TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 14/16 pp2566–2582
DOI: 10.26599/TST.2024.9010262
Volume 31, Number 5, October 2026

Research Article

Collaborative Risk-Avoidance Strategy for Multirobot Information Gathering in Adversarial Environment

Kai Di, Pan Li*, Tienyu Zuo, Fulin Chen, Yuanshuang Jiang, Yichuan Jiang, and Dan Chen

Abstract: Multirobot information gathering in adversarial environments, where enemies can destroy detected robots, presents unique challenges not addressed by existing algorithms designed for safe settings. The advent of multirobot systems enables collaborative risk-avoidance behavior, exemplified in pursuit-evasion scenarios, where concentrated robot groups enhance their ability to evade and overcome pursuing enemies. However, current multirobot information gathering algorithms have not yet integrated this collaborative risk-avoidance model, potentially leading to robot damage and reduced efficiency. This study adopts a decomposition-and-grafting mechanism to separate the problem into two weakly coupled subproblems: task execution and task allocation. For task execution, we propose an exact algorithm based on the branch-and-pricing method. Our task execution algorithms are seamlessly integrated with a novel task map composition algorithm designed to identify high-utility solutions with minimal computational overhead, addressing the task allocation problem. Extensive simulations demonstrate that our algorithms significantly outperform benchmarks by increasing the number of collaborative risk-avoidance activities conducted by robots during information gathering tasks in adversarial environments. This research advances multirobot information gathering by incorporating collaborative risk-avoidance, enhancing robot survivability and efficiency in hazardous settings.

Key words: multirobot information gathering; robotics in adversarial environment; path planning; multiagent system; multiagent collaboration

1 Introduction

Multirobot information gathering, a common task in many real-world robotic applications[1–3], has traditionally been studied in the context of safe environments devoid of adversarial elements[4, 5]. However, as the field of robotics advances, robots are increasingly deployed in adversarial environments[1, 6], where they face the risk of destruction by enemies. Therefore, the focus of robotic information gathering is shifting towards risk-avoidance behavior to ensure the safe retrieval of collected data.

Recent research has demonstrated that deploying multirobot teams can enhance the risk-avoidance

• Kai Di and Pan Li are with the Hangzhou School of Automation, Zhejiang Normal University, Hongzhou 311231, China. E-mail: dikai@seu.edu.cn; lipan@seu.edu.cn.
• Tienyu Zuo, Yuanshuang Jiang, and Yichuan Jiang are with School of Computer Science and Engineering, Southeast University, Nanjing 211189, China. E-mail: tienyu.zuo@seu.edu.cn; yuanshuangjiang@seu.edu.cn; yjiang@seu.edu.cn.
• Fulin Chen is with School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China. E-mail: chenfulin@seu.edu.cn.
• Dan Chen is with College of Electronics and Information Engineering, Beibu Gulf University, Qinzhou 535000, China. E-mail: cd@bbgu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2024-10-25; revised: 2024-12-01; accepted: 2024-12-26

© The author(s) 2026. The articles published in this open access journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/).

capabilities of individual robots[7, 8]. As the number of robots working together in an adversarial environment increases, so does the probability of successfully overcoming enemy threats[9−12]. Leveraging this characteristic[9, 10], we introduce a collaborative risk-avoidance model to optimize the frequency of collaborative risk-avoidance activities performed by robots during information gathering tasks, ultimately improving their ability to defend against adversaries.

Traditional multirobot information gathering strategies, often designed for safe environments, are not well-suited for adversarial settings, where the collaborative risk-avoidance model is crucial. Incorporating this model presents two main challenges.

(1) Curse of structure: The collaborative risk-avoidance model introduces new constraints to traditional problems, requiring a balance between the completion cost of information gathering tasks and the utility gained from collaborative risk-avoidance activities. This balance complicates the structure of the solution space.

(2) Curse of dimensionality: The strategies adopted by robots for task execution and allocation impact both the final utility and the feasibility of successfully traversing assigned paths, thereby increasing the dimensionality of the solution space.

To address these challenges systematically, a decomposition-and-grafting mechanism[13] is adopted to decompose this problem into two weakly coupled problems, including the information gathering task execution problem and the information gathering task allocation problem.

For the information gathering task execution problem, an exact algorithm is designed to yield optimal solutions. This algorithm embeds the column generation algorithm[14−16] into the framework of the branch-and-bound algorithm[17, 18] and uses it to calculate the lower bound of the subproblem to improve the convergence speed[19, 20].

For the information gathering task allocation problem, we propose a task map composition (TMC) algorithm inspired by the iterated greedy approach. This algorithm utilizes the minimum spanning tree (MST) algorithm[21, 22] to estimate the cooperative risk-avoidance gain from different task assignment strategies. It then greedily assigns the tasks that can currently yield the highest cooperative risk-avoidance utility.

The effectiveness and efficiency of the proposed algorithms are validated through rigorous theoretical analyses and extensive experiments. By comparing with classic benchmark algorithms[23−26], there does not appear to be a statistical difference between the CPLEX (the optimal approach) and our proposed algorithms. However, our algorithms demonstrate a substantial reduction in execution time compared to CPLEX.

The main intellectual contributions of this paper are summarized as follows:

(1) It is the first work, to the best of our knowledge, that develops a cooperative risk-avoidance model for the multirobot information gathering problem, which extends the previous basic form of the multirobot information gathering problem.

(2) A decomposition-and-grafting mechanism[13] is adopted in this paper to decompose the problem into two weakly coupled subproblems, including the task execution problem and the task allocation problem. For the task execution problem, we design an exact algorithm based on the branch-and-pricing method. For the task allocation problem, we design an iterated greedy algorithm with polynomial complexity.

(3) This paper performs both rigorous theoretical analyses and simulation results, demonstrating that our algorithms achieve comparable effectiveness and efficiency to benchmark multirobot algorithms[23−26] across various parameter settings.

2 Related Work

The recent growth in demand and complexity of the multiagent information gathering systems has led to the development of different approaches for an optimized information gathering system. In this section, we survey relevant academic and practical approaches for such systems.

2.1 Multirobot information gathering system

The mechanisms learned from biological systems have been applied to design the behavior of robots in multirobot information gathering systems[27−35]. This type of research can be regarded as a mechanism-to-behavior study[36], as distinct from the behavior-to-mechanism study mentioned previously.

Among these mechanisms extracted from the swarm behavior emerged in biological systems, the “conflict-free” mechanism is the most widely used. Utilizing this mechanism, a multirobot information gathering system was constructed[37, 38]. In such system, no conflict will

occur during the information gathering process, improving the efficiency of multirobot information gathering activity. Exploiting this mechanism, information gathering strategies for unmanned aerial vehicles (UAVs) were designed, where robots are UAVs and the items fetched back are different sources of information[39, 40].

However, the goal of being "conflict-free" is quite different from the goal of performing collaborative risk-avoidance behavior studied in this paper. More specifically, the phenomena where robots are in the same area at the same time can improve the success rate of avoiding danger when detected by the adversary in environments[41–43]. Combined with the above analysis, the traditional study of multirobot information gathering problems is not applicable to adversarial environments, where collaborative risk-avoidance behavior needs to be considered.

In addition to the differences in the form of path planning mentioned above, the introduction of collaborative risk-avoidance behavior in adversarial environments also results in different forms of utility gained from information gathering activities. In the traditional multirobot information gathering problem, the corresponding utility gained by robots visiting information gathering locations in environments is usually fixed and does not change over time[37, 38]. Unlike the traditional multirobot information gathering problem, the utility gained by performing collaborative risk-avoidance behavior with other robots at a information gathering location is different and time-varying. The reason for this phenomenon is that robots may perform collaborative risk-avoidance behavior with different numbers of robots when they visit a information gathering location at different times, resulting in different utilities.

In summary, the traditional multirobot information gathering algorithms[37, 38] are not applicable to the multirobot information gathering problem considering the collaborative risk-avoidance behavior due to the different forms of revenue calculation and path planning. In response to this situation, this paper introduces the path planning constraints brought by cooperative risk-avoidance behavior into the traditional multirobot information gathering problem and systematically investigates cooperative risk-avoidance strategies for multirobot information gathering systems.

2.2 Collaborative risk-avoidance behavior of robot in adversarial environment

Designing collaborative risk-avoidance behavior for robots is an important issue that needs to be considered seriously when adopting multirobot systems in adversarial environments. The most representative study in this area is the multirobot pursuit-evasion problem, and we focus on this area of research in Section 2.2.

In recent years, researchers have investigated the multirobot pursuit-evasion problem from different perspectives, including the optimization time as well as the optimization success rate. For the former, Stiffler and O'Kane[44] investigated how to maximize the time spent in pursuit-evasion process, i.e., maximizing the survival time of evaders. For the latter, Pierson et al.[45] examined how to maximize the evasion success rate for the evaders.

Through a series of studies on the multirobot pursuit-evasion problems, researchers have noted that as the number of evaders increases, the efficiency of evasion increases. More specifically, the time it takes for the pursuers to catch evaders is greatly increased[45] and the success rate of the evasion is also greatly improved[45, 46]. Researchers have conducted in-depth studies on this phenomenon from the perspective of the environment, in which the multirobot pursuit-evasion event takes place, and the results confirm that the above conclusions hold true whether in two-dimensional environments, three-dimensional environments, or flow fields.

Existing studies on multirobot risk-avoidance behavior neglect to exploit the mentioned feature, missing the opportunity to improve the success rate of risk-avoidance activity. For example, authors in Refs. [47, 48] assumed that the number of evaders and pursuers is fixed before the start of the entire activity, ignoring the optimization of evaders involving in the risk-avoidance activity. Chen et al.[49] studied situations where the number of evaders is known in advance and will not change. In addition, Sun and Tsiotras[50] also ignored the improvement in the risk-avoidance success rate brought by the optimization of the number of robots involving in the risk-avoidance activity.

To sum up, this series of studies shows that the number of robots participating in collaborative risk-avoidance activity is closely related to the final risk-

avoidance efficiency. At this stage, however, there is a lack of methods to optimize the number of robots involved in the risk-avoidance activity. To fill this gap, this paper studies how to involve more robots in collaborative risk-avoidance activity during the information gathering process, with the aim of increasing the success rate of risk-avoidance behavior when detected by the adversary.

3 Problem Formulation and Analysis

To illustrate the research problem in this paper clearly, we first introduce the motivation and problem settings of multirobot information gathering problem considering the collaborative risk-avoidance behavior in Section 3.1. Next, we present the formal formulation of this problem in Section 3.2. Finally, we analyze the complexity of the problem in Section 3.3.

3.1 Problem statement for the cooperative risk-avoidance problem in multirobot system

Designing a collaborative risk-avoidance strategy for a team of robots in adversarial environments adds additional constraints to the basic model of multiagent collaboration problem.

Environment model. In the initial stage, robots need to depart from the home location to information gathering locations scattered in different areas $\mathcal{O}=\{O_1,O_2,\ldots,O_l\}$ of the environment G, where l represents the number of areas in G, as shown in Fig. 1. Considering a model for the environment where robots perform cooperative risk-avoidance behaviors, we characterize it by a fully connected undirected graph $G=\langle V,E\rangle$. Let $V=\{v_0\}\cup V_e$ be the set of locations in the environment, v_0 is the starting point (return point) of the robots, and $V_e=\{v_1,v_2,\ldots,v_m\}$ is the set of information gathering locations in the environment, where m represents the number of information gathering locations[51]. Let $E\subseteq V\times V$ be the set of edges between all locations in the environment. Each edge $e_{ij}\in E$ is associated with a travel time cost $c_{ij}\in\mathbb{R}^+$ between locations v_i and v_j[52].

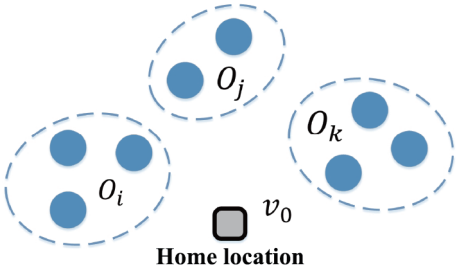


Fig. 1 Environment model.

Robot model. Let $A=\{a_1,a_2,\ldots,a_n\}$ denote the set of robots, where n represents the number of robots, and let δ denote the time cost needed to perform any information gathering task. Robot $a_i\in A$ departs from home location v_0 to the k-th information gathering location $v_k\in V_e$ to perform information gathering tasks following the predetermined information gathering strategy[53, 54]. Due to the existence of the power limit, robots need to return to home location v_0 within $t_{\max}$ time steps after departing from this point, and then the whole information gathering process ends[55].

Collaborative risk-avoidance behavior. During the execution of information gathering tasks, some robots may appear in the same area. Previous studies have shown that in adversarial environments, the collaborative risk-avoidance ability of robots will be enhanced with the increasing number of robots in the same area[9–11]. The issue where robots perform information gathering tasks in the same area in a certain time period is defined as "collaborative risk-avoidance behavior" in this paper. As shown in Fig. 2, robots a_i, a_j, and a_k carry out collaborative risk-avoidance behavior when they perform information gathering tasks at v_m, v_n, and v_l in the same area during a certain time period, and their collaborative risk-avoidance ability can be then enhanced[9–11].

Information model. This paper discusses the case where the information about information gathering strategies adopted by each robot is known by all other robots in a multirobot team. The reason for this case is that all information gathering strategies are designed prior to task execution and shared in real-time via global communication[56–59]. Meanwhile, the focus of studying this case is how to use such accurate information to make more efficient decisions[60, 61]. The opposite case where information is mutually unknown in the multirobot team is also common in real-world

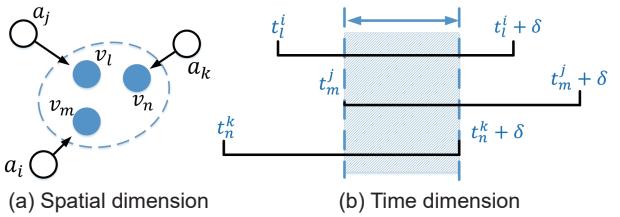


Fig. 2 Simple case to illustrate the cooperative risk-avoidance behavior of robot. The determination of a collaborative risk-avoidance behavior needs to consider two dimensions, including the spatial dimension and the time dimension.

applications, it has a different emphasis on how to infer information during performing tasks[62]. Both cases are significant, and we will study unknown cases in future work.

Optimization objective. The objective of the multirobot team A is to maximize the number of collaborative risk-avoidance activities performed while executing information gathering strategies (subsequently referred to as "collaborative risk-avoidance utility"). To provide a formal formulation of the objective function, let us consider a information gathering strategy π_i adopted by robot $a_i \in A$, $\pi_i = \{v_0, e_{01}, v_1, e_{12}, ..., v_j, e_{jk}, v_k, e_{k0}, v_0\}$[63, 64]. We say that robot $a_i \in A$ performs a collaborative risk-avoidance behavior with robot $a_j \in A \setminus \{a_i\}$ if a_i and a_j perform information gathering tasks in the same area during a certain time period[9–12]. The associated collaborative risk-avoidance utility is calculated as

$$r_{kh}^{ij} = \begin{cases} \alpha_{kh}, \max\left\{t_k^i, t_h^j\right\} \leqslant \min\left\{t_k^i + \delta, t_h^j + \delta\right\}; \\ 0, \quad \text{otherwise} \end{cases} \tag{1}$$

where t_k^i and t_h^j are start times for robots a_i and $a_j \in A$ to perform information gathering tasks at the k-th and the h-th information gathering locations v_k and $v_h \in V_e$, respectively. The Boolean variable is used to indicate whether information gathering locations v_k and v_h are in the same area, if the proposition holds then $\alpha_{kh} = 1$ (i.e., $\exists O_l \in \mathcal{O}, v_k, v_h \in O_l$), otherwise $\alpha_{kh} = 0$. Given a set of information gathering strategies $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$, the collaborative risk-avoidance utility gained by performing Π is calculated as

$$U = \sum_{i \in A} \sum_{j \in A \setminus \{i\}} \sum_{k \in V_e} \sum_{h \in V_e \setminus \{k\}} r_{kh}^{ij} \tag{2}$$

To enable further analysis of this problem, we formulate the definition of the collaborative risk-avoidance problem (CRAP) as follows:

Definition 1 Collaborative risk-avoidance problem Given an environment represented by an undirected complete graph G, find a information gathering strategy for the multirobot team A with the maximum collaborative risk-avoidance utility.

3.2 Problem formulation for CRAP in multirobot system

To provide a mathematical formulation of the problem, we first introduce the following decision variables.

- For each robot $a_i \in A$ and each edge $e_{kh} \in E$, the binary decision variable $x_{kh}^i = 1$ indicates whether robot a_i passes edge e_{kh} in its information gathering path.
- For each robot $a_i \in A$ and each information gathering location $v_k \in V$, the binary decision variable $y_k^i = 1$ indicates whether robot a_i passes information gathering location v_k in its information gathering path.

The CRAP can be then formulated as follows:

$$\max \sum_{i \in A} \sum_{j \in A \setminus \{i\}} \sum_{k \in V_e} \sum_{h \in V_e \setminus \{k\}} r_{kh}^{ij} \tag{3}$$

$$\text{s.t.,} \sum_{k \in V_e} x_{0k}^i = 1, \forall a_i \in A \tag{4}$$

$$\sum_{k \in V_e} x_{k0}^i = 1, \forall a_i \in A \tag{5}$$

$$\sum_{h \in V} x_{kh}^i = \sum_{l \in V} x_{kl}^i = y_k^i, \ \forall a_i \in A, \forall v_k \in V_e \tag{6}$$

$$\sum_{i \in A} y_k^i = 1, \ \forall v_k \in V_e \tag{7}$$

$$t_0^i \geqslant 0, \ \forall a_i \in A \tag{8}$$

$$t_k^i + \delta + c_{k0} \leqslant t_0^k + t_{\max}, \ \forall a_i \in A, \forall v_k \in V_e \tag{9}$$

$$t_k^i + \delta + c_{kh} - M\left(1 - x_{kh}^i\right) \leqslant t_h^i, \ \forall a_i \in A, \ \forall v_k, v_h \in V_e \tag{10}$$

$$r_{kh}^{ij} \leqslant \alpha_{kh}, \ \forall a_i, a_j \in A, \forall v_k, v_h \in V \tag{11}$$

$$r_{kh}^{ij} \cdot t_k^i \leqslant t_h^j + \delta, \ \forall a_i, a_j \in A, \forall v_k, v_h \in V \tag{12}$$

$$M\left(1 - r_{kh}^{ij}\right) + t_k^i + \delta \geqslant t_j^i, \ \forall a_i \in A, \forall v_k, v_h \in V_e \tag{13}$$

$$x_{kh}^i \in \{0, 1\}, \ \forall a_i \in A, \forall v_k, v_h \in V \tag{14}$$

$$y_k^i \in \{0, 1\}, \ \forall a_i \in A, \forall v_k \in V \tag{15}$$

where M is a large integer, δ is the duration of each information gathering task, and c_{kh} is the travel time cost between locations v_k and v_h.

The constraints in Formulas (4)–(15) of CRAP can be classified into three categories as follows:

- **Flow constraint.** Constraints in Formulas (4)–(7) are flow constraints that describe whether information gathering strategy $\forall \pi_i \in \Pi$ is feasible[65]. Constraints in Formulas (4) and (5) require that the resulted information gathering path must be a self-loop (i.e., robots perform information gathering tasks from home location v_0 and return to v_0 after completing all assigned information gathering tasks)[66, 67]. Meanwhile, constraints in Formulas (6) and (7) require that robot $a_i \in A$ will leave from information gathering

location $v_k \in V_e$ after completing information gathering tasks at v_k, and information gathering tasks in $\forall v_k \in V_e$ can be performed once and only once by robots[4, 5].

- **Time constraint.** Constraints in Formulas (8)–(10) are time constraints used to represent the task start time, task execution time, task completion time, etc. Constraint in Formula (8) represents the task start time t_0^i for robot $a_i \in A$, it is worth noting that t_k^i is the time step when robot $a_i \in A$ arrives at information gathering location $v_k \in V_e$[68–70]. Constraint in Formula (9) requires that robots must return to home location v_0 within $t_{\max}$ time steps from the start of performing information gathering tasks[62]. Constraint in Formula (10) represents the start time of performing the information gathering task in $v_h \in V_e$ after leaving from $v_k \in V_e$[71, 72].
- **Cooperative risk-avoidance constraint.** Constraints in Formulas (11)–(13) are cooperative risk-avoidance constraints proposed to determine the cooperative risk-avoidance behavior. Constraint in Formula (11) presents the cooperative risk-avoidance utility $r_{kh}^{ij} = \{0,1\}$ gained by robots a_i and $a_j \in A$ in v_k and $v_h \in V_e$, where α_{kh} is a Boolean variable used to indicate whether information gathering locations v_k and v_h are in the same area[9–11]. Constraints in Formulas (12) and (13) denote the determination of a collaborative risk-avoidance activity. More specifically, if robot $a_i \in A$ performs information gathering tasks in $v_k \in V_e$ when robot $a_j \in A$ also performs information gathering tasks in $v_h \in V_e$ within the same area, then they generate the cooperative risk-avoidance utility and $r_{kh}^{ij} = 1$, otherwise $r_{kh}^{ij} = 0$. The specific determination process is shown in Fig. 3. In Case 1 $t_k^i < t_h^j < t_k^i + \delta$, and in Case 2 $t_h^j < t_k^i < t_h^j + \delta$.

3.3 Property analysis

In this subsection, we discuss the complexity of CRAP.

Theorem1 The CRAP is NP-hard.

Proof This problem is clearly in NP, because one can easily construct a set of feasible information gathering paths for robots and then verify its collaborative risk-avoidance utility in polynomial time[73].

The NP-hardness of CRAP follows from its single-robot version (i.e., $|A| = 1$). To prove its NP-hardness, we use a reduction from the Hamiltonian circuit problem, which has been proven NP-complete[73, 74]. Given any instance of the Hamilton circuit problem, relative to a graph $G^h = \langle V^h, E^h \rangle$ with location set $V^h = \{v_1, v_2, \ldots, v_n\}$ and edge set $E^h = \{(v_i, v_j) : v_i, v_j \in V^h, v_i \neq v_j\}$, we can construct an instance of CRAP as follows.

First, we construct a complete graph $G^h = \langle V^h, E^h \rangle$, where $E^h = \{(v_i, v_j) : v_i, v_j \in V^h, v_i \neq v_j\}$. If $e_{ij} \in E^h$, then the travel time cost function is

$$c_{ij} = \begin{cases} 1, & \forall e_{ij} \in E^h; \\ 2, & \text{otherwise} \end{cases}$$

The "if and only if" direction. Note that there is no self-loop path in G^h, because G^h is an undirected graph. Therefore, this construction can be performed in polynomial time. Then, G^h is a Hamiltonian circuit if and only if the solution utility of this instance equals $|V^h|$. Thus, the CRAP is NP-hard. ■

4 Task Execution Model

In this section, an exact information gathering task execution algorithm is proposed. Considering the case where information gathering locations are scattered sparsely in an adversarial environment, there are few information gathering tasks for robots, resulting in a relatively small scale of problem input. Even if we take an exact algorithm to obtain the optimal solution of the problem, the time consumed is within an acceptable range[75, 76].

To obtain an exact solution for small-scale cases, we address this problem with a column generation algorithm[77, 78] embedded within a branch-and-price (BP) algorithm[14, 15]. A series of acceleration algorithms, such as the internal point stabilization

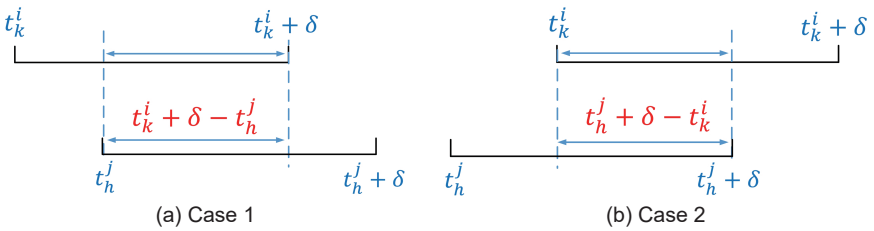


Fig. 3 Determination of cooperative risk-avoidance behavior.

algorithm[16, 17], are then used to accelerate the convergence of the algorithm.

4.1 Column generation

Let us begin with the definition of the master problem (MP) and pricing subproblem.

4.1.1 Master problem

This section introduces a set packing formulation[77, 78] for the master problem, where each column corresponds to a information gathering strategy. The decision variable x_w is a binary variable with value 1 when information gathering strategy w is adopted, 0 otherwise.

$$\min c_w x_w,$$

$$\text{s.t.,} \sum_{w\in\Omega} a_{iw}x_w = 1,\ \ i \in V_{\mathrm{e}} \tag{16}$$

$$\sum_{w\in\Omega} x_w = 1 \tag{17}$$

$$x_w = \{0,1\},\ \ w \in \Omega \tag{18}$$

where Ω is the set of all feasible information gathering strategies. c_w is the total cost of executing information gathering tasks following strategy w, and its value is equal to $\sum_{i\in A}\sum_{j\in A\setminus\{i\}}\sum_{k\in V_{\mathrm{e}}}\sum_{h\in V_{\mathrm{e}}\setminus\{k\}} -r_{kh}^{ij}$. Coefficient $a_{iw}=1$ if the robot visits information gathering location v_i in strategy w, otherwise $a_{iw}=0$.

Among them, Formula (16) corresponds to Eq. (7) in the original problem, which is used to describe that a feasible strategy must include all assigned information gathering tasks. Equation (17) describes the number of feasible strategies that can be selected in the final solution.

4.1.2 Pricing subproblem

Each pricing subproblem is an elementary shortest path problem with resource constraint (ESPPRC) problem[18, 19]. Let us construct the reduced cost (RC)[20] used in the pricing subproblem on the auxiliary graph. If π_i and π_0 are the dual values corresponding to Eqs. (17) and (18), the reduced cost is

$$\mathrm{rc}(w) = c_w - \sum_{i\in V^A}\pi_i y_i^w - \pi_0 = \sum_{i\in V^A}(c_w-\pi_i)y_i^w - \pi_0 \tag{19}$$

To provide a mathematical formulation of the problem, we first introduce the following two decision variables.

- For the feasible strategy $w\in\Omega$ and each edge $e_{ij}\in E^A$, binary decision variable x_{ij}^w indicates whether strategy w includes the edge e_{ij} in the information gathering path.
- For the feasible strategy $w\in\Omega$ and each information gathering location $v_i\in V^A$, binary decision variable y_i^w indicates whether strategy w includes information gathering location v_i in the information gathering path.

Through the definition of decision variables, the pricing subproblem can then be written as follows:

$$\min \sum_{i\in V^A}(c_i-\pi_i)y_i^w - \pi_0,$$

$$\text{s.t.,} \sum_{j\in V} x_{ij}^w = y_i^w,\ \ \forall i\in V^A \tag{20}$$

$$\sum_{i\in V^A} x_{ih}^w = \sum_{j\in V^A} x_{hj}^w,\ \ \forall h\in V^A \tag{21}$$

$$\sum_{i\in V^A} x_{0i}^w = \sum_{i\in V^A} x_{i0}^w = 1 \tag{22}$$

$$t_0^w \geqslant 0 \tag{23}$$

$$t_i^w + \delta + t_{ij}^A - M(1-x_{ij}^w) \leqslant t_j^w,\ \ \forall (i,j)\in E^A \tag{24}$$

$$s_i \leqslant t_i^w \leqslant e_i,\ \ \forall i\in V^A \tag{25}$$

$$x_{ij}^w \in \{0,1\},\ \ \forall (i,j)\in E^A \tag{26}$$

$$y_i^w \in \{0,1\},\ \ \forall i\in V^A \tag{27}$$

where s_i represents the required starting time for visiting the gathering location v_i. Formulas (20)–(22) are flow constraints used to describe the information gathering path of the individual robots. Formulas (23)–(25) are time constraints used to ensure the feasibility of information gathering tasks execution strategy. Formulas (26) and (27) are decision variables.

4.1.3 Column generation algorithm

The column generation algorithm is embedded in a branch-and-price framework[14, 77, 78] that generates the search tree, and the solution space is reduced by forcing a portion of the base variables into non-base variables for solving the lower bound corresponding to nodes of the search tree[15–17]. More specifically, each node corresponds to the optimal solution of the restricted linear master problem (RLMP). The details are shown in Algorithm 1.

- **Step 1:** At the root of the search tree, RLMP is initialized with a set of feasible strategies. The number

of columns thus corresponds to the number of information gathering locations in the environment. For other nodes in the search tree, the algorithm uses the set of the column in the last node considered to initialize the RLMP, after removing columns that are feasible due to branching.

- **Step 2:** Because the number of columns is very large, we gradually add columns to the master problem to construct a series of restricted master problems (RMPs). The linear relaxation of each RMP can be solved by a linear programming solver (e.g., CPLEX, Gurobi).
- **Step 3:** Dual variables corresponding to the current optimal solution of RLMP are used to define a pricing subproblem. By solving this pricing subproblem, a feasible solution with a negative value of reduced cost is determined.
- **Step 4:** Next, these feasible solutions are added to the current RLMP to obtain the next RLMP, which is then solved to obtain a new dual variable. This iterative process will be repeated until there is no solution with a negative reduced cost. So far, we can find an optimal solution for RLMP.

The above process is the specific operation process of the column generation algorithm. In practice, the column generation algorithm has a disadvantage, where many iterations are often needed to prove optimality, while the optimal value of the RLMP does not change significantly from one iteration to the next. This happens especially when the master problem shows some degrees of degeneracy. Therefore, many researchers have proposed stabilization techniques to speed up the convergence speed of the column generation algorithm.

Here, the interior-point stabilization technique proposed in Refs. [16, 17] is used. This technique exploits the solutions taken inside the optimal dual polyhedron, rather than extreme points, because their dual value tends to better approximate the true test number. In practice, several dual solutions are generated inside the polyhedron and a convex combination of dual values is used within the pricing subproblem.

4.2 Branch-and-price algorithm

Column generation is applied at every node of a search tree generated by the branch-and-price algorithm. Let us describe our search and branch strategies in this section.

Algorithm 1 Column generation algorithm

1. Generate an initial set of columns Ω_1
2. **while** intermediate variable $\Gamma \neq \varnothing$ **do**
3. Solve MP(Ω_1)
4. $\Gamma \leftarrow$ column(s) provided by the pricing subproblem
5. $\Omega_1 \leftarrow \Omega_1 \bigcup \Gamma$
6. **End while**

4.2.1 Search strategy

The search tree is explored according to a best-first policy[79, 80], where subproblems are ranked according to the value of their lower bound. The reason why best-first is chosen over depth-first[81, 82] is that the latter yields inferior results in preliminary experiments.

4.2.2 Branch strategy

The design of the branch strategy is the core of the branch-and-price algorithm. An effective branch strategy should cut off the decimal solution while keeping the structure of the problem unchanged, and generate a relatively balanced search tree at the same time[83, 84]. We use the commonly employed branching strategy based on adapted Ryan-Foster rules (e.g., see Ref. [85]). We look for two fractional routes sharing the same pair of nodes v_i and v_j but with one route using the edge e_{ij} and not the other. It can be shown that such routes can always be found. In each branch, the shortest-path subproblem can be modified easily (simply by removing edges from the underlying graph) without breaking the structure of the problem.

4.2.3 Analysis of algorithm

Next, we prove that the proposed branch-and-price algorithm is exact, i.e., it yields the optimal solution.

Theorem 2. The branch-and-price algorithm is optimal.

Proof To achieve the optimum, the branch-and-price algorithm should consider all possible states of the problem and terminate in finite time[75, 76]. ■

5 Task Allocation Model

Through the decomposition-and-grafting mechanism[13], we decompose CRAP into the information gathering task execution subproblem and the information gathering task allocation subproblem. Through the above research, we obtain the information gathering task execution algorithms for individual robots at different scales. In order to graft the results of the above algorithms, this section proposes an efficient algorithm called task map composition algorithm Algorithm 2.

Algorithm 2 Task map composition algorithm

1. $\mathcal{S} \leftarrow \varnothing$;
2. max $\leftarrow 0$
3. **for** $i = 1 \rightarrow |V|$ **do**
4. $\mathcal{S} \leftarrow \mathcal{S} \cup \{v_i\}$;
5. **end for**
6. $\mathcal{T} \leftarrow \mathcal{S}$;
7. **while** $|\mathcal{S}| > 1$ **do**
8. **for** $i \leqslant |\mathcal{S}|$ **do**
9. **for** $j \leqslant i$ **do**
10. temp $\leftarrow$ GAIN$(\mathcal{M}_i, \mathcal{M}_j)$;
11. **if** temp > max **then**
12. max $\leftarrow$ temp;
13. $\mathcal{M}_i^{\star} \leftarrow \mathcal{M}_i$;
14. $\mathcal{M}_j^{\star} \leftarrow \mathcal{M}_j$;
15. **end if**
16. **end for**
17. **end for**
18. $\mathcal{T} \leftarrow \mathcal{T} \cup \{\mathcal{M}_i^{\star}, \mathcal{M}_j^{\star}\}$;
19. $\mathcal{S} \leftarrow (\mathcal{S} \setminus \{\mathcal{M}_i^{\star}\}) \setminus \{\mathcal{M}_j^{\star}\}$;
20. $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathcal{M}_i^{\star}, \mathcal{M}_j^{\star}\}$;
21. $\mathcal{T} \leftarrow \mathcal{S}$;
22. **end while**
23. $\mathcal{S} \leftarrow$ SELECT$(\mathcal{T})$;
24. **return** $\mathcal{S}$

5.1 Efficient task map composition algorithm

In this section, a heuristic algorithm named task map composition is proposed to assign information gathering tasks to multi-robot teams, with the objective of improving the collaborative risk-avoidance utility. The task map refers to a graph structure composed of information gathering tasks assigned to a robot and its task execution strategy.

During the execution of TMC algorithm, we need to calculate the gain in collaborative risk-avoidance utility from composing two task maps, and the exact gain is calculated as

$$\text{GAIN}^{\star}(\mathcal{M}_i, \mathcal{M}_j) = V^{\star}(\mathcal{M}_i \cup \mathcal{M}_j) - V^{\star}(\mathcal{M}_i) - V^{\star}(\mathcal{M}_j) \tag{28}$$

where $V^{\star}$ indicates the exact gain function.

According to the rigorous theoretical analysis proposed in Theorem 1, thus calculating the exact gain in collaborative risk-avoidance utility is very time-consuming[13]. In order to speed up the calculation process, we use the minimum spanning tree algorithm[21, 22] to approximate the utility collected by the path formed by points in clusters. Using this approximation method, we can compute our approximate gain for composing two task maps

$$\text{GAIN}(\mathcal{M}_i, \mathcal{M}_j) = \tilde{V}(\mathcal{M}_i \cup \mathcal{M}_j) - \tilde{V}(\mathcal{M}_i) - \tilde{V}(\mathcal{M}_j) \tag{29}$$

where $\tilde{V}$ indicates the approximate gain function.

The pseudo-code of the TMC algorithm is then presented in Algorithm 2, where $\mathcal{S}$, max, $\mathcal{T}$, temp, $\mathcal{M}_i$, $\mathcal{M}_j$, $\mathcal{M}_i^{\star}$, and $\mathcal{M}_j^{\star}$ are intermediate variables, SELECT represents selecting strategy associated with the maximum value. The details for the TMC algorithm are as follows:

- **Step 1 (Lines 1–6):** Initialize a set of task maps and construct the corresponding tree structure, dendrogram, and data structure to store the process of task map composition. More specifically, the dendrogram is stored as a sequence of compositions which are pointers in practice.
- **Step 2 (Lines 7–23):** Task maps are composed iteratively, forming a tree structure until all information gathering tasks are composed within a single task map. In the process of composition, the corresponding tree structure and its attributes are updated in real-time.
- **Step 3 (Line 24):** Task maps are selected from the resulted tree structure by returning task maps with the maximum collaborative risk-avoidance utility.

5.2 Analysis of algorithm

In order to demonstrate the theoretical properties of the algorithm, a series of rigorous theoretical analyses of TMC algorithm are performed in this section.

5.2.1 Completeness of TMC algorithm

The completeness property is used to describe solutions computed by an algorithm that guarantees to complete all the assigned tasks. Let us begin by proving the completeness of TMC algorithm with Theorem 3.

Theorem 3 Completeness Task map composition algorithm generates that all information gathering tasks are included in the information gathering strategy it outputted.

Proof TMC algorithm composes two task maps together in Line 10 of the procedure. To speed up the process of calculation in the mentioned process, TMC algorithm adopts the minimum spanning tree algorithm to construct the composed task map and approximate its associated collaborative risk-avoidance utility. On the one hand, the completeness of MST algorithm is proven in Refs. [21, 22]. On the other hand, it can be

learned from Section 3 that all locations in V_e are connected to home location v_0 in such an adversarial environment G. Thus, TMC algorithm results a task allocation strategy that includes all information gathering locations in its output. ■

5.2.2 Runtime complexity of TMC algorithm

To prove that TMC can find a task allocation strategy within a polynomial time complexity, we will discuss the runtime complexity of TMC.

Theorem 4 TMC algorithm returns a task allocation strategy $\mathcal{S}$ in $\mathcal{O}(n^4\log n)$ time, where n is the number of information gathering tasks given as an input to TMC algorithm.

Proof There are three main processes in TMC algorithm: the initialization process, the composition process, and the selection process. The time complexity of these processes is discussed as follows:

Initialization process. The initialization process is performed in Lines 1−6 of Algorithm 2. During this process, TMC algorithm initializes a set of task maps and constructs the corresponding tree structure, dendrogram, and data structure to store the process of task map composition. More specifically, the dendrogram is stored as a sequence of compositions which are pointers in practice. The runtime complexity of the initialization process is $\mathcal{O}(n)$.

Composition process. The composition process is performed in Lines 7−23 of Algorithm 2. During this process, TMC algorithm runs the progressive task map composition procedure on the current task maps with $\mathcal{O}(n^3)$ runtime complexity. To calculate the gain in collaborative risk-avoidance utility by composing two task maps, TMC algorithm runs the MST procedure for each pair of task maps with $\mathcal{O}(n\log n)$ runtime complexity. The runtime complexity of the composition can be then calculated as $\mathcal{O}(n^4\log n)$.

Selection process. The selection process is performed in Line 23 of Algorithm 2. During this process, TMC algorithm selects a set of task maps with the maximum collaborative risk-avoidance utility from the constructed tree structure. The runtime complexity of the initialization process is $\mathcal{O}(n)$.

Hence, the runtime complexity of TMC is $\mathcal{O}(n^4\log n)$. ■

6 Experiment

In this section, a series of experiments are carried out to test the proposed algorithms, and the effectiveness and the efficiency of the proposed algorithms are verified by comparison with classical benchmark.

6.1 Experimental setting

The experiments are implemented in Java 11 and tested on an Intel (R) Core (TM) CPU i7-4770 at 3.4 GHz and with 16 GB memory. All results are averaged over 5000 instances. Each record is statistically significant at a 95% confidence level.

6.2 Benchmark algorithm

In experiments, we compare our proposed algorithms against these benchmarks: the graph search (GS) algorithm, the weighted nearest neighbor (WNN) algorithm, the region-based path planning (RPP) algorithm, the adaptive insertion (ADI) algorithm, and the optimal result Cpl calculated by solver CPLEX. We outline these benchmarks as follows:

- The graph search algorithm is an efficient method to solve the path planning problem for individual agents[23], the idea of which is to use a total ordering of vertices based on the calculated cost-to-arrival to find an optimal path between defined start and goal vertices.
- The weighted nearest neighbor algorithm is a conventional approach that has been used in previous path planning problems for individual agents[26]. When following this approach, robots will complete all assigned information gathering tasks with the nearest neighbor algorithm within the minimum time, and then use the remaining time to perform as many collaborative risk-avoidance activities as possible.
- The region-based path planning algorithm is an efficient approach that has been used in previous path planning problems for multirobot teams[24]. In this approach, the entire environment is first split into a certain number of parts based on the utility gained by visiting different vertices. Robots then perform tasks separately along the cyclic path formed by the vertices in order of the highest to lowest utility.
- The adaptive insertion algorithm is a brand-new model adopted to solve the multiagent path planning problem[25]. The aim is to minimize the maximum time spent by agents on their assigned tasks, which implies a load-balancing task allocation concept.
- Cpl is the result returned by solving the CRAP with the optimal solver CPLEX (version 12.6).

6.3 Optimization objective

By considering the optimization objective of this study,

we define the following indices to evaluate the performance of different approaches.

- **Collaborative risk-avoidance utility.** This index is introduced to evaluate the proposed optimization objective of maximizing the expected utility gained by following the given information gathering strategy. The calculation process of this index is shown in Eq. (2).
- **Collaborative risk-avoidance proportion.** This index is also introduced to measure the solution quality of information gathering strategies in another dimension. Let p_i denote the ratio of collaborative risk-avoidance utility gained by a given information gathering strategy to that gained by the optimal information gathering strategy. Given a total of N test instances, the collaborative risk-avoidance proportion is calculated as $\sum_{i=1}^{N} p_i/N$.
- **Robustness.** This index is introduced to measure the robustness of proposed algorithms in different settings, including different numbers of information gathering tasks V_e, different values of maximum working time $t_{\max}$, and different numbers of robots $|A|$.
- **Running time of algorithm.** This index is introduced to measure the scalability and computational efficiency of our proposed algorithms and other benchmark algorithms.

6.4 Test of the influence of the number of robots

In this section, we test the effectiveness of our proposed TMC+BP algorithm and three other benchmark algorithms for different numbers of robots. Among these algorithms, Cpl is the result returned by solving the CRAP with the optimal solver CPLEX (version 12.6).

Parameter setting. The value of the maximum working time $t_{\max}$ of robots is set to 10, and the number of information gathering tasks is set to 15. By varying the number of robots from 2 to 6, we obtain a line graph of the collaborative risk-avoidance utility gained by adapting the TMC+BP algorithm and the other three benchmark algorithms, as shown in Fig. 4.

Phenomenon. As the number of robots increases, the collaborative risk-avoidance utilities gained by following all these algorithms show an upward trend. Furthermore, the TMC+BP algorithm outperforms the other benchmark algorithms (i.e., RPP and ADI), and the performance of the TMC+BP algorithm is close to the optimal solution Cpl.

Reason. Due to the increase in the number of robots, both the total amount of travel time for each robot and the total amount of time for task execution decrease. Meanwhile, the time interval that robots spend in waiting at each information gathering location increases, and thus the collaborative risk-avoidance utility tends to increase. Since the ADI algorithm considers the number of information gathering tasks to be equally allocated among robots, the collaborative risk-avoidance utility does not change much with the number of robots.

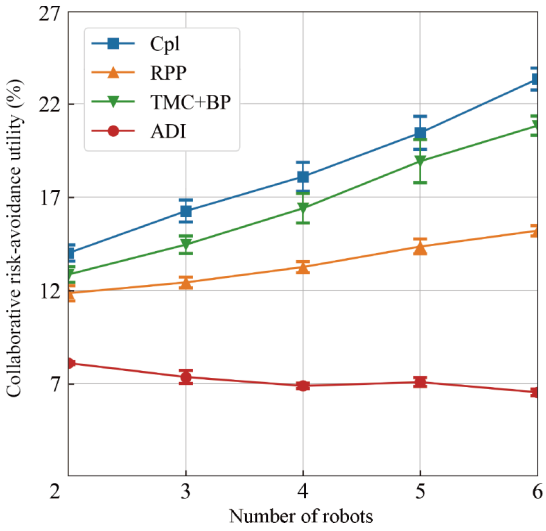


Fig. 4 Test of the influence of the number of robots.

6.5 Test of the influence of the maximum working time

In this section, we test the effectiveness of our proposed TMC+BP algorithm and the other three benchmark algorithms for different values of $t_{\max}$. Among these algorithms, Cpl is the result returned by solving CRAP with the optimal solver CPLEX (version 12.6).

Parameter setting. The number of robots is set to 3, and the number of information gathering tasks is set to 15. By varying the maximum working time from 6 to 10, we obtain a line graph of the collaborative risk-avoidance utility gained by adapting the TMC+BP algorithm and the other three benchmark algorithms, as shown in Fig. 5.

Phenomenon. As the value of $t_{\max}$ increases, the collaborative risk-avoidance utilities gained by following all these algorithms show an upward trend. Furthermore, the collaborative risk-avoidance utilities obtained by the exact algorithm Cpl and the TMC+BP algorithm outperform the other benchmark algorithms

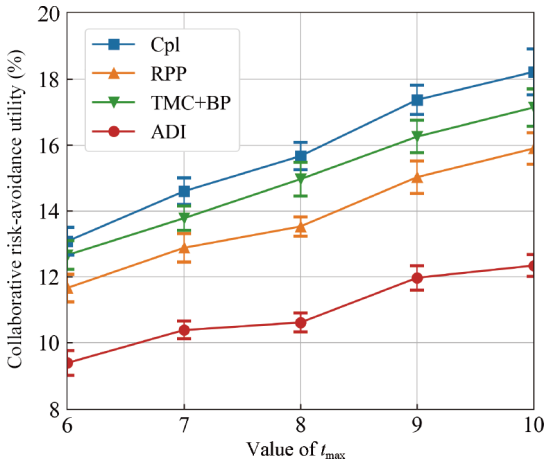


Fig. 5 Test of the influence of the maximum working time.

(i.e., RPP and ADI), and the collaborative risk-avoidance utility gained by following the TMC+BP algorithm is close to the optimal solution Cpl. The phenomena are similar under different datasets, and we will not repeat them here.

Reason. As t_{max} increases, robots have more time to wait at some information gathering locations for a greater collaborative risk-avoidance utility, and thus the resulted collaborative risk-avoidance utilities gained by following all these algorithms tend to increase.

6.6 Test of the collaborative risk-avoidance proportion

In this section, we test the effectiveness of our proposed TMC+BP algorithm and three other benchmark algorithms under different settings. Among these algorithms, Cpl is the result returned by solving the CRAP with the optimal solver CPLEX (version 12.6).

Parameter setting. In Case I, the value of the maximum working time t_{max} of robots is set to 10, and the number of information gathering tasks is set to 15. In Case II, the number of robots is set to 3, and the number of information gathering tasks is set to 15.

Phenomenon. As shown in Fig. 6, for Case I, the collaborative risk-avoidance proportions gained by following all these algorithms show a downward trend with the growth in the number of robots. As for Case II, the collaborative risk-avoidance proportions gained by following all these algorithms show an upward trend with the growth in the value of t_{max}. Furthermore, we can learn from Fig. 6 that the collaborative risk-avoidance proportion obtained by the TMC+BP

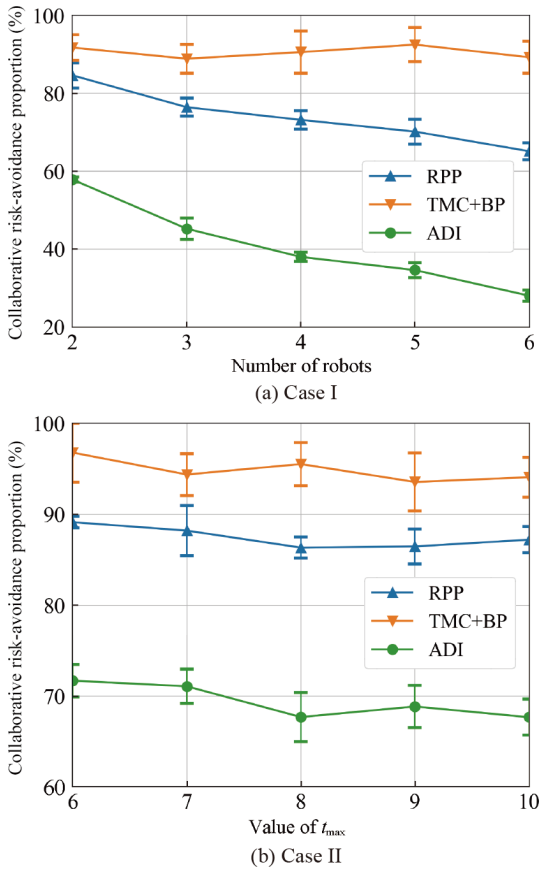


Fig. 6 Test of the collaborative risk-avoidance proportion.

algorithm outperforms the other benchmark algorithms (i.e., RPP and ADI).

Reason. As the number of robots increases, the total travel time and task execution time become larger. Although the collaborative risk-avoidance utility gained by following these algorithms will grow, the time interval that can be used for waiting at each information gathering location decreases. Therefore, the collaborative risk-avoidance proportion decreases, and the phenomena in Case I emerge. As the value of t_{max} increases, the time interval that can be used for waiting at each information gathering location increases. Therefore, the collaborative risk-avoidance proportion increases, and the phenomena in Case II emerge.

6.7 Test of the running time

In this section, we test the efficiency of our proposed TMC+BP algorithm and three other benchmark algorithms for different numbers of robots. Among these algorithms, Cpl is the result returned by solving the CRAP with the optimal solver CPLEX (version 12.6).

Parameter setting. The value of the maximum working time $t_{\max}$ of supervisors is set to 10, and the number of information gathering tasks is set to 15. By varying the number of robots from 2 to 6, we obtain a line graph of the time to run the TMC+BP algorithm and the other three benchmark algorithms, as shown in Fig. 7.

Phenomenon. As the number of robots increases, the runtime of Cpl tends to increase exponentially. The runtime associated with the TMC+BP algorithm and the other benchmark algorithms (i.e., RPP and ADI) does not change significantly with an increasing number of robots, and the corresponding value is always stable within 10 ms.

Reason. As the number of robots increases, the dimension of decision variables used in Cpl increases, and thus the running time tends to increase exponentially. The time complexity of the remaining three algorithms is polynomial, and the running time varies within a reasonable range.

7 Conclusion and Future Work

This paper introduces the collaborative risk-avoidance problem for multirobot information gathering in adversarial environments. We propose a collaborative risk-avoidance model that enables robots to perform tasks in the same area multiple times, improving their risk-avoidance ability and survivability. The model introduces new challenges, including increased complexity of the solution space structure and dimensionality. To address these challenges, we decompose CRAP into two subproblems: task execution and task allocation. We propose an exact algorithm based on branch-and-pricing for task execution and a task map composition algorithm for task allocation.

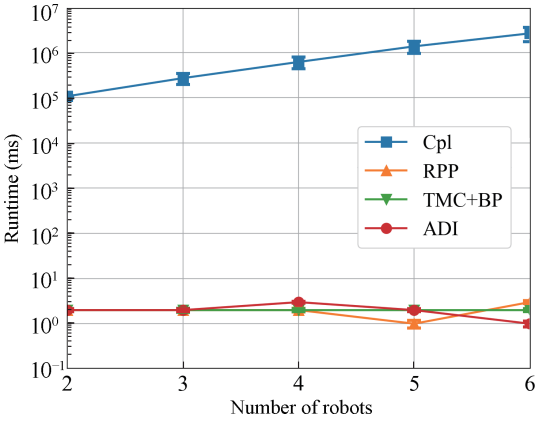


Fig. 7 Test of the running time.

Theoretical analyses and experiments demonstrate the effectiveness and efficiency of our algorithm. Compared to benchmarks, our approach significantly increases the number of collaborative risk-avoidance activities, achieves a higher risk-avoidance proportion, exhibits better robustness, and requires less running time than the CPLEX solver.

In the future, we will mainly explore the adaptation of the collaborative risk-avoidance model to online scenarios, in which the collaborative risk-avoidance activity must be performed without the use of any prior knowledge of other units.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62476121, 62303111, 62076060, and 61932007), the Guangxi Science and Technology Major Program (No. AA24206003), the Key Research and Development Program of Guangxi (No. AB2410317), the Key Research and Development Program of Jiangsu Province of China (No. BE2022157), the Open Funds of KuiYuan Laboratory (No. KY202432), and the Fellowship of China Postdoctoral Science Foundation (No. 2022M720715).

References

[1] Z. Beck, L. Teacy, A. Rogers, and N. R. Jennings, Online planning for collaborative search and rescue by heterogeneous robot teams, in *Proc. Int. Conf. Autonomous Agents & Multiagent Systems*, Singapore, 2016, pp. 1024–1033.

[2] E. Castello, T. Yamamoto, F. D. Libera, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach, *Swarm Intell.*, vol. 10, no. 1, pp. 1–31, 2016.

[3] Y. Jiang, K. Di, R. Qian, X. Wu, F. Chen, P. Li, X. Fu, and Y. Jiang, Optimizing risk-aware task migration algorithm among multiplex UAV groups through hybrid attention multi-agent reinforcement learning, *Tsinghua Science and Technology*, vol. 30, no. 1, pp. 318–330, 2025.

[4] Z. Xiong, H. Xu, W. Li, and Z. Cai, Multi-source adversarial sample attack on autonomous vehicles, *IEEE Trans. Veh. Technol.*, vol. 70, no. 3, pp. 2822–2835, 2021.

[5] D. A. Shell and M. J. Matarić, On information gathering strategies for large-scale multirobot systems, in *Proc. Int. Conf. Intelligent Robots and Systems*, Beijing, China, 2006, pp. 2717–2723.

[6] C. A. B. Baker, S. Ramchurn, W. T. L. Teacy, and N. R. Jennings, Planning search and rescue missions for UAV

teams, in *Proc. 22nd Int. Conf. Autonomous Agents and Multiagent Systems*, Hague, the Netherlands, 2016, pp. 1777–1782.

[7] N. M. Stiffler, A. Kolling, and J. M. O'Kane, Persistent pursuit-evasion: The case of the preoccupied pursuer, in *Proc. IEEE Int. Conf. Robotics and Automation*, Singapore, Singapore, 2017, pp. 5027–5034.

[8] B. P. Gerkey, S. Thrun, and G. Gordon, Visibility-based pursuit-evasion with limited field of view, *Int. J. Robot. Res.*, vol. 25, no. 4, pp. 299–315, 2006.

[9] N. M. Stiffler and J. M. O'Kane, A sampling-based algorithm for multi-robot visibility-based pursuit-evasion, in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, Chicago, IL, USA, 2014, pp. 1782–1789.

[10] N. M. Stiffler and J. M. O'Kane, Pursuit-evasion with fixed beams, in *Proc. IEEE Int. Conf. Robotics and Automation*, Stockholm, Sweden, 2016, pp. 4251–4258.

[11] N. M. Stiffler and J. M. O'Kane, Visibility-based pursuit-evasion with probabilistic evader models, in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, 2011, pp. 4254–4259.

[12] A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff, Approximation algorithms for orienteering and discountedreward TSP, *SIAM J. Comput.*, vol. 37, no. 2, pp. 653–670, 2007.

[13] W. Wang, Z. Dong, B. An, and Y. Jiang, Toward efficient city-scale patrol planning using decomposition and grafting, *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 747–757, 2021.

[14] G. Desaulniers, J. Desrosiers, and M. M. Solomon, *Column Generation*. New York, NY, USA: Springer, 2005.

[15] M. E. Lübbecke and J. Desrosiers, Selected topics in column generation, *Oper. Res.*, vol. 53, no. 6, pp. 1007–1023, 2005.

[16] L. M. Rousseau, M. Gendreau, and D. Feillet, Interior point stabilization for column generation, *Oper. Res. Lett.*, vol. 35, no. 5, pp. 660–668, 2007.

[17] R. Baldacci, N. Christofides, and A. Mingozzi, An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts, *Math. Program.*, vol. 115, no. 2, pp. 351–385, 2008.

[18] Y. Chen, D. Li, Z. Zhang, M. I. M. Wahab, and Y. Jiang, Solving the battery swap station location-routing problem with a mixed fleet of electric and conventional vehicles using a heuristic branch-and-price algorithm with an adaptive selection scheme, *Expert Syst. Appl.*, vol. 186, p. 115683, 2021.

[19] S. Irnich and G. Desaulniers, Shortest path problems with resource constraints, in *Column Generation*, G. Desaulniers, J. Desrosiers, and M. M. Solomon, eds. New York, NY, USA: Springer, 2005.

[20] G. Desaulniers, Managing large fixed costs in vehicle routing and crew scheduling problems solved by column generation, *Comput. Oper. Res.*, vol. 34, no. 4, pp. 1221–1239, 2007.

[21] R. L. Graham and P. Hell, On the history of the minimum spanning tree problem, *Ann. Hist. Comput.*, vol. 7, no. 1, pp. 43–57, 1985.

[22] S. Pettie and V. Ramachandran, An optimal minimum spanning tree algorithm, *J. ACM*, vol. 49, no. 1, pp. 16–34, 2002.

[23] J. J. Chung, A. J. Smith, R. Skeele, and G. A. Hollinger, Risk-aware graph search with dynamic edge cost discovery, *Int. J. Robot. Res.*, vol. 38, nos. 2&3, pp. 182–195, 2019.

[24] R. Yehoshua and N. Agmon, Multi-robot adversarial coverage, in *Proc. 2nd European Conf. Artificial Intelligence*, Hague, the Netherlands, 2016, pp. 1493–1501.

[25] P. Kitjacharoenchai, M. Ventresca, M. Moshref-Javadi, S. Lee, J. M. A. Tanchoco, and P. A. Brunese, Multiple traveling salesman problem with drones: Mathematical model and heuristic approach, *Comput. Ind. Eng.*, vol. 129, pp. 14–30, 2019.

[26] N. C. Rowe and R. S. Alexander, Finding optimalpath maps for path planning across weighted regions, *Int. J. Robot. Res.*, vol. 19, no. 2, pp. 83–95, 2000.

[27] T. Balch and M. Hybinette, Social potentials for scalable multi-robot formations, in *Proc. Int. Conf. Robotics and Automation*, San Francisco, CA, USA, 2000, pp. 73–80.

[28] D. Yu, Y. Zou, J. Yu, Y. Zhang, F. Li, X. Cheng, F. Dressler, and F. C. M. Lau, Implementing the abstract MAC layer in dynamic networks, *IEEE Trans. Mobile Comput.*, vol. 20, no. 5, pp. 1832–1845, 2021.

[29] X. Han, F. Y. L. Chin, H. F. Ting, G. Zhang, and Y. Zhang, A new upper bound 2.5545 on 2d online bin packing, *ACM Trans. Algorithms*, vol. 7, no. 4, p. 50, 2011.

[30] J. Lin, S. Feng, Y. Zhang, Z. Yang, and Y. Zhang, A novel deep neural network based approach for sparse code multiple access, *Neurocomputing*, vol. 382, pp. 52–63, 2020.

[31] R. Ma, G. Dósa, X. Han, H. F. Ting, D. Ye, and Y. Zhang, A note on a selfish bin packing problem, *J. Global Optim.*, vol. 56, no. 4, pp. 1457–1462, 2013.

[32] J. Gautrais, C. Jost, and G. Theraulaz, Key behavioural factors in a self-organised fish school model, *Ann. Zool. Fenn.*, vol. 45, no. 5, pp. 415–428, 2008.

[33] G. J. Hunt, R. E. Page, M. K. Fondrk, and C. J. Dullum, Major quantitative trait loci affecting honey bee foraging behavior, *Genetics*, vol. 141, no. 4, pp. 1537–1545, 1995.

[34] L. A. Real, Uncertainty and pollinator-plant interactions: The foraging behavior of bees and wasps on artificial flowers, *Ecology*, vol. 62, no. 1, pp. 20–26, 1981.

[35] S. Alers, D. Bloembergen, D. Hennes, S. de Jong, M. Kaisers, N. Lemmens, K. Tuyls, and G. Weiss, Bee-inspired foraging in an embodied swarm, in *Proc. 10th Int. Conf. Autonomous Agents and Multiagent Systems*, Taipei, China, 2011, pp. 1311–1312.

[36] S. Alers, D. Claes, K. Tuyls, and G. Weiss, Biologically inspired multi-robot foraging, in *Proc. Int. Conf. Autonomous Agents and MultiAgent Systems*, Paris, France, 2014, pp. 1683–1684.

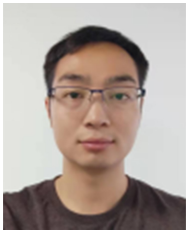
[37] M. Gendreau, G. Laporte, and R. Séguin, Stochastic vehicle routing, *Eur. J. Oper. Res.*, vol. 88, no. 1, pp. 3–12, 1996.

[38] T. Dinh, R. Fukasawa, and J. Luedtke, Exact algorithms for the chance-constrained vehicle routing problem, *Math. Program.*, vol. 172, no. 1, pp. 105–138, 2018.
[39] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, A review of dynamic vehicle routing problems, *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 1–11, 2013.
[40] H. N. Psaraftis, M. Wen, and C. A. Kontovas, Dynamic vehicle routing problems: Three decades and counting, *Networks*, vol. 67, no. 1, pp. 3–31, 2016.
[41] K. Li, G. Luo, Y. Ye, W. Li, S. Ji, and Z. Cai, Adversarial privacy-preserving graph embedding against inference attack, *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6904–6915, 2021.
[42] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, Generative adversarial networks: A survey toward private and secure applications, *ACM Comput. Surv.*, vol. 54, no. 6, pp. 132, 2022.
[43] H. Xu, Y. Li, O. Balogun, S. Wu, Y. Wang, and Z. Cai, Security risks concerns of generative AI in the IoT, *IEEE Internet Things Mag.*, vol. 7, no. 3, pp. 62–67, 2024.
[44] N. M. Stiffler and J. M. O'Kane, Visibility-based pursuit-evasion with probabilistic evader models, in *Proc. IEEE Int. Conf. Robotics and Automation*, Shanghai, China, 2011, pp. 4254–4259.
[45] A. Pierson, A. Ataei, I. C. Paschalidis, and M. Schwager, Cooperative multi-quadrotor pursuit of an evader in an environment with no-fly zones, in *Proc. IEEE Int. Conf. Robotics and Automation*, Stockholm, Sweden, 2016, pp. 320–326.
[46] B. P. Gerkey, S. Thrun, and G. Gordon, Visibility-based pursuit-evasion with limited field of view, *Int. J. Robot. Res.*, vol. 25, no. 4, pp. 299–315, 2006.
[47] F. Yan, J. Jiang, K. Di, Y. C. Jiang, and Z. F. Hao, Multiagent pursuit-evasion problem with the pursuers moving at uncertain speeds, *J. Intell. Robot. Syst.*, vol. 95, no. 1, pp. 119–135, 2019.
[48] F. Yan and Y. Jiang, Pursuing a faster evader based on an agent team with unstable speeds, in *Proc. 16th Conf. Autonomous Agents and MultiAgent Systems*, São Paulo, Brazil, 2017, pp. 1766–1768.
[49] J. Chen, W. Zha, Z. Peng, and D. Gu, Multi-player pursuit-evasion games with one superior evader, *Automatica*, vol. 71, pp. 24–32, 2016.
[50] W. Sun and P. Tsiotras, Sequential pursuit of multiple targets under external disturbances via Zermelo-Voronoi diagrams, *Automatica*, vol. 81, pp. 253–260, 2017.
[51] H. Shi, C. Wang, H. Zhao, S. Wang, and Y. Chen, Bioinvasion risk analysis based on automatic identification system and marine ecoregion data, *High-Confidence Comput.*, vol. 4, no. 4, p. 100210, 2024.
[52] H. Min, J. Cao, T. Zhou, and Q. Meng, IPSA: A multi-view perception model for information propagation in online social networks, *Big Data Min. Anal.*, vol. 8, no. 1, pp. 241–256, 2025.
[53] B. P. Gerkey and M. J. Matarić, A formal analysis and taxonomy of task allocation in multi-robot systems, *Int. J. Robot. Res.*, vol. 23, no. 9, pp. 939–954, 2004.
[54] A. F. T. Winfield, Foraging robots, in *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, ed. New York, NY, USA: Springer, 2009, pp. 3682–3700.
[55] F. Yan and K. Di, Capturing a faster evader by some energy-limited speed-controllable pursuers, in *Proc. IEEE Int. Conf. Systems, Man, and Cybernetics* (*SMC*), Toronto, Canada, 2020, pp. 1354–1361.
[56] Z. Xiong, W. Li, and Z. Cai, Federated generative model on multi-source heterogeneous data in IoT, in *Proc. 37th AAAI Conf. Artificial Intelligence*, Washington DC, USA, 2023, pp. 10537–10545.
[57] X. Zheng, L. Tian, G. Luo, and Z. Cai, A collaborative mechanism for private data publication in smart cities, *IEEE Internet Things J.*, vol. 7, no. 9, pp. 7883–7891, 2020.
[58] E. S. Lin, N. Agmon, and S. Kraus, Multi-robot adversarial patrolling: Handling sequential attacks, *Artif. Intell.*, vol. 274, pp. 1–25, 2019.
[59] R. Yehoshua, N. Agmon, and G. A. Kaminka, Robotic adversarial coverage of known environments, *Int. J. Robot. Res.*, vol. 35, no. 12, pp. 1419–1444, 2016.
[60] K. Di, Y. Zhou, J. Jiang, F. Yan, S. Yang, and Y. Jiang, Risk-aware collection strategies for multirobot foraging in hazardous environments, *ACM Trans. Auton. Adapt. Syst.*, vol. 16, nos. 3&4, p. 8, 2021.
[61] H. L. Choi, L. Brunet, and J. P. How, Consensus-based decentralized auctions for robust task allocation, *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, 2009.
[62] F. Yan, K. Di, J. Jiang, Y. Jiang, and H. Fan, Efficient decision-making for multiagent target searching and occupancy in an unknown environment, *Robot. Auton. Syst.*, vol. 114, pp. 41–56, 2019.
[63] P. Li, K. Di, X. Bai, F. Chen, Y. Jiang, and X. Fu, Time-of-use price resource scheduling in multiplex networked industrial chains, *Tsinghua Science and Technology*, vol. 30, no. 1, pp. 303–317, 2025.
[64] Y. Jiang, K. Di, X. Wu, Z. Hu, F. Chen, P. Li, and Y. Jiang, Layered temporal spatial graph attention reinforcement learning for multiplex networked industrial chains energy management, *Tsinghua Science and Technology*, vol. 30, no. 2, pp. 528–542, 2025.
[65] Z. Beck, L. Teacy, A. Rogers, and N. R. Jennings, Online planning for collaborative search and rescue by heterogeneous robot teams, in *Proc. Int. Conf. Autonomous Agents & Multiagent Systems*, Singapore, Singapore, 2016, pp. 1024–1033.
[66] E. Castello, T. Yamamoto, F. D. Libera, W. Liu, A. F. T. Winfield, Y. Nakamura, and H. Ishiguro, Adaptive foraging for simulated and real robotic swarms: The dynamical response threshold approach, *Swarm Intell.*, vol. 10, no. 1, pp. 1–31, 2016.
[67] S. Liemhetcharat, R. Yan, and K. P. Tee, Continuous foraging and information gathering in a multi-agent team, in *Proc. 14th Int. Conf. Autonomous Agents and Multiagent Systems*, Istanbul, Turkey, 2015, pp. 1325–1333.
[68] G. Reinelt, TSPLIB—A traveling salesman problem library, *ORSA J. Comput.*, vol. 3, no. 4, pp. 376–384, 1991.
[69] G. Laporte, A. Asef-Vaziri, and C. Sriskandarajah, Some applications of the generalized travelling salesman

problem, *J. Oper. Res. Soc.*, vol. 47, no. 12, pp. 1461–1467, 1996.
[70] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, The orienteering problem: A survey, *Eur. J. Oper. Res.*, vol. 209, no. 1, pp. 1–10, 2011.
[71] C. Archetti, N. Bianchessi, and M. G. Speranza, The capacitated team orienteering problem with incomplete service, *Optim. Lett.*, vol. 7, no. 7, pp. 1405–1417, 2013.
[72] C. Chen, S. F. Cheng, and H. C. Lau, Multi-agent orienteering problem with time-dependent capacity constraints, *Web Intell. Agent Syst. Int. J.*, vol. 12, no. 4, pp. 347–358, 2014.
[73] D. S. Johnson, The NP-completeness column: An ongoing guide, *J. Algorithms*, vol. 6, no. 3, pp. 434–451, 1985.
[74] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.
[75] P. Shi, M. Zhao, W. Wang, Y. Zhou, J. Jiang, J. Zhang, Y. Jiang, and Z. Hao, Best of both worlds: Mitigating imbalance of crowd worker strategic choices without a budget, *Knowl. Based Syst.*, vol. 163, pp. 1020–1031, 2019.
[76] P. Shi, W. Wang, Y. Zhou, J. Jiang, Y. Jiang, Z. Hao, and J. Yu, Practical POMDP-based test mechanism for quality assurance in volunteer crowdsourcing, *Enterp. Inf. Syst.*, vol. 13, no. 7, pp. 979–1001, 2019.
[77] G. B. Dantzig and P. Wolfe, Decomposition principle for linear programs, *Oper. Res.*, vol. 8, no. 1, pp. 101–111, 1960.
[78] E. Choi and D. W. Tcha, A column generation approach to the heterogeneous fleet vehicle routing problem, *Comput. Oper. Res.*, vol. 34, no. 7, pp. 2080–2095, 2007.
[79] R. Ebendt, W. Gunther, and R. Drechsler, Combining ordered best-first search with branch and bound for exact BDD minimization, *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 24, no. 10, pp. 1515–1529, 2005.
[80] B. W. Wah and C. F. Yu, Stochastic modeling of branch-and-bound algorithms with best-first search, *IEEE Trans. Software Eng*, vol. SE-11, no. 9, pp. 922–934, 1985.
[81] R. Dechter and J. Pearl, Generalized best-first search strategies and the optimality of A*, *J. ACM*, vol. 32, no. 3, pp. 505–536, 1985.
[82] R. E. Korf, Linear-space best-first search, *Artif. Intell.*, vol. 62, no. 1, pp. 41–78, 1993.
[83] S. Poikonen, B. Golden, and E. A. Wasil, A branch-and-bound approach to the traveling salesman problem with a drone, *INFORMS J. Comput.*, vol. 31, no. 2, pp. 335–346, 2019.
[84] S. Tanaka, K. Tierney, C. Parreño-Torres, R. Alvarez-Valdes, and R. Ruiz, A branch and bound approach for large pre-marshalling problems, *Eur. J. Oper. Res.*, vol. 278, no. 1, pp. 211–225, 2019.
[85] D. M. Ryan and B. A. Foster, An integer programming approach to scheduling, *Computer scheduling of public transport urban passenger vehicle and crew scheduling*, 1981, pp. 269–280.



Kai Di received the PhD degree from School of Computer Science and Engineering, Southeast University, China, in 2022. He is currently an assistant professor and the director at School of Computer Science and Engineering, Southeast University, Nanjing, China. He has authored or coauthored several scientific articles in refereed journals and conference proceedings, such as *ACM Transactions on Autonomous and Adaptive Systems*, *ACM Transactions on Intelligent Systems and Technology*, and *IEEE International Conference on Systems, Man, and Cybernetics*. His research interests include scheduling optimization and multivariate time series analysis.



Pan Li received the MS degree from Hebei University, Baoding, China in 2022. He is currently pursuing the PhD degree at School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include scheduling optimization and multivariate time series analysis.



Tienyu Zuo received the MS degree in electronic information engineering from Nanjing University of Information Science and Technology, Nanjing, China, in 2024. He is currently pursuing the PhD degree in computer science at Southeast University, Nanjing, China. His works have been published in *IEEE Transactions on Intelligent Transportation Systems* and *Applied Soft Computing*. His research interests include multi-agent systems with large language models (LLMs), social networks, and social simulation.



Fulin Chen received the MS degree in automation control from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2021. He is currently pursuing the PhD degree at School of Cyber Science and Engineering, Southeast University, Nanjing, China. His research interests include data mining, scheduling optimization, and load forecasting.



Yuanshuang Jiang received the MS degree from Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China, in 2021. He is currently pursuing the PhD degree in computer science at Southeast University, Nanjing, China. His research interests include reinforcement learning and scheduling optimization.



Dan Chen received the BS degree in radio technology engineering from Guangxi University, China, in 1992. She is currently the vice-chancellor at Beibu Gulf University, China. Her research interests include communication technology, artificial intelligence, pattern recognition, and collaborative optimisation technology.



Yichuan Jiang received the PhD degree in computer science from Fudan University, China, in 2005. He is currently a distinguished professor and the director at School of Computer Science and Engineering, Southeast University, Nanjing, China. He has authored or coauthored more than 100 scientific articles in refereed journals, such as the *IEEE Transactions on Parallel and Distributed Systems*, *IEEE Journal on Selected Areas in Communications*, *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Systems*, *Man*, and *Cybernetics*: *Systems*, *IEEE Transactions on Cybernetics*, *ACM Transactions on Autonomous and Adaptive Systems*, *ACM Transactions on Intelligent Systems* and *Technology*, and *Autonomous Agents* and *Multi-Agent Systems*, and in conference proceedings, such as IJCAI, AAAI, and AAMAS. His research interests include multi-agent systems, social networks, and social simulation.