Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Dynamic grouping of heterogeneous agents for exploration and strike missions*

Chen CHEN[†1], Xiaochen WU[†‡1], Jie CHEN[1,2], Panos M. PARDALOS[†‡3], Shuxin DING[4]

[1]*School of Automation, Beijing Institute of Technology, Beijing 100081, China*

[2]*Department of Control Science and Engineering, Tongji University, Shanghai 200092, China*

[3]*Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL 32611, USA*

[4]*Signal and Communication Research Institute, China Academy of Railway Sciences Corporation Limited, Beijing 100081, China*

[†]E-mail: xiaofan@bit.edu.cn; wsygdhrwxc@sina.com; pardalos@ufl.edu

Received July 17, 2020; Revision accepted Nov. 16, 2020; Crosschecked Sept. 1, 2021

Abstract: The ever-changing environment and complex combat missions create new demands for the formation of mission groups of unmanned combat agents. This study aims to address the problem of dynamic construction of mission groups under new requirements. Agents are heterogeneous, and a group formation method must dynamically form new groups in circumstances where missions are constantly being explored. In our method, a group formation strategy that combines heuristic rules and response threshold models is proposed to dynamically adjust the members of the mission group and adapt to the needs of new missions. The degree of matching between the mission requirements and the group's capabilities, and the communication cost of group formation are used as indicators to evaluate the quality of the group. The response threshold method and the ant colony algorithm are selected as the comparison algorithms in the simulations. The results show that the grouping scheme obtained by the proposed method is superior to those of the comparison methods.

Key words: Multi-agent; Dynamic missions; Group formation; Heuristic rule; Networking overhead
https://doi.org/10.1631/FITEE.2000352 **CLC number:** TP391.9

1 Introduction

Multi-agent systems (MASs) are widely used to perform complex missions in different fields (Merabet et al., 2014), such as fire control and rescue missions or military detection and strike missions. The first problem to be solved in those missions is how to organize multiple agents to complete a mission, that is, how to assign the overall missions to each agent and ensure that the agents effectively cooperate. Group formation has a great influence on the ultimate performance of the whole MAS.

When agents have different abilities or play different roles, it is particularly important to form their groups according to the needs of the mission. The problem of finding a reasonable partition of agents so that some utility functions are maximized is known to be NP-hard concerning different utility functions (Gerkey and Matarić, 2004; Vig and Adams, 2006). In the field of artificial intelligence and cooperative systems, especially in distributed collaboration, experts and scholars have done a lot of research on the organizational structure and synergy of MAS. They have focused on topics such as emergent rule theory

‡ Corresponding authors

* Project supported by the National Natural Science Foundation of China (No. 61773066) and the Foundation of China Academy of Railway Sciences Corporation Limited (No. 2019YJ071)

ORCID: Chen CHEN, https://orcid.org/0000-0001-9354-6974; Xiaochen WU, https://orcid.org/0000-0002-9871-5543; Panos M. PARDALOS, https://orcid.org/0000-0001-9623-8053

(Murphey and Pardalos, 2002), game theory (Pardalos et al., 2008), cooperative autonomous systems (Butenko et al., 2003; Kim Y et al., 2008; Khoshnoud et al., 2019), and the hierarchical cooperation model (Butenko et al., 2003; Hirsch et al., 2009).

Many scholars have studied the self-organization or dynamic grouping of agents. Research results in the context of confrontation are rich (Ducatelle et al., 2010; Singh et al., 2010; Liu et al., 2013; Necsulescu and Schilling, 2015; Orfanus et al., 2016; Skorobogatov et al., 2020).

In terms of solutions, in addition to the classic models and the methods described above, some heuristic rules have been used in the formation of agent groups and their mission allocation. Ramchurn et al. (2010) and Padmanabhan and Suresh (2015) focused on solving the mission group formation problem by heuristic methods; Oh et al. (2018), Nejad and Kashan (2019), and Guo et al. (2020) designed heuristic methods to deal with the mission allocation problem.

However, as the size of the agent community expands, the versatility of some heuristic methods becomes limited and they no longer apply to more complex mission environments. In this case, people turn to the individual behavior of the natural community and its emerging group behavior, and apply it to the agent system so that the individuals can spontaneously form groups to perform complex missions according to dynamic mission information. In this process, agents demonstrate greater self-organization, collaboration, and adaptability to the environment. For example, in Yang et al. (2014) and Khan et al. (2019), a special ant colony algorithm was used to solve the problem of constructing an intelligent dynamic alliance. In addition to the ant colony optimization (ACO), other bio-population-based heuristics have been used in group formation problems. In George et al. (2010) and Manathara et al. (2011), the particle swarm algorithm and some new heuristic strategies were used to solve the problem of group formation.

As the scale of the task group formation problem continues to change, researchers have tried different methods to solve it. These existing methods have relatively good results when dealing with the dynamic grouping of a single type of agents. However, with the expanded number of agent types, the existing methods can no longer perform the task of group formation based on the mission requirement for heterogeneous agents. Therefore, in this study, different from the existing results, the matching of heterogeneous agents' capabilities with mission requirements and new evaluation criteria are specifically considered in the grouping process.

This study aims to solve the mission group formation problem of heterogeneous agents in the battlefield environment. Each mission has different priority and capability requirements, and different agents must interact cooperatively. The ability requirement represents the minimum ability required to destroy the target. The purpose of the mission is to find the targets and eliminate them as soon as possible.

Fig. 1 briefly describes the process of collaborative mission execution by heterogeneous agents. In the left module, there are many different types of agents that need to form mission groups to perform missions 1 and 2. When new missions (missions 3 and 4) are discovered, agents must adjust the grouping pattern according to the new mission status, form a new topology structure, and adapt to the new mission requirements (as shown in the right-hand side of Fig. 1). An algorithm needs to be designed to achieve dynamic grouping. In the algorithm, heuristic rules and the response threshold method are combined to form a hybrid grouping strategy, which ensures the realization of the above process.

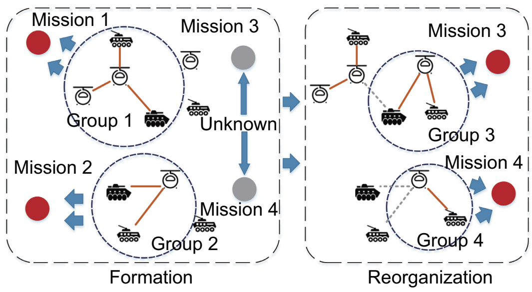

Fig. 1 Formation and reconstruction of mission groups

When agents perform missions in the group form, a mobile network (or some kind of connections) will be constructed to maintain information exchange among members in the group. However, excessive traffic will increase the network burden and the probability of the agent being detected by the enemy. To avoid the undesirable consequences caused by too much information transmission, unnecessary communication should be reduced when designing the heuristic rules for grouping.

The main contributions of this study can be summarized as follows: first, a model is established to describe the attributes of the mission and the agent, and the grouping scheme evaluation method is given; second, a hybrid algorithm combining heuristic rules and the improved response threshold method is designed to solve the dynamic agent grouping problem proposed in this study.

2 Problem formulation

Our research focuses on the dynamic group formation problem, which involves heterogeneous agents with a mission to attack some enemy targets. In the actual grouping process, a unified model is needed to accurately describe the status of the mission and the behavior and capabilities of the agents.

The model contains a mission area G, which is a rectangular, two-dimensional plane. In area G, there are q enemy targets (including stationary and moving targets) to be eliminated, and their initial positions are completely random. Each target is treated as a separate mission. Also, we deploy p freely movable agents in the area to detect and strike targets. Because each mission requires agents with various capabilities to collaborate, different types of agents must be grouped to accomplish the mission. When the mission situation changes based on the original grouping, the agents will reform their group according to the new mission list. The agents need to eliminate as many enemy targets as possible.

2.1 Agents

1. Definitions

First, we give the following definitions of the agents used in this study:

(1) Each agent is a carrier of resource capabilities and a mission platform with certain autonomous capabilities;

(2) There are different types of agents with different capabilities;

(3) The number of agents is limited and it is not possible to perform all missions at the same time;

(4) Networking overhead will be generated when agents form new mission groups.

2. Types and topology

This study adopts two types of agents: detection agent and attack agent. The detection agents mainly conduct large-scale reconnaissance operations, discover new targets, and provide real-time updates of mission intelligence. The updated data serve as the basis for the current attack agent grouping. As the name implies, the attack agents mainly attack the enemy targets. The dynamic grouping method of these agents based on mission intelligence is our main research content. The attack agents can be divided into several types, because they have different capabilities to tackle various types of targets. Thus, we have to design a rational grouping method based on different types of agents to improve the efficiency of mission execution.

In the process of forming a mission group, a mobile network topology is built among agents and used for information exchange between individuals. The hierarchical network concept is introduced into the system, and we achieve a three-dimensional topology through hierarchical modeling.

All the agents involved in the mission form a hierarchical topology structure as shown in Fig. 2, which consists of two layers. The lower layer is the mission execution layer and contains mission groups composed of attack agents; the upper layer is the coordination layer, which contains only detection agents. In actual combat, a communication link is formed between the detection agent and the leader of the mission group. In addition to exploring the mission location and posting mission information, the detection agents need to coordinate among the mission groups when the mission is released. If each mission group is regarded as a small network, the detection agent can be understood as a mobile gateway node that is used for communication and coordination between networks.

3. Capabilities

The agents' capabilities are described by a simple slot model, which has been used by researchers in the area of resource collection (Moritz

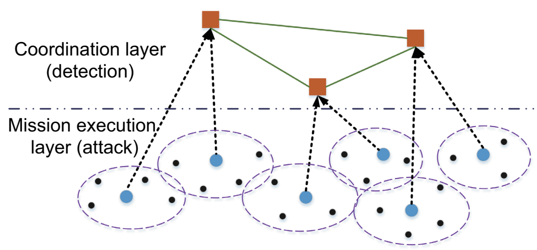

Fig. 2 Hierarchical topology

and Middendorf, 2015). In this model, a slot is the smallest relevant unit of the agents' capabilities.

As Fig. 3 shows, slots of different colors represent the different agent capabilities. The number of slot types represents the number of capabilities the agent has. Agents have different slots, which means that their ability levels are different. There are three capability slots in Fig. 3. This indicates that the agent has three different capabilities. We use p_{ij} to represent the value of the j^{th} capability of agent i. When $p_{ij} > 0$, it means that agent i has the j^{th} capability. In the problem we study, the numbers of agent capability slots are different, and the value of each capability is also different.

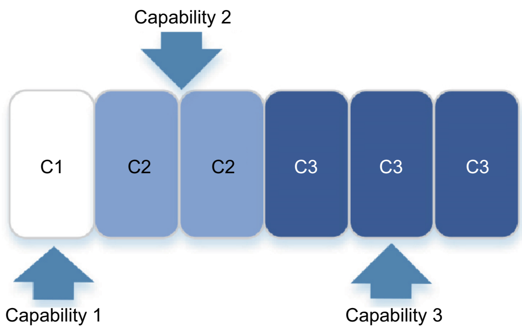

Fig. 3 Slot model

4. Constraints

Ignoring the impact of the environment, we assume that all agents can reach any location in the mission area. Thus, constraints on path feasibility are not considered in this study. In addition, regarding the characteristics of the agent itself, we consider two types of constraints in our research.

Each agent has an energy storage device. When the energy in the device is exhausted, the agent cannot move or participate in any mission group, and it will take some time to replenish the energy. We choose to use the maximum distance, $L_{\max}$, that the agent could move to indicate the maximum capacity of the battery or fuel tank. Let $E_{i\max} = L_{i\max}$; $E_{i\max}$ represents the maximum energy of agent i. Thus, the energy currently available on agent i can be expressed as

$$E_{ic} = f_{\text{ch}} E_{i\max} - L_{i\text{m}}, \tag{1}$$

where $L_{i\text{m}}$ indicates the mileage of agent i and f_{ch} represents the number of charges.

In the actual confrontation process, the amount of ammunition carried by the agent is limited, so in addition to energy constraints, ammunition constraints should be considered. We translate the ammunition constraints into the number of missions in which the agent could participate. Let am_i indicate the remaining number of times that agent i can participate in the mission. When

$$\begin{cases} E_{ic} > 0, \\ \text{am}_i > 0, \end{cases} \tag{2}$$

agent i is in a state that can be grouped.

2.2 Mission

The characteristics of the agent and the mission scenario have been introduced above. Next, some attributes of the mission are introduced.

In the combat environment of this study, there are multiple missions at the same time, and each mission is independent. Due to the limited capabilities of the agents, when attacking enemy targets, they need to form groups to complete the mission. Because our study focuses on the dynamic grouping mechanism of heterogeneous agents, we ignore the impact of the environment on their movement.

To improve the model's versatility, the mission settings need to be as close as possible to the actual situation. During the simulation, the positions of some targets are unknown and need to be obtained through exploration. The continuous updating of the mission list ensures the dynamic nature of the grouping process. In addition, some targets are removable, which improves the authenticity of the model. Moreover, the mission should be completed within a specified time, and when the time limit is exceeded, the mission is considered to have failed.

1. Mission characteristics

(1) Mission duration

The time elapsed from the generation of the mission to the announcement of the failure of the mission is denoted by t_{d}.

(2) Mission requirements for capabilities

Here, the capability requirement vector is used to represent a mission for each capability requirement. For a specific mission, the capability requirements match the types of capabilities that all intelligent agents have; that is, the dimension of the vector is the same as the number of the agent's slot types. The vector of capability requirements can be expressed as

$$\boldsymbol{D}_k = [d_{k1}, d_{k2}, \ldots, d_{kn}], \tag{3}$$

where d_{km} $(m = 1, 2, \ldots, n)$ represents the demand for the m^{th} capability of mission k, and n is the number of types of capability.

(3) Mission time

The mission time t_{ck} represents the time required to complete mission k. The sum of each ability requirement of the mission is positively correlated with the number of agents dispatched to perform this mission; thus, we fix the value of t_{ck}, which does not change with the needs of the mission.

Fig. 4 shows the basic mission flow. When the missions are not completed, the mission group needs to be reconstructed according to the new mission requirements until all missions are completed.

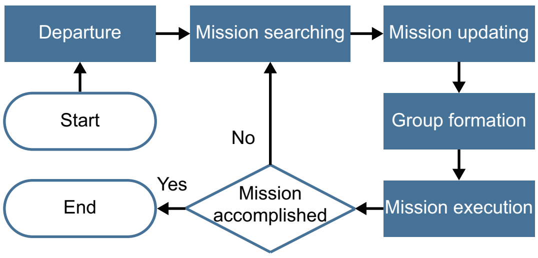

Fig. 4 Basic mission flow

2. Constraints

We assume that there is a mission k and a corresponding mission group i. Then the relationship between k and i meets the following condition:

$$\forall\, 1 \leq j \leq n,\ \exists\, d_{kj} \leq \mathrm{Cap}(p_{ij}), \tag{4}$$

where j represents a certain ability and $\mathrm{Cap}(p_{ij})$ represents the sum of ability j in group i. This condition ensures that the mission can be executed smoothly.

2.3 The proposed model

1. Objective 1: mission reward $(R(M))$

In the process of forming a group, the sum of the capabilities of the members in the group is required to be greater than the mission's demand for capabilities. According to the matching idea, certain principles should be satisfied for each mission group: the higher the degree of matching between the mission group's capabilities and the needs of the mission, the greater the benefit of mission completion (Shehory and Kraus, 1998). This is because, in ensuring the completion of the mission, if a mission group is used whose ability far exceeds the mission demand, the agent's capability resources will be wasted and the overall profit of the mission will be reduced. We measure mission reward $R(M)$ using the degree of ability matching. According to the above ideas, we give the numerical calculation method of $R(M)$ based on the matching degree:

$$R(M) = \sum_{k=1}^{n} P_k r_{Mk}, \tag{5}$$

where $R(M)$ represents the overall mission reward, P_k represents the priority of mission k, r_{Mk} represents the reward of mission k based on the matching degree, and the calculation method of r_{Mk} is as follows:

$$r_{Mk} = \begin{cases} b_k - \mathrm{ne}_k, & \text{accomplished}, \\ 0, & \text{failed}, \end{cases} \tag{6}$$

$$b_k = \gamma \sum_{j=1}^{n} d_{kj}, \tag{7}$$

$$\mathrm{ne}_k = \frac{\boldsymbol{O}_k \boldsymbol{D}_{\mathrm{r}k}^{\mathrm{T}}}{\sum_{j=1}^{n} D_{\mathrm{r}kj}}, \tag{8}$$

where b_k represents the ideal reward of mission k (that is, the benefits generated when the sum of the capabilities of the members of the group is exactly the same as the mission's capability needs), and it is measured by the sum of the capability requirements of mission k. Parameter γ is a weight coefficient.

According to the relationship between the reward and the matching degree mentioned above, when the capabilities cannot be fully matched, the negative reward ne_k generated by the redundant part of the capabilities needs to be subtracted from b_k. Eq. (8) is used to calculate the negative reward ne_k, where $\boldsymbol{O}_k$ represents the redundant part of the capability, and can be calculated by

$$\boldsymbol{O}_k = \mathbf{Cap}(\boldsymbol{p}_i) - \boldsymbol{D}_k. \tag{9}$$

$\boldsymbol{D}_{\mathrm{r}k}$ is a vector consisting of the reciprocal of each element in $\boldsymbol{D}_k$, and $\frac{\boldsymbol{D}_{\mathrm{r}k}^{\mathrm{T}}}{\sum_{j=1}^{n} D_{\mathrm{r}kj}}$ is used as the weight coefficient of $\boldsymbol{O}_k$ to measure the impact of each capability's overflow on the reward. $\boldsymbol{p}_i$ represents the sum of the capabilities of all agents in group i (represented in vector form, with p_{ij} (in Eq. (4)) being the j^{th} capability in vector $\boldsymbol{p}_i$). When the mission is completed, the specific mission reward can be calculated; otherwise, $r_{Mk} = 0$.

2. Objective 2: fuel cost $(F(M))$

In addition to mission reward, we have to calculate the cost of the missions. $F(M)$ represents the fuel cost generated during the movement of all agents and is described by the average moving distance of the agents. It can be calculated by

$$F(M)=\frac{1}{p}\sum_{i=1}^{p}L_{i\mathrm{m}}, \tag{10}$$

where $L_{i\mathrm{m}}$ indicates the mileage of agent i and p is the number of agents.

3. Objective 3: networking overhead ($E(M)$)

In the process of mission execution, periodic data interaction between individuals must be guaranteed by each mission group. In the grouping algorithm, we can influence only the communication data that are generated and the energy consumed during the networking process. We use energy consumption $E(M)$ as a parameter to measure the communication overhead and its impact when networking. The larger the value of $E(M)$, the greater the communication volume and energy consumed during networking, and the greater the cost of the mission.

To ensure stable data interaction during the mission, we choose a fixed distribution type, time division multiple access (TDMA), as the method for nodes to access the network. We do not study the access protocol or data structure; we only calculate the energy that is consumed in sending application data when the node uses the TDMA protocol to access the network. The following formula gives the calculation method:

$$E_{i\mathrm{pbit}}=\sum_{j=1}^{N_{\mathrm{s}i}}\left[\left(P_{\mathrm{ct}}+P_{\mathrm{cr}}\right)/\left(\zeta R_{\mathrm{s}}\right)+Td_{\mathrm{t}j}^{n_{\mathrm{t}}}\right], \tag{11}$$

where n_{t} is a parameter related to the transmission environment.

Eq. (11) was given by Cui et al. (2004) and Jiang et al. (2010), and it is used to calculate the energy consumption of nodes' transmitting data. $E_{i\mathrm{pbit}}$ is the energy consumed by node i to the leader per 1-bit of data transmission; $N_{\mathrm{s}i}$ is the hop number from node i to the group leader; P_{ct} and P_{cr} are the transmitting circuit power and receiving circuit power, respectively; ζ represents modulation parameter; R_{s} is the bit rate; under the condition of point-to-point transmission, T can be regarded as a constant, and it depends on the modulation form, circuit compensation, antenna power gain, and other parameters; $d_{\mathrm{t}j}$ represents the transmission distance from node j to the next node in the transmission link.

Under the conditions in this study, except for $d_{\mathrm{t}j}$ and $N_{\mathrm{s}i}$, the remaining parameters can be regarded as constants, and the values have been given by Jiang et al. (2010).

For all missions, the total energy consumption during the networking process is

$$E(M)=\sum_{i=1}^{N_{\mathrm{e}}}S_iE_{i\mathrm{pbit}}, \tag{12}$$

where N_{e} is the number of times that all nodes are connected to the network, and S_i represents the total amount of application data sent by node i.

Based on the above description, the model is formulated as follows:

$$\begin{cases}\max & R(M),\\ \min & F(M),\\ \min & E(M),\end{cases}$$
$$\text{s.t. conditions (2) and (4) are satisfied.}$$

We give three objectives in terms of mission reward and mission costs. The decision variables include the sum of the capabilities of the mission group $\mathbf{Cap}(p)$, the transmission distance d_{t}, the number of nodes applying to the network N_{e}, and the agent's mileage $\{L_{i\mathrm{m}}\}$. The values of the above decision variables depend only on the grouping scheme.

3 Dynamic group formation method based on the utility function and heuristic rules

In the previous grouping method, the mission team was immediately disbanded after completing the mission, and then the decentralized agents formed a new group according to the mission requirements.

Unlike previous research, we introduce a "dynamic adjustment" mechanism in the mission group reconstruction strategy. Once a mission is completed, team members will be adjusted to meet the new mission needs by combining available mission groups, absorbing new agents, group splitting, and other operations, instead of being disbanded immediately. Individuals in the group share member and mission information, and each group moves and performs the mission as a whole.

3.1 Utility function

Before grouping, we design the utility function to measure the matching degree between the agent and the mission. The higher the value of the utility function, the more suitable the agent is to complete the mission. We use the calculation results as the basis for dynamic grouping.

When determining the utility function, we consider the following factors:

1. Urgency of mission k, defined as follows:

$$\mathrm{ur}_k = \frac{1}{t_\mathrm{d} - t_{\mathrm{e}k}}, \tag{13}$$

where $t_{\mathrm{e}k}$ represents the time that has passed since the mission was discovered.

2. Euclidean distance

The distance d_{ik} from agent i to mission k is also an important factor that determines whether the agent is suitable for performing the mission:

$$d_{ik} = \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2}, \tag{14}$$

where (x_i, y_i) and (x_k, y_k) are the coordinates of agent i and mission k, respectively.

3. The evaluation value conf_{ik} represents the evaluation given by agent i on mission k, which can be defined as follows:

$$\mathrm{conf}_{ik} = \mathrm{e}^{-(t - t_{fk})}, \tag{15}$$

where t represents the current moment, and t_{fk} is the moment when mission k was discovered. The greater the value of conf_{ik}, the higher the probability that the agent believes that mission k can be completed.

Based on the above three factors, the utility function of agent i for mission k can be expressed as

$$u_{ik} = \frac{\alpha \cdot \mathrm{ur}_k + \mathrm{conf}_{ik} \cdot \beta}{d_{ik}}, \tag{16}$$

where α and β are the weight coefficients.

Eq. (16) can be used to calculate the utility value of each agent for mission k. For group j, the average utility value $\bar{u}_{jk}$ can be calculated to determine whether group j is suitable for performing mission k:

$$\bar{u}_{jk} = \frac{1}{n_j} \sum_{i=1}^{n_j} u_{ik}, \tag{17}$$

where n_j represents the number of agents in group j.

3.2 Heuristic rules for dynamic group formation

Next, a networking-overhead-based constructive heuristic (NCH) self-organizing rule is introduced. When designing the heuristic rules, we consider the network overhead and try to maintain the original group staffing during the formation of the new mission group, to reduce the communication overhead when networking.

We use mlist to save the missions that need to be executed currently, that is, a list of missions. |mlist| represents the number of missions in mlist.

Step 1: Sort current missions in mlist based on urgency.

Step 2 (for $1 < k < |\mathrm{mlist}|$): Based on the latest mission information recorded by the detection agent, determine the capability demand vector $\boldsymbol{D}_k$ of each mission in mlist, and the form of $\boldsymbol{D}_k$ is given by Eq. (3).

Step 3 (for $1 < k < |\mathrm{mlist}|$): According to the existing grouping situation, the currently idle mission groups are counted to form oglist. The number of groups in oglist is represented by |oglist|.

Step 4 (for $1 < j < |\mathrm{oglist}|$): Select groups in order in oglist, and calculate the average utility value $\bar{u}_{jk}$ of each group for mission k according to Eq. (17).

Step 5: Sort the groups in oglist according to the value of $\bar{u}_{jk}$ from high to low, and save the new group order in glist. The number of groups in glist is represented by |glist|, |glist| = |oglist|.

Step 6: Let group_k be the group used to perform mission k. Based on the needs of mission k, we will select the appropriate members in glist to join group_k to perform mission k. The purpose of this step is to select multiple individuals that are most suitable for performing mission k to form a group while maintaining the original mission group as much as possible.

For $1 < j < |\mathrm{glist}|$: When selecting members to form the mission k group, we will compare the capabilities of groups 1 to |glist| with the demand $\boldsymbol{D}_k$ of mission k in the order of glist. According to whether condition (4) is satisfied, it is divided into the following two cases:

(a) If the relationship between the capabilities of group j and mission k does not satisfy condition (4), it means that group j does not meet the current needs $\boldsymbol{D}_k$. Let all members of group j join group_k.

The difference between the capabilities of group j and mission k is calculated as the new $\boldsymbol{D}_k$. Then return to step 6, $j=j+1$.

(b) Conversely, if condition (4) is satisfied, it means that group j meets the current needs of mission k. At this time, if all the members of group j join group$_k$, some individual capabilities may be wasted. Therefore, we need to combine the improved response threshold method to select suitable individuals from group j to join group$_k$ and avoid wasting agents. The improved threshold model comes from Kim MH et al. (2014):

$$P(S_{uk},\theta_{uk})=\frac{S_{uk}^2}{S_{uk}^2+a\theta_{uk}^2+\tau_{uk}^{2b}}, \tag{18}$$

where S_{uk} represents the mission's stimulus for agent u, θ_{uk} is the threshold, τ_{uk} represents the time required for agent u to reach the position of mission k, and a and b are parameters. The lower an agent's threshold or the higher a mission's stimulus, the more likely it is for the agent to accept the mission.

Mission k has different stimuli for different agents, which can be calculated by

$$S_{uk}=\max\{\boldsymbol{D}_k\}\cdot \mathrm{Cap}\,(p_{uv}), \tag{19}$$

where $\mathrm{Cap}\,(p_{uv})$ is the v^{th} capability of agent u, and its type is the same as the type of capability most needed by mission k. If $\max\{\boldsymbol{D}_k\}=d_{kn}$, then $v=n$ and $\mathrm{Cap}\,(p_{uv})=\mathrm{Cap}\,(p_{un})$.

We let the agent choose mission k with probability $P(S_{uk},\theta_{uk})$ every second. After each selection, let the individuals who choose mission k join group$_k$. There are also two cases at this time. When the relationship between group$_k$ and $\boldsymbol{D}_k$ satisfies condition (4), the grouping of mission k is completed. Otherwise, if the responding agent is insufficient this time, subtract the sum of the capabilities of the responding agent from the current mission demand to obtain the new mission demand, update S_{uk}, and continue to respond at the next simulation step. Repeat the above operation until group$_k$ and mission k meet condition (4). After obtaining group$_k$, the remaining agents in group j form a new group j and continue to participate in the grouping of subsequent missions.

Through the above operations, we incorporate the response threshold method into the heuristic framework, and effectively solve the problem of screening agents.

Step 7: repeat steps 4–6 and terminate the grouping process until one of the following two conditions occurs:

(a) All missions in the current mission list are performed by a certain group.

(b) When forming a group for mission k in the list, the remaining idle agents are not enough to perform that mission.

When situation (b) occurs, to save time, the remaining idle agents go to the vicinity of mission k and stand by.

In addition, all agents participating in the grouping must satisfy constraint (2).

During the grouping process, some of the original connections will be disconnected, and new connections will be formed in the new mission. When choosing a leader for a new group, we try to choose the original leader included in the group, so that the connection between the leader and the surrounding nodes can be maintained.

Through the above method, the dynamic agent grouping problem can be solved. After the formation of the mission group, when the agent moves to the vicinity of the mission, if its distance from the leader or the nodes around the leader is less than the communication radius, it can send an application to join the network. We stipulate that the information transmission link from the member to the group leader should not exceed two hops at most.

4 Simulations

After designing and describing the model and the dynamic agent self-organizing method, we conduct a series of simulations based on the self-organizing method designed in the study. We want to determine the algorithm's performance using different scales and different scenarios through simulations.

The response threshold method introduced from Kim MH et al. (2014) and the adjusted ACO which is based on the model in this study are selected as the comparison algorithms.

Generally speaking, the ant colony algorithm is used to set up the population in the mission environment and spread the pheromone along the way through the ants. In the problem of this study, the pheromone needs to be set at the mission position to attract agents to perform this mission. The concentration of the pheromone of the k^{th} mission is represented by τ_k, and the probability that agent i chooses mission k can be calculated as follows:

$$P_{ik} = \frac{(\tau_k)^{\alpha_{\mathrm{d}}}}{\sum_{s=1}^{q}(\tau_s)^{\alpha_{\mathrm{d}}}}, \tag{20}$$

where τ_s represents the pheromone of the s^{th} mission and α_{d} is a heuristic factor. After a round of selection, if the needs of mission k are met, set τ_k to zero; otherwise, update τ_k according to the following formula and continue to attract agents:

$$\tau_k = \tau_k + \Delta\tau, \tag{21}$$

where $\Delta\tau$ is the concentration of increased pheromone.

4.1 Settings

Based on the problems studied in this study, we design four sets of simulations to compare the application of the algorithm with different mission numbers. Tables 1 and 2 show the different values used for the test runs for all model parameters. The parameter values in Table 1 can be adjusted in the simulation, and the parameter values in Table 2 are derived from Cui et al. (2004) and Jiang et al. (2010). From these tables, we can see the specific parameter settings when we perform the four sets of missions of different sizes in the same mission area.

In the simulation, we use one type of detection agents and three types of attack agents. The parameters of the four agent types are given in Table 3. Some values in Table 3 refer to the relevant parameters of actual weapons and equipment. Table 4 sets the parameters of the algorithms for comparison. For the two comparison algorithms, we select parameters that can achieve better simulation results.

Fig. 5 shows the simulation scenario. The enemy

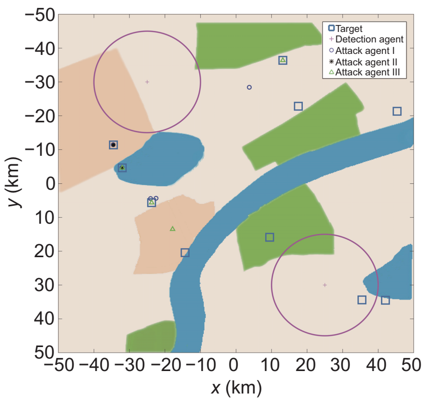

Fig. 5 Simulation scenario

Table 1 Variable parameters in simulations

Parameter	Definition	Value			
		Scenario 1	Scenario 2	Scenario 3	Scenario 4
G	Mission area (km^2)	100×100	100×100	100×100	100×100
q	Number of enemy units	10	25	35	50
–	Enemy location	Random	Random	Random	Random
–	Number of types of agents	4	4	4	4
–	Number of types of capabilities	3	3	3	3
t_{sp} (s)	Simulation step	1	1	1	1
$t_{\mathrm{c}k}$ (s)	Time required to complete mission k	5	5	5	5
t_{d} (s)	Mission duration	20	20	25	30
t_{r} (s)	Replenishing time	2	2	2	2
α	Weight coefficient	10	10	10	10
β	Weight coefficient	5	5	5	5
γ	Weight coefficient	0.6	0.6	0.6	0.6

Table 2 Fixed parameters in simulations*

Parameter	Definition	Value	Parameter	Definition	Value
P_{ct}	Transmitting circuit power	98.2 mW	R_{s}	Bit rate	10^4 symbols/s
P_{cr}	Receiving circuit power	112.5 mW	n_{t}	Empirical parameter (constant)	3
ζ	Modulation parameter	1	S_i	Amount of application data	20 bits
T	Empirical parameter (constant)	10^{-18}			

* Taken from Cui et al. (2004) and Jiang et al. (2010)

deployment units (targets) are randomly generated in the mission area as the simulation advances. The large circle indicates the detection range of a detection agent positioned at the centroid. After the simulation starts, the detection agent loops through the mission area to update the mission information. The three types of attack agents perform strikes based on the grouping results from the edge of the mission area.

4.2 Results

By simulating the four mission scenarios, we can compare the running results of the dynamic group formation strategy under different mission numbers. We separately conduct 20 simulations on the four mission scenarios given in Table 1.

Figs. 6–9 show the simulation results obtained under the four different mission scenarios. In these figures, the results of three objectives under different algorithms are shown. We perform simulations on missions of different sizes, and the results show that the NCH method is superior to the comparison algorithms in all three objectives. As the number of missions increases, the performance of NCH is improved. Figs. 8 and 9 clearly reflect the advantages of NCH compared to the two comparison algorithms

Table 3 Parameters of agents

Agent type	Velocity (km/h)	Fuel	Ammunition	Anti-air	Ground attack	Maneuverability	Quantity
1	300	8000	–	–	–	10	2
2	100	800	16	3	4	4	8
3	60	400	40	1	5	2	12
4	70	400	12	5	2	3	10

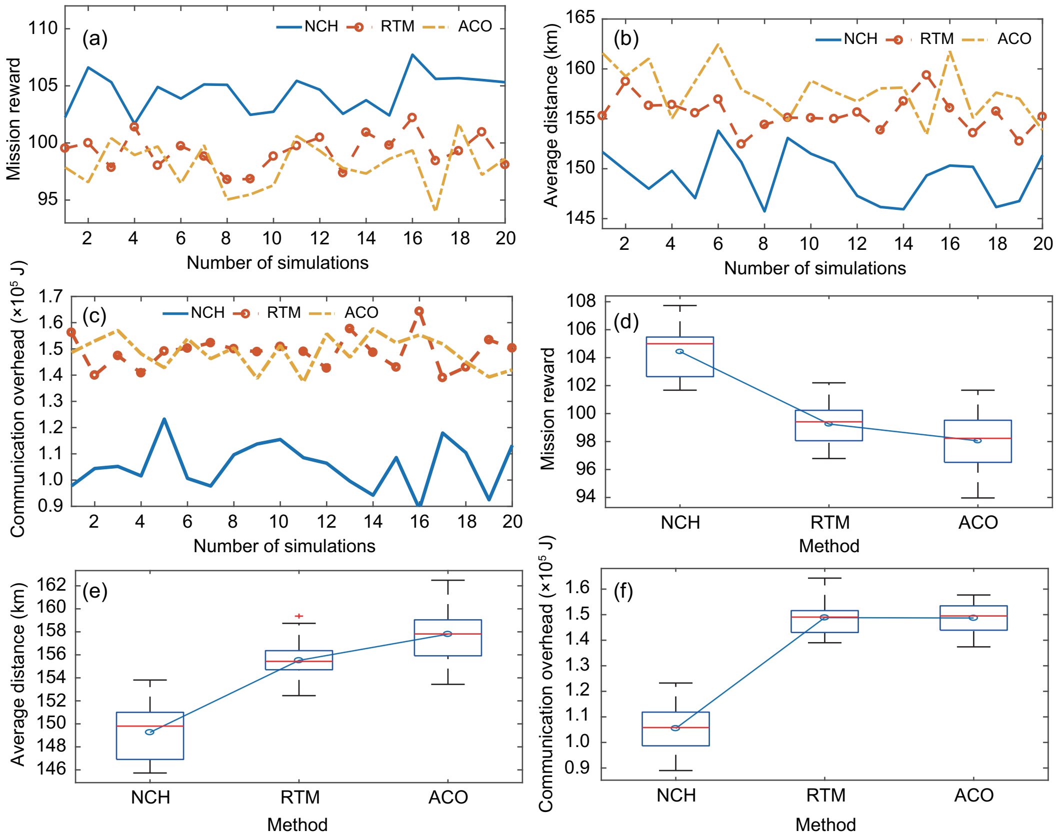

Fig. 6 Simulation results under the scenario with 10 missions (scenario 1): mission reward (a), average distance (b), and communication overhead (c) under the three algorithms; box plots showing the mean value and fluctuation range of the mission reward (d), average distance (e), and communication overhead (f) under the three algorithms

Table 4 Parameters of the algorithms to be compared

Algorithm	Parameter	Definition	Value	Algorithm	Parameter	Definition	Value
ACO	α_d	Heuristic factor	1.5	RTM*	Θ_{max}	Maximum threshold	40
	$\Delta\tau$	Pheromone increment	0.2		a	Weight coefficient	2
					b	Weight coefficient	1.5

* Taken from Kim MH et al. (2014)

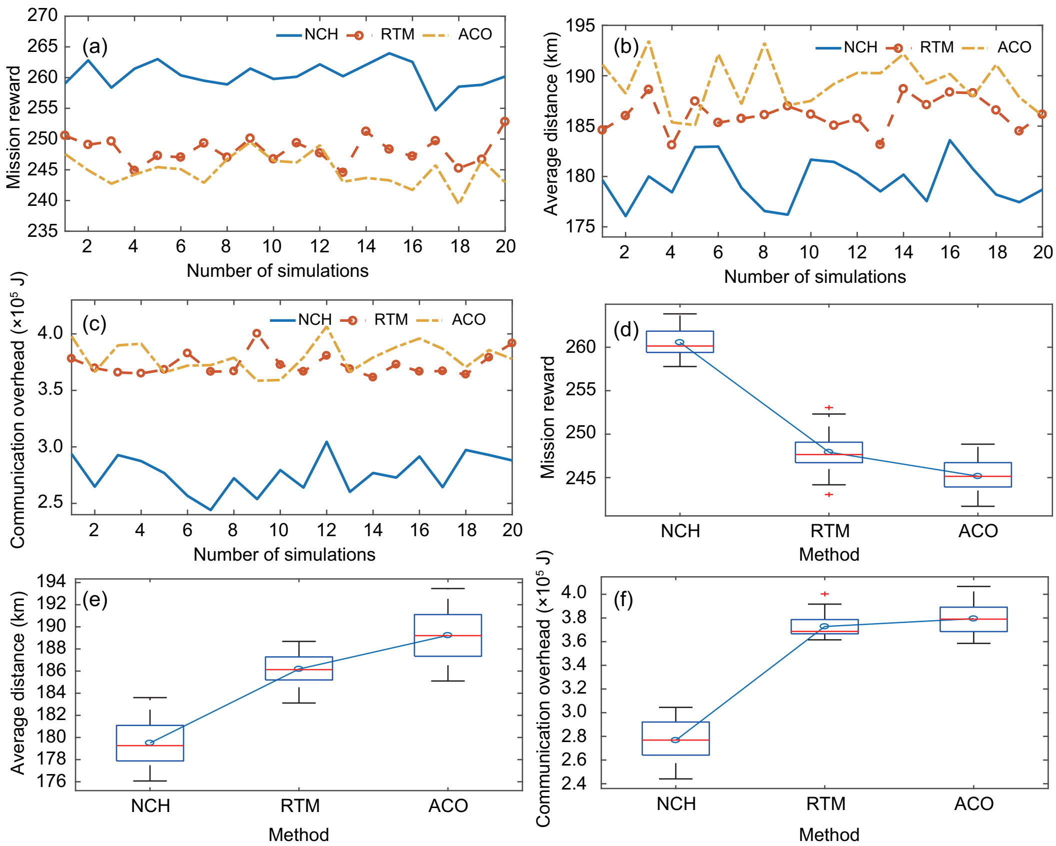

Fig. 7 Simulation results under the scenario with 25 missions (scenario 2): mission reward (a), average distance (b), and communication overhead (c) under the three algorithms; box plots showing the mean value and fluctuation range of the mission reward (d), average distance (e), and communication overhead (f) under the three algorithms

when the number of missions is large.

Moreover, box plots of the three objectives, representing the mean value and fluctuation range of different objectives under different methods, are displayed in these figures. Through the box plot, simulation and comparison results can be more intuitively reflected. As the box plots show, compared with the comparison algorithms, when NCH is used to dynamically form the mission group, the average of the three objectives is better. However, the advantages of NCH are not obvious in terms of the volatility of the solution results, which means that in terms of stability, our method (NCH) has room for improvement.

Table 5 shows specific grouping statistics of scenario 1. Through the grouping statistical results, it can be intuitively understood that because the heuristic rules of the NCH method consider the energy consumption factor, the original group member structure can be maintained as much as possible when the method is used for dynamic grouping. In contrast, the memberships of the mission groups of the two comparison algorithms are more random. The comparison results can prove the effectiveness of heuristic rules and the NCH algorithm.

In terms of the algorithm characteristics, the

Table 5 Statistics of grouping results (scenario 1)

Method	T=10 s		T=20 s		T=30 s	
	Mission	Agents	Mission	Agents	Mission	Agents
NCH	2	3, 4, 5, 27, 28, 29, 30	1	10, 11, 12, 28	3	2, 17, 18, 19, 20, 22, 26
	4	2, 17, 18, 19, 20, 26	7	1, 13, 14, 15	10	3, 4, 5, 23, 24, 25
	5	1, 13, 14, 15, 16	8	16, 21		
	6	21, 22, 23, 24, 25	9	6, 7, 8, 9		
ACO	2	2, 4, 5, 18, 26, 28, 29	1	7, 10, 12, 13, 21	7	1, 6, 7, 8, 13
	4	3, 7, 17, 19, 20, 27	3	9, 11, 18, 19, 26, 27, 30	9	5, 10, 14, 15
	5	1, 11, 14, 23, 25, 30	8	17, 22, 29	10	3, 4, 9, 20, 24, 28
	6	6, 15, 16, 22, 24, 26				
RTM	2	1, 4, 13, 24, 27, 28, 30	1	11, 12, 16, 23	7	1, 12, 13, 17, 18
	4	2, 5, 7, 17, 19, 21, 25	3	2, 14, 18, 19, 20, 22, 26	8	11, 15, 23
	5	3, 14, 15, 16, 18	9	6, 7, 8, 9, 17		
	6	6, 20, 22, 23, 26, 29	10	3, 4, 5, 10, 13, 24, 25		

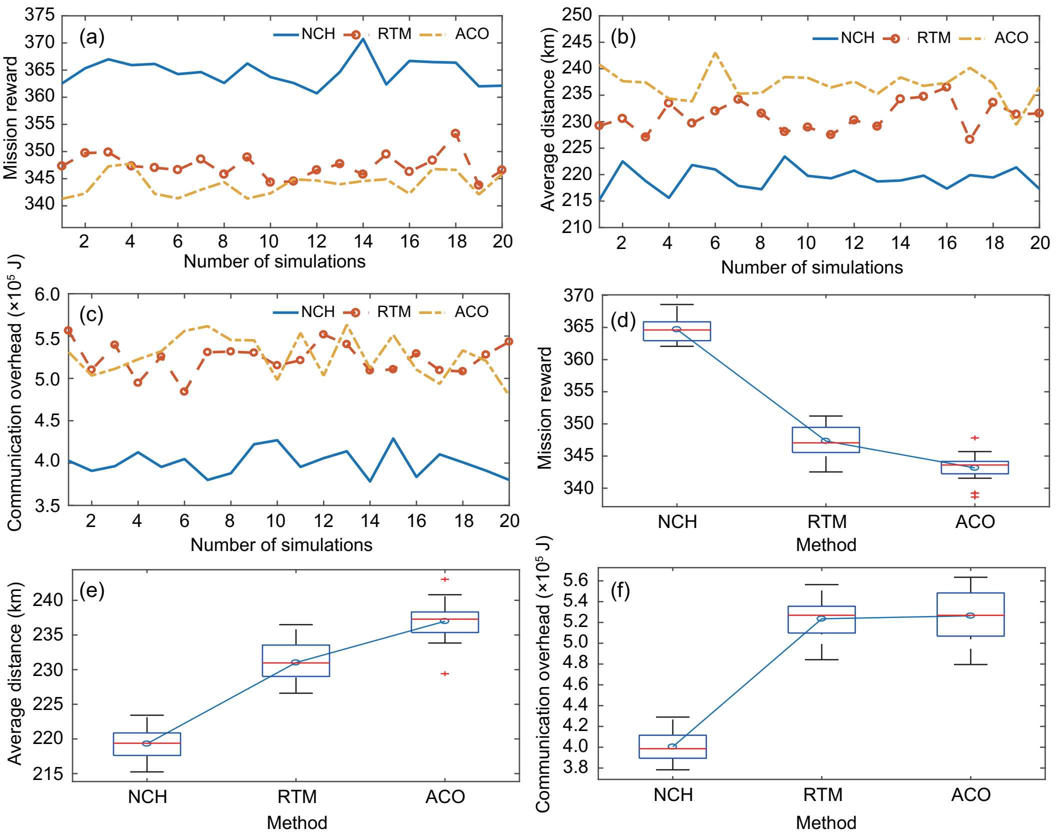

Fig. 8 Simulation results under the scenario with 35 missions (scenario 3): mission reward (a), average distance (b), and communication overhead (c) under the three algorithms; box plots showing the mean value and fluctuation range of the mission reward (d), average distance (e), and communication overhead (f) under the three algorithms

use of the NCH method is based on the layered distributed system designed in this study. The implementation of heuristic rules also depends on some simple decisions made by the detection agent (gateway node), such as sorting groups according to the utility value. Therefore, NCH is not completely a distributed algorithm, but combines some features of a centralized algorithm. The comparison algorithms (ACO and RTM) are distributed algorithms, which can completely realize the dynamic self-organization

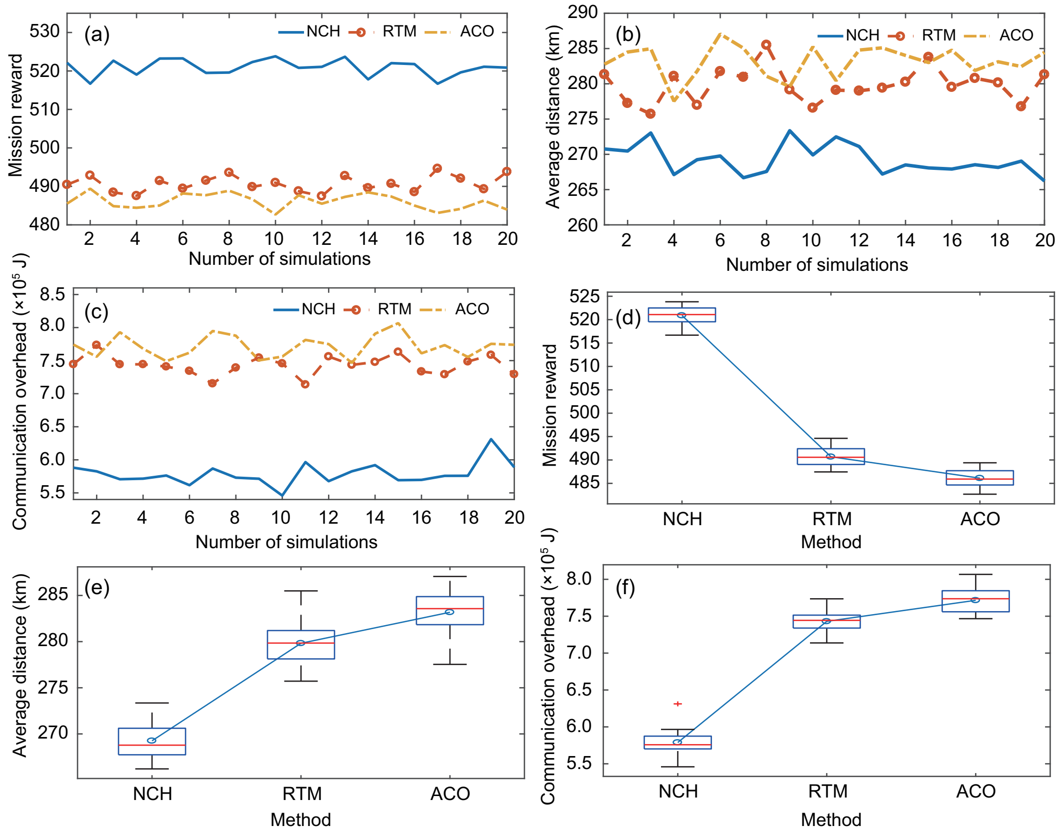

Fig. 9 Simulation results under the scenario with 50 missions (scenario 4): mission reward (a), average distance (b), and communication overhead (c) under the three algorithms; box plots showing the mean value and fluctuation range of the mission reward (d), average distance (e), and communication overhead (f) under the three algorithms

of agents without relying on superior nodes. In this respect, the performance of NCH is worse than that of the comparison algorithms. In other words, NCH has certain advantages in solving the problems in this paper, but under other conditions, the performance of NCH may not be as good.

5 Conclusions and future work

The purpose of our research is to design a heuristic mission group formation approach with some self-organizing characteristics based on the dynamic mission requirements. In the actual battlefield, frequent transmission of data may cause nodes to be detected, or consume too much energy and lose communication ability for a period of time. We have designed a series of heuristic rules to preserve the original group's organization as much as possible when forming a new group. This strategy effectively reduces the traffic generated by related steps by reducing the disconnection and reconstruction operations of links between nodes. In addition, based on the ability matching principle, we have made adjustments to the existing self-organizing algorithm and reduced the wasting of agent capabilities during the grouping process. The adjusted self-organizing algorithm and heuristic rules together form the mission group dynamic formation algorithm described in this study.

In the simulations, we have designed a mission scenario where heterogeneous agents search and attack enemy targets. Three objectives have shown that the NCH method has advantages in solving this problem.

In future work, more complex problems will be considered. In actual combat, when different types of ammunition are carried, the capabilities of each agent will need to be reconfigurable. Also, the actual mission environment may contain many obstacles or unknown factors, which will affect the movement of agents and their group formation. Therefore, in the next step, we will study the dynamic grouping of agents based on the above new requirements and constraints.

In terms of applying the method, the dynamic self-organizing method studied in this study can be applied in the field of combat, and to the grouping problem of other kinds of missions. In future work, we will transform the model and consider the characteristics of other agents to expand the application area of the proposed method.

Contributors

Chen CHEN and Xiaochen WU conceived the idea of this research and studied the literature. Chen CHEN designed the simulations. Xiaochen WU processed the data. Chen CHEN and Xiaochen WU drafted the paper. Panos M. PARDALOS and Shuxin DING helped organize the paper. Chen CHEN, Xiaochen WU, and Jie CHEN revised and finalized the paper.

Compliance with ethics guidelines

Chen CHEN, Xiaochen WU, Jie CHEN, Panos M. PARDALOS, and Shuxin DING declare that they have no conflict of interest.

References

Butenko S, Murphey R, Pardalos PM, 2003. Cooperative Control: Models, Applications and Algorithms. Springer, Dordrecht, the Netherlands.

Cui SG, Goldsmith AJ, Bahai A, 2004. Energy-efficiency of MIMO and cooperative MIMO techniques in sensor networks. *IEEE J Sel Areas Commun*, 22(6):1089-1098. https://doi.org/10.1109/JSAC.2004.830916

Ducatelle F, di Caro G, Förster A, et al., 2010. Adaptive navigation in a heterogeneous swarm robotic system. Proc 4th Int Conf on Cognitive Systems, p.87-94.

George JM, Sujit PB, Sousa JB, 2010. Coalition formation with communication delays and maneuvering targets. Proc AIAA Guidance, Navigation, and Control Conf, p.28-32. https://doi.org/10.2514/6.2010-8422

Gerkey BP, Matarić MJ, 2004. A formal analysis and taxonomy of task allocation in multi-robot systems. *Int J Robot Res*, 23(9):939-954. https://doi.org/10.1177/0278364904045564

Guo M, Xin B, Chen J, et al., 2020. Multi-agent coalition formation by an efficient genetic algorithm with heuristic initialization and repair strategy. *Swarm Evol Comput*, 55:100686. https://doi.org/10.1016/j.swevo.2020.100686

Hirsch MJ, Commander CW, Pardalos PM, et al., 2009. Optimization and Cooperative Control Strategies: Proceedings of the 8th International Conference on Cooperative Control and Optimization. Springer, Berlin, Germany. https://doi.org/10.1007/978-3-540-88063-9

Jiang RY, Ji W, Zheng BY, 2010. Joint optimization of energy consumption in cooperative wireless sensor networks. *J Electron Inform Technol*, 32(6):1475-1479 (in Chinese).

Khan A, Aftab F, Zhang ZS, 2019. Self-organization based clustering scheme for FANETs using glowworm swarm optimization. *Phys Commun*, 36:100769. https://doi.org/10.1016/j.phycom.2019.100769

Khoshnoud F, Esat II, de Silva CW, et al., 2019. Quantum network of cooperative unmanned autonomous systems. *Unmanned Syst*, 7(2):137-145. https://doi.org/10.1142/S2301385019500055

Kim MH, Baik H, Lee S, 2014. Response threshold model based UAV search planning and task allocation. *J Intell Robot Syst*, 75(3-4):625-640. https://doi.org/10.1007/s10846-013-9887-6

Kim Y, Gu DW, Postlethwaite I, 2008. Real-time optimal time-critical target assignment for UAVs. In: Pardalos PM, Murphey R, Grundel D, et al. (Eds.), Advances in Cooperative Control and Optimization. Springer, Berlin, p.265-280, https://doi.org/10.1007/978-3-540-74356-9

Liu MJ, Lin J, Yuan YY, 2013. Research of UAV cooperative reconnaissance with self-organization path planning. Proc Int Conf on Computer, Networks and Communication Engineering, p.207-213. https://doi.org/10.2991/iccnce.2013.51

Manathara JG, Sujit PB, Beard RW, 2011. Multiple UAV coalitions for a search and prosecute mission. *J Intell Robot Syst*, 62(1):125-158. https://doi.org/10.1007/s10846-010-9439-2

Merabet GH, Essaaidi M, Talei H, et al., 2014. Applications of multi-agent systems in smart grids: a survey. Proc Int Conf on Multimedia Computing and Systems, p.1088-1094. https://doi.org/10.1109/ICMCS.2014.6911384

Moritz RLV, Middendorf M, 2015. Decentralized and dynamic group formation of reconfigurable agents. *Memet Comput*, 7(2):77-91. https://doi.org/10.1007/s12293-014-0149-3

Murphey R, Pardalos PM, 2002. Cooperative Control and Optimization. Springer, Boston, USA. https://doi.org/10.1007/b130435

Necsulescu P, Schilling K, 2015. Automation of a multiple robot self-organizing multi-hop mobile ad-hoc network (MANET) using signal strength. Proc Int Instrumentation and Measurement Technology Conf, p.505-510. https://doi.org/10.1109/I2MTC.2015.7151319

Nejad MG, Kashan AH, 2019. An effective grouping evolution strategy algorithm enhanced with heuristic methods for assembly line balancing problem. *J Adv Manuf Syst*, 18(3):487-509. https://doi.org/10.1142/S0219686719500264

Oh G, Kim Y, Ahn J, et al., 2018. Task allocation of multiple UAVs for cooperative parcel delivery. Proc Advances in Aerospace Guidance, Navigation and Control, p.443-454. https://doi.org/10.1007/978-3-319-65283-2_24

Orfanus D, de Freitas EP, Eliassen F, 2016. Self-organization as a supporting paradigm for military UAV relay networks. *IEEE Commun Lett*, 20(4):804-807. https://doi.org/10.1109/LCOMM.2016.2524405

Padmanabhan M, Suresh GR, 2015. Coalition formation and task allocation of multiple autonomous robots. Proc 3rd Int Conf on Signal Processing, Communication and Networking, p.1-5.
https://doi.org/10.1109/ICSCN.2015.7219891

Pardalos PM, Grundel D, Murphey R, et al., 2008. Cooperative Networks: Control and Optimization. Edward Elgar Publishing, Cheltenham, UK.

Ramchurn SD, Polukarov M, Farinelli A, et al., 2010. Coalition formation with spatial and temporal constraints. Proc 9th Int Conf on Autonomous Agents and Multiagent Systems, p.1181-1188.
https://doi.org/10.5555/1838186.1838191

Shehory O, Kraus S, 1998. Methods for task allocation via agent coalition formation. *Artif Intell*, 101(1-2):165-200. https://doi.org/10.1016/s0004-3702(98)00045-9

Singh VK, Husaini S, Singh A, 2010. Self-organizing agent coalitions in distributed multi-agent systems. Proc Int Conf on Computational Intelligence and Communication Networks, p.650-655.
https://doi.org/10.1109/CICN.2010.128

Skorobogatov G, Barrado C, Salamí E, 2020. Multiple UAV systems: a survey. *Unmanned Syst*, 8(2):149-169.
https://doi.org/10.1142/S2301385020500090

Vig L, Adams JA, 2006. Multi-robot coalition formation. *IEEE Trans Robot*, 22(4):637-649.
https://doi.org/10.1109/TRO.2006.878948

Yang Y, Qiu XS, Meng LM, et al., 2014. Task coalition formation and self-adjustment in the wireless sensor networks. *Int J Commun Syst*, 27(10):2241-2254.
https://doi.org/10.1002/dac.2470