algorithm 1 Generation of Optimal Pattern $\mathbf{Q}_l^*$

Input:
Output: $\mathbf{Q}_l^*$
Based on the function (12) and constraints (7) and (8), $\boldsymbol{\sigma}_l^*$ and K^* are calculated.
Based on the function (13) and constraints (16), (17), and (18), α_l^* and $\mathbf{d}_l^*$ are computed.
$\mathbf{Q}_l^* = \alpha_l^* \mathbf{S}_l + \mathbf{d}_l^*$
return $\mathbf{Q}_l^*$

will cause $\mathcal{R}_i^l(k)$ to move away from the previously assigned goal [20]. Thus, it is necessary to reassign $\mathcal{G}_l(k)$ and $\mathbf{Q}_l^*$ at the beginning of each iteration. We assign $-\mathbf{p}_i^l(k-1)(\mathbf{q}_j^l)^T$ to k_{ij} in the function (12) and adopt the auction algorithm to solve the assignment model; then, we obtain the allocation $\boldsymbol{\sigma}_l^*(k)$ between $\mathcal{G}_l(k)$ and $\mathbf{Q}_l^*$.

2) Update Position With the Collision-Free Velocity: Based on $\boldsymbol{\sigma}_l^*(k)$, we compute the preferred velocity $\mathbf{v}_{\text{pref}_i}^l(k)$ of $\mathcal{R}_i^l(k)$ without considering the obstacles and other robots [15]:

$$\mathbf{v}_{\text{pref}_i}^l(k) = V_p \min\left(1, \frac{\left\|\mathbf{q}_{\boldsymbol{\sigma}_l^*(k,i)}^l - \mathbf{p}_i^l(k-1)\right\|}{K_c}\right)\mathbf{v}_{\mathbf{pq}_i}^l(k)$$
$$\mathbf{v}_{\mathbf{pq}_i}^l(k) = \frac{\mathbf{q}_{\boldsymbol{\sigma}_l^*(k,i)}^l - \mathbf{p}_i^l(k-1)}{\left\|\mathbf{q}_{\boldsymbol{\sigma}_l^*(k,i)}^l - \mathbf{p}_i^l(k-1)\right\|} \tag{19}$$

where the constant $V_p > 0$ is the maximum speed of the robot; $\mathbf{q}_{\boldsymbol{\sigma}_l^*(k,i)}^l$ is the optimal pattern goal assigned to $\mathcal{R}_i^l(k)$ in the kth iteration; and $\mathbf{p}_i^l(k-1)$ is the position of $\mathcal{R}_i^l(k-1)$. The constant $K_c \geq V_p\tau$ ensures convergence, and τ is the time step.

In multirobot and dynamic environments, $\mathcal{R}_i^l$ does not need to consider other far away robots and obstacles when avoiding collisions. $\mathcal{R}_i^l$ needs to consider only other $\mathcal{R}_j$, $j \neq i$, $j \in \{1, \ldots, N\}$, and obstacles in the neighbor region NR_i^l [21]. K groups of pattern formation are completed in one area. Thus, $\mathcal{R}_i^l$ should consider the robots in $\mathcal{G}_l$ and robots in other groups.

For $\mathcal{R}_i^l(k)$, based on the velocities and positions of the other robots and obstacles in the NR_i^l, the velocity obstacle space $\text{RVO}_i^l(k)$ is obtained via the reciprocal velocity obstacles (RVO) algorithm [21]. $\mathcal{R}_i^l(k)$ selects the velocity outside the $\text{RVO}_i^l(k)$ and closest to $\mathbf{v}_{\text{pref}_i}^l(k)$ as the collision-free velocity $\mathbf{v}_{\text{cf}_i}^l(k)$.

$$\text{RVO}_i^l(k) = \bigcup_{j\in\text{NR}_i^l} \text{RVO}_{ij}^l\left(\mathbf{v}_j(k-1), \mathbf{v}_i^l(k-1)\right) \cup \bigcup_{\mathbf{o}\in\mathbf{O}} \text{VO}_i^l(\mathbf{v_o}) \tag{20}$$

where $\text{RVO}_{ij}^l(\mathbf{v}_j(k-1), \mathbf{v}_i^l(k-1))$ and $\text{VO}_i^l(\mathbf{v_o})$ are the velocity obstacle spaces of $\mathcal{R}_i^l(k)$ relative to $\mathcal{R}_j(k)$ and static obstacles in the NR_i^l. $\text{RVO}_i^l(k)$ is the union of these velocity obstacle spaces. $\mathbf{v}_i^l(k-1)$ and $\mathbf{v}_j(k-1)$, respectively, represent the velocity of $\mathcal{R}_i^l(k-1)$ and that of its neighbors. $\mathbf{v_o} = 0$ represents the velocity of the static obstacle.

Algorithm 2 Collision-free Path Planning Using An Iterative Controller

Input: $\mathbf{Q}_l^*$
Output: $\mathbf{p}_i^l(k)$
repeat
Based on the function (12) with $k_{ij} = -\mathbf{p}_i^l(k-1)(\mathbf{q}_j^l)^\top$ and constraints (7) and (8), $\boldsymbol{\sigma}_l^*(k)$ is obtained.
Based on the formula (19), $\mathbf{v}_{\text{pref}_i}^l(k)$ is computed.
Based on the formula (23), $\mathbf{v}_{\text{cf}_i}^l(k)$ is computed.
Based on the formula (24), update $\mathbf{p}_i^l(k)$.
return $\mathbf{p}_i^l(k)$
until All $\mathcal{R}_i^l$ reach the assigned goals

The robot $\mathcal{R}_i^l$ in this article is the holonomic robot that performs the continuous cycle of sensing and acting [20], [21]. We give a 2-D kinematic model as follows:

$$\begin{cases} x_i^l(k) = x_i^l(k-1) + v_{ix}^l(k)\tau \\ y_i^l(k) = y_i^l(k-1) + v_{iy}^l(k)\tau \\ \mathbf{v}_i^l(k) = \mathbf{v}_i^l(k-1) + a_i^l(k)\tau \end{cases} \tag{21}$$

where x_i^l and y_i^l are the plane position coordinates of the center of the disk robot $\mathcal{R}_i^l$; v_{ix}^l and v_{iy}^l are the velocity components; a_i^l is the acceleration; and τ is the time step.

Each robot is subject to kinematic constraints. These constraints limit the choice of feasible velocities [21]. We set the maximum speed of the robot not to exceed V_p and the maximum acceleration not to exceed a, then the feasible velocities set is AV_i^l:

$$AV_i^l = \left\{(\mathbf{v}_i^l)' \mid \left\|(\mathbf{v}_i^l)'\right\| < V_p \wedge \left\|(\mathbf{v}_i^l)' - \mathbf{v}_i^l\right\| < a\tau\right\}. \tag{22}$$

In the kth iteration, the robot $\mathcal{R}_i^l$ selects the feasible velocity outside the $\text{RVO}_i^l(k)$ and closest to $\mathbf{v}_{\text{pref}_i}^l(k)$ as the optimal obstacle avoidance velocity $\mathbf{v}_{\text{cf}_i}^l(k)$ is

$$\mathbf{v}_{\text{cf}_i}^l(k) = \underset{\mathbf{v}_i^l \notin \text{RVO}_i^l(k), \mathbf{v}_i^l \in AV_i^l}{\arg\min} \left\|\mathbf{v}_i^l - \mathbf{v}_{\text{pref}_i}^l(k)\right\| \tag{23}$$

where AV_i^l is the velocity set for which its maximum speed does not exceed VP.

$\mathcal{R}_i^l(k)$ updates the position according to the following formula:

$$\mathbf{p}_i^l(k) = \mathbf{p}_i^l(k-1) + \mathbf{v}_{cf_i}^l(k)\tau. \tag{24}$$

Repeat the iterative process until all $\mathcal{R}_i^l$ reach the assigned goals, and pattern l is formed.

The convergence guarantee for the robot reaching the assigned goal can be found in [20]. The pseudocode is provided in Algorithm 2.

The grouping-based optimization algorithm for multirobot pattern formation mainly includes grouping and optimal pattern formation. First, multiple robots are clustered into K groups by minimizing the sum of square errors of K groups. After that, each $\mathcal{G}_l$ performs the process of the optimal pattern formation in parallel. The proposed algorithm for multirobot pattern formation ends when all groups complete their optimal pattern formation. In the following, we measure the computational complexity of the proposed algorithm.

C. Measuring Computational Complexity

The grouping-based optimization algorithm for multirobot pattern formation mainly includes grouping and optimal pattern formation. The grouping strategy refers to the K-MEANS algorithm, and its computational complexity is $\mathrm{O}(N)$. After the grouping is complete, each $\mathcal{G}_l$ achieves its optimal pattern formation in parallel. The phase of the optimal pattern formation includes Algorithms 1 and 2.

In the Algorithm 1, the optimal assignment $\boldsymbol{\sigma}_l^*(x_{ij}^*)$ and the minimum function value K^* can be obtained in $\mathrm{O}(N_l^2 \log(\max(k_{ij})N_l))$ using the auction algorithm [26]. The optimal pattern parameters α_l^*, $\mathbf{d}_l^*$ can be computed in $\mathrm{O}(N_l)$ [17]. Therefore, the computational complexity of the Algorithm 1 is $\mathrm{O}(N_l^2 \log(\max(k_{ij})N_l))$. The main computational complexity of the Algorithm 2 comes from the calculation of the optimal assignment $\boldsymbol{\sigma}_l^*(k)$ at the beginning of the iteration; thus, the computational complexity of the Algorithm 2 is $\mathrm{O}(K_l N_l^2 \log(\max(k_{ij})N_l))$, where K_l is the number of iterations to complete the pattern l formation. Considering the computational complexity of Algorithms 1 and 2, the computational complexity of the optimal pattern l formation is $\mathrm{O}((K_l+1)N_l^2 \log(\max(k_{ij})N_l))$. The algorithm's computational complexity that all groups complete their optimal pattern formation depends on the final group that completes the optimal pattern formation. Therefore, the computational complexity of the optimal pattern formation is $\mathrm{O}((K_f+1)N_f^2 \log(\max(k_{ij})N_f))$, where K_f and N_f are, respectively, the number of iterations to complete the final pattern formation and the number of robots forming the final pattern.

In contrast, the computational complexity of grouping is less than the computational complexity of optimal pattern formation. Therefore, the computational complexity of the proposed algorithm is $\mathrm{O}((K_f+1)N_f^2 \log(\max(k_{ij})N_f))$.

V. Simulation and Analysis

We design several experiments of the multiletters pattern formation using MATLAB 2016a with the CPLEX solver on a computer (Windows 10, Intel Core i7-6700, CPU at 3.40 GHz with 16.0 GB of RAM) to verify that the proposed algorithm can effectively achieve optimization for the multirobot pattern formation in an obstacle environment. First, we show the process of multiletters pattern formation in the plane obstacle area to demonstrate the feasibility of the proposed algorithm. Second, we investigate the performance by comparing the runtime T (the time from grouping to form all patterns) and the total travel distance L (the total path sum of all robots to reach the goal) of our grouping-based algorithm with those of the algorithm without grouping. Finally, the time T_s for obtaining the optimal pattern $\mathbf{Q}^*$ is compared between our proposed algorithm and the algorithm in [17].

TABLE I
Simulation Results Under The Algorithm Without Grouping And Grouping-Based Algorithm Over 30 Monte Carlo Trials

	Runtime T [s]		Total travel distance L [unit]	
	without grouping	grouping-based	without grouping	grouping-based
Mean	132.45	**35.11**	543.35	**326.62**
Std	13.16	**3.90**	62.2465	**37.46**

A. Process of Multiletters Pattern Formation in an Obstacle Environment

A trial consists of randomly initializing 32 disk robots with a radius of 1 unit and 2 static disk obstacles with a radius of five units in a 45 × 45 unit area. The desired patterns are the multiletters ZZU. Fig. 3 shows snapshots of the multiletters pattern formation, where the black disk and the gray disk represent the robot and the static obstacle, respectively. Fig. 3(a) and (b) displays the grouping result in which 32 robots are clustered into three groups and each group contains 11, 11, and 10 robots. Fig. 3(c) presents the optimal pattern generation, where the points represent the optimal pattern goals, which are the same color as the corresponding group. Fig. 3(d) and (e) illustrates the iterative collision-free path planning. Fig. 3(f) shows the final multiletters ZZU pattern formation. The supplemental video contains animated simulations of the trial.

Fig. 4 shows the position error of each robot relative to the assigned goal, which indicates that each robot can reach the assigned goal within a small error range. The significant reduction in the position error of a robot in the figure occurs because, at the beginning of the collision-free path planning, the reassigned goal is closer to the robot than the previously assigned goal. When the environment is not crowded, the assigned goals change less frequently. In Fig. 5, the convergence of the position error sum of each group of robots is shown. The trial demonstrates the feasibility and convergence of the proposed algorithm.

B. Performance Comparison Between the Grouping-Based Algorithm and Algorithm Without Grouping

We perform 30 Monte Carlo trials to compare the performances of our algorithm in this article and the algorithm without grouping [20]. Each trial consists of randomly initializing 32 disk robots with a radius of one unit in a 45 × 45 unit area with two static disk obstacles with a radius of five units. The desired patterns are the multiletters ZZU. We record the T and L of each trial under our algorithm and the algorithm in [20]. The mean and standard deviation of the trial results are reported in Table I. From the experimental results, it can be seen that in terms of the runtime T, our algorithm for forming patterns in parallel is superior to the algorithm without grouping, reducing the mean by 73.49%. In terms of the total distance L, our algorithm is also better than the algorithm without grouping, reducing the mean by 39.89%.

We attribute this to the fact that each group independently solves its own optimal pattern $\mathbf{Q}^*$ in our algorithm. The algorithm without grouping takes multiple desired patterns as a

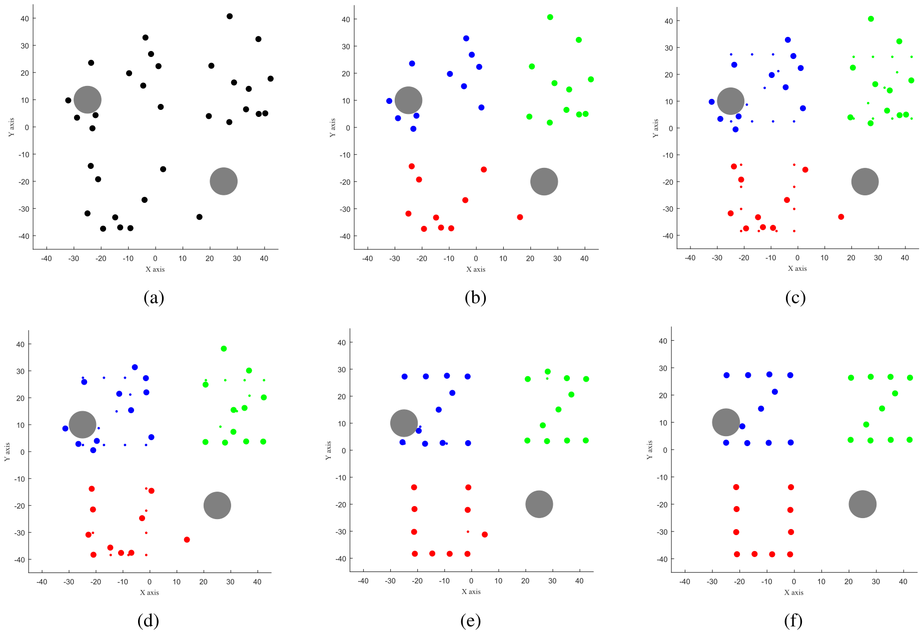

Fig. 3. Snapshots of multiletters ZZU formation.

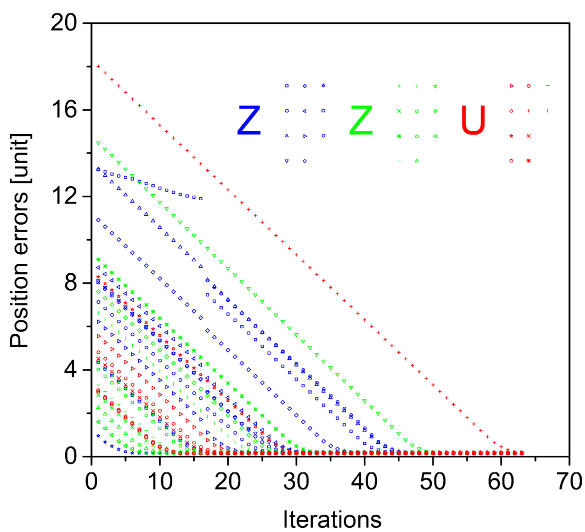

Fig. 4. Scatter plot of position error change of 32 robots in the process of optimal pattern formation.

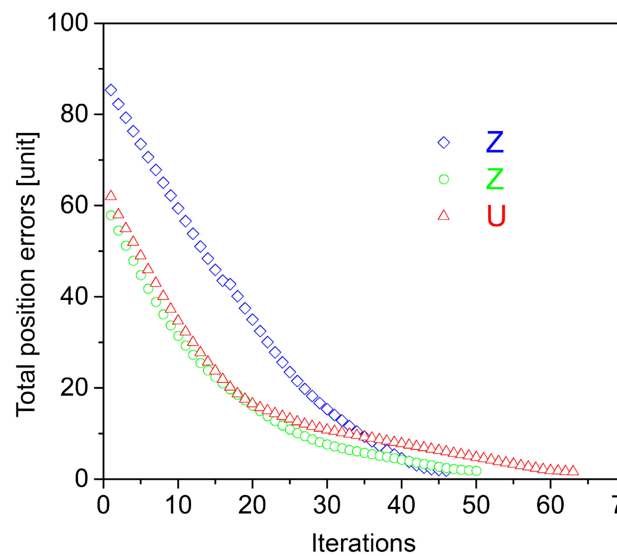

Fig. 5. Scatter plot about the sum of the position errors of each group of robots in the process of optimal pattern formation.

whole desired pattern $\mathbf{S} = \{\mathbf{S}_1, \ldots, \mathbf{S}_K\}$. Solving the whole optimal pattern $\mathbf{Q}^*$ also needs to meet the constraints of the relative positions of multiple desired patterns in the whole desired pattern. Then, the constraint increases and the feasible region decreases, which leads to poor solution quality, that is, L increases.

C. Time Comparison for Obtaining $\mathbf{Q}^*$

We perform 30 repeated trials based on the same experimental setup as in [17], where 600 robots form the multiletters UNCC. In Fig. 6, we report the distribution in terms of the solution time T_s for the 30 trials, which displays the variability in the solution time T_s over 30 repeated trials for identical initial robot positions. We attribute this to the local convergence of the grouping algorithm, which leads to different grouping results, and then, affects the solution of the optimal pattern $\mathbf{Q}^*$. As shown in Fig. 6, the solution time of our algorithm is shorter than that of the algorithm in [17] where the solution time is 45 s. We calculate T_s's mean and standard deviation of T_s over the 30 trials, which are 22.1405 and 5.9854 s, respectively. The mean solution time is reduced by 50.8%, compared with that in [17].

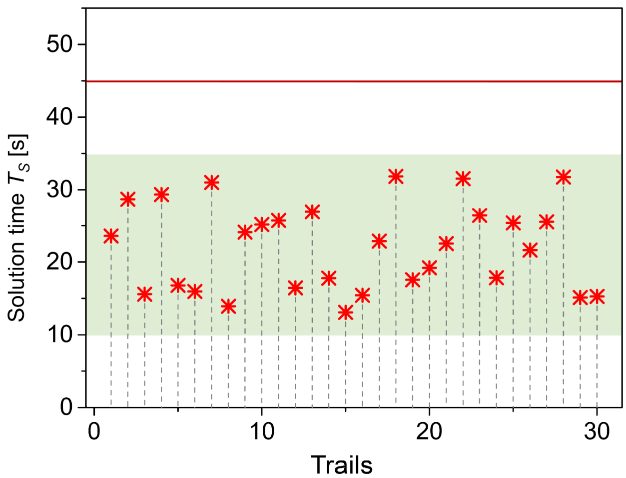

Fig. 6. T_s value distribution over 30 repeated trials.

VI. Conclusion

This article proposes a grouping-based algorithm of optimization for the multirobot pattern formation in an environment containing obstacles. Based on the specific grouping algorithm, the randomly initialized multirobots are grouped. Each group of robots first obtains its optimal pattern $\mathbf{Q}^*$ independently in parallel. Then, under an iterative controller, all robots in each group reach the assigned optimal pattern goal position without collision through coordination within and between groups to achieve the optimal pattern formation. The simulation results of the multiletters pattern formation verify the effectiveness of our algorithm. A comparison with the algorithm without grouping and the algorithm given in [17] illustrates the superior performance of our algorithm. This article studies the optimization for the pattern formation in a static obstacle environment. We use an iterative optimization method, that is, decision-making and execution, cyclically until the pattern is formed. Therefore, in each iteration, dynamic obstacles are treated as static obstacles, and the idea of iterative optimization can also be extended to a dynamic obstacle environment. The experimental results also show that the robots reach their goals at inconsistent times, which causes the robot to waste time waiting. In future work, we will consider addressing this situation.

The methodology presented here has been used for pattern formation. However, its core is to solve the global optimal objective function under the constraints of nonlinear inequalities. In the future, we can adopt a fully distributed optimization method [30], [31] to enhance the robustness of the algorithm, and extend the optimization-based problem formulation in this article to many other problems, including simultaneous localization and grouping.

References

[1] J. Lwowski, A. Majumdar, P. Benavidez, J. J. Prevost, and M. Jamshidi, "Bird flocking inspired formation control for unmanned aerial vehicles using stereo camera," *IEEE Syst. J.*, vol. 13, no. 3, pp. 3580–3589, Sep. 2019.

[2] X. Luo, X. Li, X. Li, J. Yan, and X. Guan, "Globally stable formation control of nonholonomic multiagent systems with bearing-only measurement," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2901–2912, Jun. 2020.

[3] J. Wu, H. Wang, N. Li, and Z. Su, "Formation obstacle avoidance: A fluid-based solution," *IEEE Syst. J.*, vol. 14, no. 1, pp. 1479–1490, Mar. 2020.

[4] L. Ren *et al.*, "An optimal task allocation approach for large-scale multiple robotic systems with hierarchical framework and resource constraints," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3877–3880, Dec. 2018.

[5] M. Turpin, N. Michael, and V. Kumar, "Decentralized formation control with variable shapes for aerial robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 23–30.

[6] Q. Li, "2019 world drone conference shenzhen held thousands of drones unveiled." Accessed: Jun. 21, 2019. [Online]. Available: http://sz.cnr.cn/szfwgb/szyw/20190621/t20190621_524658879.shtml

[7] Z. Ma and S. Akella, "Coordination of droplets on light-actuated digital microfluidic systems," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2012, pp. 2510–2516.

[8] V. Shekar, M. Campbell, and S. Akella, "Towards automated optoelectrowetting on dielectric devices for multi-axis droplet manipulation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 1439–1445.

[9] M. Rubenstein, A. Cornejo, and R. Nagpal, "Programmable self-assembly in a thousand-robot swarm," *Science*, vol. 345, no. 6198, pp. 795–799, 2014.

[10] P. C. Lusk, X. Cai, S. Wadhwania, A. Paris, K. Fathian, and J. P. How, "A distributed pipeline for scalable, deconflicted formation flying," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5213–5220, Oct. 2020.

[11] H. Yang, S. Cao, L. Bai, Z. Zhang, and J. Kong, "A distributed and parallel self-assembly approach for swarm robotics," *Robot. Auton. Syst.*, vol. 118, pp. 80–92, 2019.

[12] S. Jiang, J. Liang, J. Cao, J. Wang, J. Chen, and Z. Liang, "Decentralized algorithm for repeating pattern formation by multiple robots," in *Proc. IEEE 25th Int. Conf. Parallel Distrib. Syst.*, 2019, pp. 594–601.

[13] M. Coppola, J. Guo, E. Gill, and G. C. H. E. de Croon, "Provable self-organizing pattern formation by a swarm of robots with limited knowledge," *Swarm Intell.*, vol. 13, pp. 59–94, 2019.

[14] M. T. Abdullah, M. J. A. Sourov, S. Rahman, and S. Sarker, "Simulation of pattern formation of swarm with minimum shape parameters," in *Proc. Joint 9th Int. Conf. Inform., Electron. Vis./4th Int. Conf. Imag., Vis. Pattern Recognit.*, 2020, pp. 1–6.

[15] Q. Wang, J. Huang, and X. Mao, "A fast self-organizing pattern formation method for swarm robots in dynamic multi-region environments," in *Proc. 5th Int. Conf. Automat., Control Robot. Eng.*, 2020, pp. 124–129.

[16] J. Alonso-Mora, A. Breitenmoser, M. Rufli, R. Siegwart, and P. A. Beardsley, "Image and animation display with multiple mobile robots," *Int. J. Robot. Res.*, vol. 31, no. 6, pp. 753–773, 2012.

[17] G. Li, L. Dong, H. Xu, and Y. Lin, "Research on region coverage approach with swarm robots," *Jiqiren/robot*, vol. 39, no. 5, pp. 670–679, 2017.

[18] S. Agarwal and S. Akella, "Simultaneous optimization of assignments and goal formations for multiple robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 6708–6715.

[19] J. C. Derenick and J. R. Spletzer, "Convex optimization strategies for coordinating large-scale robot formations," *IEEE Trans. Robot.*, vol. 23, no. 6, pp. 1252–1259, Dec. 2007.

[20] G. Raja, S. Anbalagan, V. S. Narayanan, S. Jayaram, and A. Ganapathisubramaniyan, "Inter-UAV collision avoidance using deep-q-learning in flocking environment," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf.*, 2019, pp. 1089–1095.

[21] F. Zhang, T. Wang, Q. Li, and J. Xin, "An iterative optimization approach for multi-robot pattern formation in obstacle environment," *Robot. Auton. Syst.*, vol. 133, no. 103645, pp. 1–12, 2020.

[22] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2008, pp. 1928–1935.

[23] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[24] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of Massive Datasets*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2014.

[25] T. J. Van Roy and L. A. Wolsey, "Integer programming," Wiley, 2009, Art. no. 864. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer

[26] ILOG, *ILOG CPLEX User's Manual*, ILOG, 2002. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer

[27] D. P. Bertsekas, "The auction algorithm: A distributed relaxation method for the assignment problem," *Ann. Oper. Res.*, vol. 14, no. 1, pp. 105–123, 1988.

[28] A. G. Richards and J. P. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *Proc. Amer. Control Conf.*, 2002, pp. 1936–1941.

[29] R. Lazimy, "Mixed-integer quadratic programming," *Math. Program.*, vol. 22, no. 1, pp. 332–349, 1982.

[30] Z. Tian, W. Wu, and B. Zhang, "A mixed integer quadratic programming model for topology identification in distribution network," *IEEE Trans. Power Syst.*, vol. 31, no. 1, pp. 823–824, Jan. 2016.

[31] S. Li, Z. Wang, and Y. Li, "Using Laplacian Eigenmap as heuristic information to solve nonlinear constraints defined on a graph and its application in distributed range-free localization of wireless sensor networks," *Neural Process. Lett.*, vol. 37, no. 1, pp. 411–424, 2013.

[32] S. Li, Y. Lou, and B. Liu, "Bluetooth aided mobile phone localization: A nonlinear neural circuit approach," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 4, pp. 1–14, 2014.

Tingting Wang received the B.S. degree in railway traffic signal and control from the North University of China, Taiyuan, China, in 2017, where she is currently working toward the M.S. degree in control engineering with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China.

Her research interests include multiagent, pattern formation, and nonlinear programming.

Fangfang Zhang received the B.E. degree in in applied mathematics, the M.E. degree in applied mathematics, and the Ph.D. degree in control science and engineering from Shandong University, Jinan, China, in 2008, 2011, and 2015, respectively.

He is currently an Associate Professor with Zhengzhou University, Zhengzhou, China.

His research interests include optimal control of multiagent systems, multirobot formation, and machine vision.

Jianbin Xin received the M.Sc. degree in control science and engineering from Xi'an Jiaotong University, Xi'an, China, in 2010, and the Ph.D. degree specialized in operational control of automated container terminals from the Delft University of Technology, Delft, Netherlands, in 2015.

He is currently an Associate Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou, China.

His research interests include modeling and control of smart logistics systems and hybrid systems control.

Yanhong Liu received the B.E. degree in automation from the Zhengzhou University of Light Industry, Zhengzhou, China, in 1992, and the M.E. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, China, in 2002 and 2006, respectively.

From 2012 to 2013, she worked as a Visiting Scholar with the University of California at San Diego, La Jolla, CA, USA. She is currently a Professor with the School of Electrical Engineering, Zhengzhou University, Zhengzhou. Her research interests include nonlinear system modeling and control, robotic control, and human–robot interactions and collaborations.