Adversarial Decision Making Against Intelligent Targets in Cooperative Multiagent Systems

Yang Li, Huaping Liu, *Senior Member, IEEE*, Fuchun Sun, and Zhihong Chen

***Abstract*—In this article, we propose a distributed adversarial decision-making approach for multiagent system (MAS), which is supposed to detect a team of intelligent targets. The MAS is demanded to achieve the best detection benefit against the intelligent targets that can shelter part of their features and prevent the detection according to an expert defense policy. One of the key challenges is how to achieve higher detection benefit that is both determined by the target assignment action of the MAS and the nonpredictable defense policy of the targets. To handle this problem, we first formulate the multiagent decision-making problem as a max–min problem of detection benefit and break it into separate parts that are easier to be optimized. Then, we introduce a new variant of distributed alternating direction method of multipliers (ADMM) to search the optimal solutions under the worst defense policy that the targets choose. To overcome the lack of access to the global convergence of multiblock ADMM, we add local additional variables to formulate a penalty for nonconvex parts of the local objective function. The convergence to an equilibrium and the optimality of the detection benefit are empirically validated by numerical simulations. The influence of the parameter setting is also presented and can be regarded as a prior suggestion for real applications.**

***Index Terms*—Adversarial decision making, alternating direction method of multiplier (ADMM), multiagent system (MAS).**

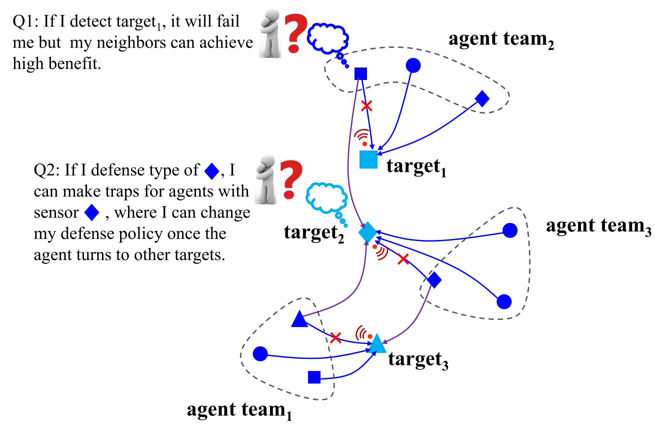


Fig. 1. Motivating application of adversarial decision making for MAS. There exist nine agents (dark blue) that are demanded to detect three targets (light blue). The benefit of detection is decided by the distance. Short distance brings high benefit. The agents can only detect the targets with different shapes, however, the shape of targets is unknown. The minimal detection benefit is attained when the initial assignment action of MAS is shown as the blue arrows. If we change the assignment as purple arrows, a better detection benefit will be obtained.

I. Introduction

The MULTIAGENT system (MAS) has been widely used in practical applications, such as data collection [1], [2], [3], identifying targets [4], environment monitoring [5], [6], [7], etc. Extensive research has been developed to improve the autonomy and to enhance the intelligence level of MAS, including decision making, SLAM, and navigation [8], [9], [10], [11]. Among them, the decision making that is focused on task assignment for MAS is a key direction because it mostly determines the maximum of efficiency level [12], [13]. The benefit can be formulated based on prior knowledge of tasks and agents [14], [15]. The goal is to find the best assignment action that satisfies different constraints according to specific applications [11], [16], [17], [18]. However, the traditional methods suffer from the adversarial settings. For example, as shown in Fig. 1, the MAS is demanded to detect a team of intelligent targets and MAS wants to maximize the detection benefit with the unknown defense of the targets. All targets can hide themselves by interfering with sensors of MAS to stop agents using the same type of sensors from detecting them. We assume, without losing generality, that the targets are aware of MAS beforehand, including the types and ranges of the sensors. Every target has the option of choosing the best defense policy or the interference that minimizes the benefit of the detection. The agents are supposed to maximize the benefit of detection and take into account the worst case scenario, where all targets can always make the optimal interference. The adversarial decision-making problems for task assignment have been discovered to be NP-hard [19]. In this article, we develop an effective solver to deal with distributed decision-making problem for task assignment considering the adversary target interference. Our contributions can be summarized as follows.

Manuscript received 27 July 2022; revised 27 October 2022 and 20 December 2022; accepted 24 December 2022. Date of publication 4 January 2023; date of current version 12 February 2024. This work was supported in part by the National Natural Science Fund for Distinguished Young Scholars under Grant 62025304, and in part by the China Postdoctoral Science Foundation under Grant 2022M711750. *(Corresponding author: Huaping Liu.)*

Yang Li, Huaping Liu, and Fuchun Sun are with the State Key Laboratory of Intelligent Technology and Systems, Beijing National Research Center for Information Science and Technology, and the Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: liyang90@tsinghua.edu.cn; hpliu@tsinghua.edu.cn; fcsun@tsinghua.edu.cn).

Zhihong Chen is with the Beijing Institute of Precision Mechatronics and Controls, China Aerospace Science and Technology Corporation, Beijing 100076, China (e-mail: zhihongch@outlook.com).

Digital Object Identifier 10.1109/TCDS.2023.3234061

1) We propose a brand-new distributed multiagent decision-making framework for task assignment to maximize the benefit of detection in the worst case scenario where the targets choose the best interference. Considering the distributed and connected topology of MAS, we split the detection benefit into separate parts that are easier to be optimized.
2) We design additional penalties to encourage the assignment action to cover all targets. We also talk about the relationship between the performance of our algorithm and the parameter settings of the penalties.
3) The results of numerical simulations demonstrate that our approach can yield the optimal performance with the greatest benefit of detection. To the best of our knowledge, it is the first work that adopts alternating direction method of multiplier (ADMM) to solve the distributed adversarial decision-making problem.

The remainder of this article is organized as follows. In Section II, related works about adversarial decision making for task assignment are introduced and summarized. In Section III, we formulate the problems as a max–min optimization problem and describe our algorithm in Section IV. We validate the effectiveness of our methods in Section V. Finally, the conclusion of this article are drawn in Section VI.

II. Related Work

A. Decision Making for Task Assignment

The decision-making methods for task assignment can be classified into two groups according to whether the decision variables are continuous or discrete. The discrete binary set of $\{0, 1\}$ is the most common choice and constrained integer linear programming (ILP) problems can be formulated [20], [21]. Because they have been proven to be NP-hard [22], many researchers attempted to directly estimate the optimal solution by auction-based algorithms [16], simplex-based algorithms [23], max-sum algorithms [24], heuristic-based search algorithms [25], [26], and evolutionary algorithms [27]. Another common method is using a bounded interval random variable in $[0, 1]$ to represent how likely the tasks will be assigned to the agents. For example, automated-guided vehicles (AGVs) task assignment problems are solved by converting ILP problems into linear programming (LP) problems [28]. Haksar et al. [29] translated the ILP into an LP using relaxation methods to solve the persistent surveillance problem subject to battery constraints.

B. Adversarial Elements

Due to the increasing risk of failure, adversarial factors are particularly notable in decision making for task assignment [30]. To counter the defense system, Luo et al. [31] studied the missile target assignment problems to fight against the defense system. They construct a deep Q-Network to learn the best policy from historical data. Reily and Zhang [32] concentrated on the impact of the passive team change caused by adversarial attacks or jamming. Park and Hutchinson [33] proposed a robust assignment algorithm to cover a specific region of interest using a team of sensors. They use Lloyd's method to manage the worst case scenario when a sensor betrays the team and tries to impede the coverage. False and misleading information are also typical antagonistic components that make it challenging for MAS to complete the tasks. Schwartz and Tokekar [34] tried to ensure the performance even when agents fail accidentally. Research in [35] solves the task assignment problem when the numbers of sensors and agents are nonstationary. They demonstrate that the methods can cope with sudden failures of MAS. Although researchers propose many algorithms on how to detect the adversarial or noncooperative ones [36], it is still crucial to provide more reliable assignment action to immunize against the uncertainties of the adversarial elements. In [37], the payoff of each agent assignment action is stochastic. The consumption is guaranteed within the limits with a high probability. Likewise, Rudolph et al. [38] solved the decision-making problem for task assignment with the uncertain capacity of agents, which ensures various of the task constraints.

C. ADMM in Decision Making for Task Assignment

A dual-phase optimization is a promising tool for many optimization problems [39]. ADMM is a dual-phase optimization method, which is first proposed in [40] and has received a significant amount of attention because it can solve equality-constrained optimization problems more efficiently. For many applications, ADMM requires fewer iterations to converge with acceptable accuracy [41]. In order to optimize the objective function in primal and dual directions, ADMM combines the dual ascent and the multiplier approach making it more suitable for scenarios where the subproblems are relatively simple to address. It means that ADMM can be used to coordinate many processors, each of which solves a substantial problem. It also has been utilized successfully in solving many machine learning problems [42], [43]. The deployment of parallel processing and appropriate convergence accuracy is two major benefits of ADMM. As a result, many researchers use ADMM to resolve decision-making problems for task assignment, including railway rolling stock assignment [44], base stations assignment for uplink transmission [45], decentralized parking lots assignment, etc. However, in adversarial scenarios, we aim to maximize the benefit, but the adversarial elements try to minimize it simultaneously. Since ADMM has been used successfully in distributed optimization, we will try to figure out the equivalent max–min problem in our adversarial decision making for target assignment.

III. Problem Formulation

Suppose a team of heterogeneous agents $\mathcal{A} = \{1, \ldots, A\}$ equipped with different sensors $\mathcal{S} = \{1, \ldots, S\}$ is demanded to detect targets $\mathcal{M} = \{1, \ldots, M\}$. First, we denote the sensor configuration of MAS by $C = [c_1, c_2, \ldots, c_A]$, and

$$[c_a]_s = \begin{cases} 1, & \text{agent } a \text{ equipped with sensor} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $[c_a]_s$ is the sth element of vector c_a. Each target has a limited anti-detection capacity to disable one type of the sensors in our adversarial settings. We denote the corresponding defense of targets as $D = [d_1, d_2, \ldots, d_m]$, and

$$[d_m]_s = \begin{cases} 1, & \text{target } m \text{ interferes the sensor } s \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $[d_m]_s = 1$ means that the target m cannot be detected by sensor s. The detection result is also influenced by the detection capability of sensors, which is defined as $\lambda \in \mathbb{R}_+^S$. The decay coefficients caused by the distance between agents and targets are denoted as $B = [b_1, b_2, \ldots, b_A]$. And we use b_a to represent the coefficients vector and $b_a \in \mathbb{R}^M$. Second, we define the assignment action of MAS as $X = [x_1, x_2, \ldots, x_A]$, and

$$[x_a]_m = \begin{cases} 1, & \text{target } m \text{ is assigned to agent } a \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In this article, we define the status-related benefit of MAS as function $f(X, D)$, and

$$\begin{aligned} f(X, D) &= \sum_{a=1}^{A}\sum_{m=1}^{M}\sum_{s=1}^{S}[b_a]_m \lambda_s [c_a]_s (1 - [d_m]_s) \\ &= 1_A^\top \left\{ B \circ \left[C^\top \Lambda (1_{S\times M} - D) \right]^\top \circ X \right\} 1_M \end{aligned} \tag{4}$$

where $\circ$ means the Hadamard product of matrices. 1_A and 1_M represent the size A and M row vectors of 1 s. We use continuous variables within [0, 1] to describe the assignment actions and target interference decisions. Specifically, we set $X \in [0, 1]^{M\times A}$ and $D \in [0, 1]^{S\times M}$. Feasible sets of X and D are denoted as Ω_X and Ω_D. Intuitionistic constraints are supposed to be satisfied as follows:

$$\Omega_D = \left\{ D \,\middle|\, \begin{array}{l} D \succeq 0 \\ 1_S^\top D = 1_M^\top \end{array} \right\} \tag{5}$$

$$\Omega_X = \left\{ X \,\middle|\, \begin{array}{l} X \succeq 0 \\ 1_M^\top X = 1_A^\top \\ \|[X]^m\|_\infty = 1 \ \forall m = 1, \ldots, M \end{array} \right\} \tag{6}$$

where $\succeq 0$ means all elements greater than or equal to 0. The second term limits the measure of the entire sample space to 1. The third term means that every target is at least assigned to one agent, i.e., coverage of target detection.

The detection benefit $f(X, D)$ should be maximized under the worst scenario where the defense policy of targets is optimal. We formulate the centralized decision-making problem that is focused on task assignment as follows.

Problem 1 (Centralized Adversarial Decision-Making Problem for Task Assignment): Given the sensor configuration of MAS C, detection ability of sensors Λ, and decay coefficients B, the centralized adversarial decision-making problem for task assignment is equivalent to the following max–min problem:

$$\begin{cases} \max\limits_{X} \min\limits_{D}, & f(X, D) \\ \text{subject to}, & X \in \Omega_X, D \in \Omega_D. \end{cases} \tag{7}$$

However, the stability of MAS is limited by the central agent, which is easily overcome by distributed topology [46]. Each agent can communicate with other agents within its communication range in a distributed way. We assume that the topology of MAS is connected and refer to the set of an agent's neighbors as $\mathcal{N}_i$. Additionally, we make a general assumption of stable and synchronous communication. Fortunately, it is possible to split the detection benefit function into separate parts as

Algorithm 1 ADMM for Consensus Problem [40]

Initialization: $\left\{x_i^1\right\}, \rho$
repeat
  Update $x_i^{k+1} = \arg\min f_i(x_i) + \rho/2\|x_i - \bar{x}^k + \frac{1}{\rho}y_i^k\|$
  Update $y_i^{k+1} = y_i^k + \frac{1}{\rho}(x_i^{k+1} - x_i^k)$
  Update $\bar{x}^{k+1} = \frac{1}{n}\sum_{i=1}^{n} x_i^k$
until $\sum_{i=1}^{n}\|x_i^k - \bar{x}^k\|^2 \le \epsilon_c, n\rho^2\|\bar{x}^{k+1} - \bar{x}^k\|^2 \le \epsilon_o$
Output: $x_i^* = x_i^k$

$$f(X, D) = \sum_{i=1}^{A} f_i(x_i, D) \tag{8}$$

where $f_i(x_i, D) = 1_M^\top\{b_i \circ [c_i^\top \Lambda (1_{S\times M} - D)]^\top \circ x_i\}$

Using $f(X, D) = \sum_{i=1}^{A} f_i(x_i, D)$, we can formulate the distributed task assignment problem as a distributed max–min problem as follows.

Problem 2 (Distributed Adversarial Decision-Making Problem for Task Assignment): Given the condition of Problem 1 and the topology of MAS $G = (V, E)$, the distributed adversarial decision-making problem for task assignment is equivalent to solve the following max–min problem:

$$\begin{cases} \max\limits_{x_1,\ldots,x_A} \min\limits_{D}, \ \sum_{i=1}^{A} f_i(x_i, D) \\ \text{subject to}, \ X = [x_1, \ldots, x_A] \in \Omega_X, D \in \Omega_D. \end{cases} \tag{9}$$

There already exist several studies on the centralized decision making for task assignment both in nonadversarial and adversarial settings, i.e., Problem 1 as stated before, we mainly seek an effective solver for Problem 2 in this article as follows.

IV. Method

A. ADMM on Consensus Problems

ADMM is a powerful tool for distributed optimization if the objective function can be split into separate parts, i.e., $f(x) = \sum_{i=1}^{n} f_i(x)$. The optimization problem can be reformulated as the consensus problem by adding supplement variables $\{x_1, \ldots, x_n\}$ and equality constraints as follows:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{n} f_i(x_i) \\ &\text{subject to} && x_1 = x_2 = \cdots = x_n. \end{aligned} \tag{10}$$

The global optimum of Problem 1 can be achieved iteratively by optimizing each local part and the dual variable alternatively (as shown in Algorithm 1). The convergence is ensured by the equality constraints. According to the optimality conditions proposed in [40], two stopping criteria can be designed to represent the primal and dual residuals of the optimality as r^{k+1}, s^{k+1}

$$r^{k+1} = \sum_{i}^{n} \|x_i^k - \bar{x}^k\|^2 \tag{11}$$

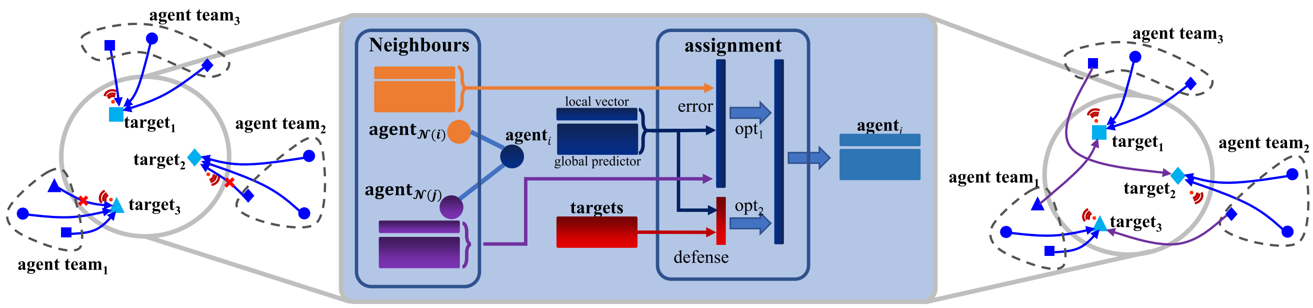


Fig. 2. Distributed adversarial decision-making framework for task assignment. There are two main steps in each optimization iteration. The agent tunes its own assignment action and the estimations of neighbors iteratively until convergence is reached.

$$s^{k+1} = n\rho^2 \|\bar{x}^{k+1} - \bar{x}^k\|^2 \tag{12}$$

where $\bar{x}^k$ is the average value of x_i at iteration k. We see that the primal residual r^{k+1} can represent the convergence and the dual residual s^{k+1} can represent the suboptimality of the objective function. Once the stopping criteria are satisfied, i.e., convergence error ϵ_c and optimization error ϵ_o are both satisfied, the optimal solution of (10) is achieved.

B. Framework of Distributed Decision Making

We propose a distributed decision-making framework for task assignment to solve Problem 2. Fig. 2 depicts the structure and primary flow, which is an extension of the consensus-ADMM proposed in [40]. Considering the distributed communication topology, the assignment action of each agent should be obtained separately. To get the identical assignment actions among the MAS, we split the assignment actions of each agent into local and global components. The global vector makes an estimate of the assignment action of its neighbors, while the local vector describes its own action. We then get two errors that are supposed to be minimized from both assignment actions: 1) benefit optimization error (dual error called in ADMM) and 2) the consensus error (primal error called in ADMM). Then, we design a two-phase optimization method to address the errors synchronously until the bounds are met.

C. Constraints of Task Assignment

We use two indicator functions $g(X)$ and $h(D)$ to represent the feasible set constraints of Ω_X and Ω_D, i.e., $g(X) = \mathbf{1}_{(X\in\Omega_X)}$, $h(D) = \mathbf{1}_{(D\in\Omega_D)}$, and $\mathbf{1}_{(\text{condition})} = 1$. It should be noted that $g(X)$ and $h(D)$ are easily split into $g(X) = \sum_{p=1}^{3} g_p(X)$, $h(D) = \sum_{q=1}^{2} h_q(D)$, and

$$\begin{cases} g_1(X) = \mathbf{1}_{(X\succeq 0)} \\ g_2(X) = \mathbf{1}_{(1_M^\top X = 1_A^\top)} \\ g_3(X) = \mathbf{1}_{(\|[X]^m\|_\infty = 1)} \end{cases} \begin{cases} h_1(D) = \mathbf{1}_{(D\succeq 0)} \\ h_2(D) = \mathbf{1}_{(1_S^\top D = 1_M^\top)}. \end{cases} \tag{13}$$

The equality constraints in (5) and (6) are both affine functions except the coverage constraints term of $\|[X]^m\|_\infty = 1$. However, the support set of the equality term is not convex, which leads to nonconvergence. Motivated by [47] and [48], we introduce the normalization penalization term $\|X\|_F^2$ to encourage each element of X to converge to 0 or 1. Besides, we bound the sum of the assignment solution on each target ≥ 1 to guarantee that all targets are assigned to at least one agent. Specifically, we add supplement objective $\|X\|_F^2$ and the indicator constraint function $g_3(X) = \mathbf{1}_{(X 1_A \geq 1_M)}$ in the objective function $f(X, D)$.

D. ADMM-Based Distributed Optimization Method

An example of an ADMM distributed solver to an optimization problem is the sharing problem. By tightening the global consistency constraints on local solutions, ADMM achieves the distributed solution of optimization problems, which can significantly increase the convergence rate of those problems in their early phases. When the optimization problem involves additional constraints, it must also fulfill the local separability of those constraints. The objective function must be separable in order to meet the sharing problem. MAS works in a distributed manner, which means that each agent can only obtain information of its neighbors, i.e., x_i is only available for agents set $x_j \in \mathcal{N}_i$. It makes the optimization unfeasible because the constraints of $g_3(X)$ cannot be split. It also means that every agent will focus on its local detection benefit and makes the MAS noncooperatively. To handle this issue, we add supplement variables for each agent to predict the assignment actions of MAS, i.e.,

$$X_i = \left[\hat{x}_1, \ldots, x_i, \ldots, \hat{x}_A\right] \tag{14}$$

where $\hat{x}_j, j \neq i$, is the assignment action estimation of other agents. Thus, we can reformulate Problem 2 as a consensus problem as follows:

$$\begin{aligned} \max_{X_1,\ldots,X_A} \min_{D} \quad & \sum_{i=1}^{A} f_i(X_i, D) \\ \text{subject to} \quad & X_i \in \Omega_X \ \forall i = \{1, \ldots, A\} \\ & X_i = X_{j\in\mathcal{N}_i} \ \forall i = \{1, \ldots, A\} \\ & D \in \Omega_D \end{aligned} \tag{15}$$

where

$$f_i(X_i, D) = 1_M^\top \left(B \circ \left[C^\top \Lambda (1_{S\times M} - D) \right]^\top \circ E_i \circ X_i \right) 1_A \tag{16}$$

and E_i is a matrix that all elements are zeros except the ith column is 1_M. Notice that $1_M^\top (X \circ D) 1_A$ equals to $\mathrm{tr}(X^\top D)$, we transform $f_i(X_i, D)$ with respect to X_i as

$$f_i(X_i, D) = \mathrm{tr}\left(\left[B \circ \left[C^\top \Lambda (1_{S\times M} - D) \right]^\top \circ E_i \right]^\top X_i \right). \tag{17}$$

Algorithm 2 Adversarial Distributed Decision Making for Task Assignment

Input: Agent capacity C, sensor coefficients B, topology $G = (V, E)$
1: Initialization: primal and dual error bounds ϵ_c, ϵ_o, augment parameter ρ, assignment action X_i^1 and defense action D^1
2: **repeat**
3: Update $X_i^{k+1} \triangleq \arg\max_{X_i} \mathcal{L}_\rho$
4: Update $D^{k+1} \triangleq \arg\min_D \mathcal{L}_\rho$
5: Update $Y_{ij} = Y_{ij} + \frac{1}{\rho}(X_i^{k+1} - X_j^{k+1})$
6: **until** $\sum_{i=1}^A \|X_i^k - \bar{X}^k\|_F^2 \le \epsilon_c$, $A\rho^2\|\bar{X}^{k+1} - \bar{X}^k\|_F^2 \le \epsilon_o$
Output: Optimal assignment action $X^* = \bar{X}_i^k$ and defense action $D^* = D^k$

We also formulate $\sum_{i=1}^A f_i(X_i, D)$ with respect to D as

$$f_i(X_i, D) = -\mathrm{tr}\Big(\Big[B \circ \Big(\textstyle\sum_{i=1}^A E_i \circ X_i\Big)\Big]C^\top \Lambda D\Big) + \mathrm{const}(D) \tag{18}$$

where const(D) is an uncorrelated constant matrix of D. The new objective function can be formulated as follows:

$$\begin{aligned}\mathcal{L}_\rho = &\sum_{i=1}^A \left\{ f_i(X_i, D) + \sum_{p=1}^3 g_p(X_i) + \mu\|X_i\|_F^2 \right\} \\ &+ \sum_{i=1}^A \sum_{j\in\mathcal{N}_i} \left\{ \mathrm{tr}\Big(Y_{ij}^\top (X_i - X_j)\Big) + \frac{\rho}{2}\|X_i - X_j\|_F^2 \right\} \\ &+ \sum_{q=1}^2 h_q(D).\end{aligned} \tag{19}$$

Based on (19), we propose an ADMM-based algorithm to optimize (15) iteratively as shown in Algorithm 2.

By optimizing $\mathcal{L}_\rho$ alternatively with respect to X_i and D, we eventually narrow down the primal and dual errors of problem in (15), i.e., $\sum_{i=1}^A \|X_i^k - \bar{X}^k\|_F^2$ and $A\rho^2\|\bar{X}^{k+1} - \bar{X}^k\|_F^2$. The corresponding bounds of ϵ_c and ϵ_o guarantee the consistency of MAS and the closeness to the optimal solutions. They are supposed be provided by the designer according to the real applications.

1) Optimization of X_i: Given the prior D^k, we update X_i^{k+1} to maximize the detection benefit by maximizing (19). It should be noticed that the objective $\mathcal{L}_\rho$ in (19) can be optimized with respect to X_i in parallel. Then, we formulate the distributed local optimization as follows:

$$\begin{aligned}X_i^{k+1} = &\arg\min_{X_i} -f_i\Big(X_i, D^k\Big) - \mu\|X_i\|_F^2 + \sum_{p=1}^3 g_p(X_i) \\ &+ \sum_{i=1}^A \sum_{j\in\mathcal{N}_i} \left\{ \mathrm{tr}\Big(Y_{ij}^{k\top}\Big(X_i - X_j^k\Big)\Big) + \frac{\rho}{2}\|X_i - X_j^k\|_F^2 \right\} \\ &+ \sum_{j\in\mathcal{N}_i} \left\{ \mathrm{tr}\Big(Y_{ji}^{k\top}\Big(X_j^k - X_i\Big)\Big) + \frac{\rho}{2}\|X_j^k - X_i\|_F^2 \right\}.\end{aligned} \tag{20}$$

The indicators $g_p(X_i)$ are defined in (13). The local optimization problem (20) can be solved in a compact form as follows:

$$\begin{aligned}X_i^{k+1} = &\arg\min_{X_i} -f_i(X_i, D) + \sum_{p=1}^3 g_p(X_i) \\ &+ (\rho|\mathcal{N}_i| - \mu)\Big\|X_i - Z_i^k\Big\|_F^2\end{aligned} \tag{21}$$

where

$$Z_i^k = \frac{\rho}{\rho|\mathcal{N}_i| - \mu} \sum_{j\in\mathcal{N}_i} \left[X_j^k - \frac{1}{2\rho}\Big(Y_{ij}^k - Y_{ji}^k\Big)\right]. \tag{22}$$

To handle the nonlinear part in (21), i.e., $g_p(X_i)$, we split the solution of X_i^{k+1} into four parts $\{X_{i1}, X_{i2}, X_{i3}, X_{i4}\}$ with the equality constraint $X_{i1} = X_{i2} = X_{i3} = X_{i4}$. Thus, we can obtain X_i^{k+1} based on a consensus solver in Algorithm 3.

The consensus solver in Algorithm 3 is similar to the ADMM-based solver in Algorithm 1. One of the main differences is that we use matrices other than vectors to represent the primal and dual variables. And we use the Frobenius norm of the primal and dual errors. We use the same primal and dual error bounds ϵ_c and ϵ_o here to achieve the same optimization precision.

Algorithm 3 Update of X_i^{k+1}

Input: Defense action D^k, constant parameter Z_i^k, augment parameter ρ_x, primal and error bounds ϵ_c, ϵ_o
1: Initialization: X_{ij}^1, W_{ij}^1
2: **repeat**
3: Update $X_{i0}^{k'+1}$ based on (23)
4: Update $X_{i1}^{k'+1}$ based on (24)
5: Update $X_{i2}^{k'+1}$ based on (25)
6: Update $X_{i3}^{k'+1}$ based on (26)
7: Update $W_{ij}^{k+1} = W_{ij}^k + \frac{1}{\rho_x}(X_{ij}^{k'+1} - X_{ij}^{k'})$
8: Update $\bar{X}_i^{k'+1} = \frac{1}{4}\sum_{i=1}^4 X_i^{k'}$
9: $k' = k' + 1$
10: **until** $\sum_{j=1}^4 \|X_{ij}^{k'} - \bar{X}^{k'}\|_F^2 \le \epsilon_c$, $4{\rho_x}^2\|\bar{X}^{k'+1} - \bar{X}^{k'}\|_F^2 \le \epsilon_o$
Output: $X_i^k = \bar{X}_i^{k'}$

To achieve a better representation and avoid abuse of notations, we use k' to denote the local iteration index of local optimization. The main details of the process in Algorithm 3 are presented in (23)–(26), where we solve the local consensus problem in parallel

$$\begin{aligned}X_{i0}^{k'+1} = &\arg\min_{X_{i0}} -f(X_{i0}) + (\rho|\mathcal{N}_i| - \mu)\Big\|X_{i0} - Z_i^k\Big\|_F^2 \\ &+ \frac{\rho'}{2}\|X_{i0} - \bar{X}_i^{k'} + \frac{1}{\rho}W_0^{k'}\|_F^2\end{aligned} \tag{23}$$

$$X_{i1}^{k'+1} = \arg\min g_1(X_{i1}) + \frac{\rho'}{2}\|X_{i1} - \bar{X}_i^{k'} + \frac{1}{\rho}W_1^{k'}\|_F^2 \tag{24}$$

$$X_{i2}^{k'+1} = \arg\min g_2(X_{i2}) + \frac{\rho'}{2}\|X_{i2} - \bar{X}_i^{k'} + \frac{1}{\rho}W_2^{k'}\|_F^2 \tag{25}$$

$$X_{i3}^{k'+1} = \arg\min g_3(X_{i3}) + \frac{\rho'}{2}\|X_{i3} - \bar{X}_i^{k'} + \frac{1}{\rho}W_3^{k'}\|_F^2. \tag{26}$$

For (24)–(26), we achieve the solutions by Euclidean projections because $\{g_p(X)\}$ are indicator functions on convex sets. For example, $\Pi_{\geq 0}(V)$ returns the Euclidean projections of matrix $V \in \mathbb{R}^{M\times A}$ on a proper cone of $R_+^{M\times A}$, and

$$\Pi_{\geq 0}(V) = V_+ \tag{27}$$

where $V_+ = \max(V, 0)$. For (25) and (26), we denote $X = [x_1, \ldots, x_A]$ as a combination of vectors, i.e., $x = [x_1^\top, \ldots, x_A^\top]^\top$. Constraints of $1_M^\top X = 1_A^\top$ and $X 1_A \geq 1_M$ are equivalent with respect to x as follows:

$$P_1 x = 1_A \quad P_2 x \geq 1_M \tag{28}$$

where $P_1 \in \mathbb{R}^{A\times MA}$ and $P_2 \in \mathbb{R}^{A\times MA}$. P_1 is a diagonal matrix and its diagonal elements are $1_M^\top$. The jth row of P_2 contains A copies of $e_j^\top$. The projections in (25) and (26) are, therefore, solved by

$$x_{i2} = \left[x_1^\top, \ldots, x_A^\top\right]^\top = \Pi_{P_1 x = 1_A} \tag{29}$$

$$x_{i3} = \left[x_1^\top, \ldots, x_A^\top\right]^\top = \Pi_{P_2 x \geq 1_M} \tag{30}$$

where $\Pi_{Ax=b}(v)$ and $\Pi_{Ax\geq b}(v)$ are Euclidean projections on the polyhedra and can be easily solved as shown in [49].

For (23), we introduce a quadratic proximal operator of the matrix variable here. Given X and its quadratic scalar function $f(X) = \mathrm{tr}(Q^\top X) + \frac{\mu}{2}\|X - U\|_F^2$, the proximal operator can be defined as $\mathrm{Prox}_{\rho,\mu,Q,U}(V)$, and

$$\begin{aligned}\mathrm{Prox}_{\rho,\mu,Q,U}(V) &= \arg\min_X f(X) + \frac{\rho}{2}\|X - V\|_F^2 \\ &= \arg\min_X \mathrm{tr}\left(Q^\top X\right) + \frac{\mu}{2}\|X - U\|_F^2 + \frac{\rho}{2}\|X - V\|_F^2\end{aligned}$$

where ρ is the parameter and $\rho > 0$. When $\mu > -\rho$, it can be obtained analytically as

$$\mathrm{Prox}_{\rho,\mu,Q,U}(V) = \frac{1}{\mu + \rho}(\mu U + \rho V - Q). \tag{31}$$

Notice that we chose the parameter $\mu < \rho|\mathcal{N}_i|$, which ensures $(\rho|\mathcal{N}_i| - \mu) + \frac{\rho'}{2} > 0$. Thus, (23) can be obtained by $\mathrm{Prox}_{\rho',(\rho|\mathcal{N}_i|-\mu),Q_x,U_x}(V_x)$, and $Q_x = -B \circ [C^\top \Lambda(1_{S\times M} - D^k)]^\top \circ E_i$, $U_x = Z_i^k$, $V_x = \bar{X}_i^{k'} - \frac{1}{\rho}W_0^{k'}$.

2) Optimization of D: This step optimizes D^{k+1} when X_i^{k+1} is obtained. According to (19), the local optimization problem of D^{k+1} becomes

$$D^{k+1} = \arg\min_D \sum_{i=1}^{A} f_i\left(X_i^{k+1}, D\right) + \sum_{q=1}^{2} h_q(D). \tag{32}$$

Similarly, we solve (32) by splitting D into three parts $\{D_1, D_2, D_3\}$ and apply a consensus solver to calculate D^{k+1} as shown in Algorithm 4.

The main processes of Algorithm 4 are represented in (33)–(35) as follows:

$$D_0^{k'+1} = \arg\min_{D_0} \sum_{i=1}^{A} f_i\left(X_i^{k+1}, D_0\right) + \frac{\rho_d}{2}\|D_0 - D_0'\|_F^2 \tag{33}$$

$$D_1^{k'+1} = \arg\min_{D_1} h_1(D_1) + \frac{\rho}{2}\left\|D_1 - \bar{D}^{k'} + \frac{1}{\rho}W_1^{k'}\right\|_F^2 \tag{34}$$

$$D_2^{k'+1} = \arg\min_{D_2} h_2(D_2) + \frac{\rho}{2}\left\|D_2 - \bar{D}^{k'} + \frac{1}{\rho}W_2^{k'}\right\|_F^2 \tag{35}$$

Algorithm 4 Update of D^{k+1}

Input: X_i^{k+1}, augment parameter ρ_d, primal and dual error bounds ϵ_c, ϵ_o
1: Initialization: D_1^1, D_2^1, D_3^1
2: **repeat**
3: Update $D_1^{k'+1}$ based on (33)
4: Update $D_2^{k'+1}$ based on (34)
5: Update $D_3^{k'+1}$ based on (35)
6: Update $W_j^{k'+1} = W_j^{k'} + \frac{1}{\rho_d}(D_j^{k'+1} - D_j^{k'})$
7: Update $\bar{D}^{k'+1} = \frac{1}{3}\sum_{i=0}^{2} D_i^{k'}$
8: $k' = k' + 1$
9: **until** $\sum_{i=0}^{2}\|D_i^{k'} - \bar{D}^{k'}\|_F^2 \leq \epsilon_c$, $3\rho_d^2\|\bar{D}^{k'+1} - \bar{D}^{k'}\|_F^2 \leq \epsilon_o$
Output: $D^{k+1} = \bar{D}^{k'}$

where $D_0' = \bar{D}^{k'} + (1/\rho_d)W_0^{k'}$. And (33) can be calculated by the proximal operator in (31), i.e.,

$$D_0^{k'+1} \triangleq \mathrm{Prox}_{\rho_d,0,Q_d,U_d}\left(\bar{D}^{k'} - \frac{1}{\rho_d}W_0^{k'}\right) \tag{36}$$

where $Q_d = -\Lambda C^\top [B \circ (\sum_{i=1}^{A} E_i \circ X_i)]^\top$ and $U_d = 0_{M\times A}$. We denote $d = [d_1^\top, \ldots, d_M^\top]^\top \in \mathbb{R}^{S\times M}$ and solve (34), (35) by vector Euclidean projections on $\mathbb{R}_+^{S\times M}$ and $P_3 d = 1_M$ as in

$$D_1^{k'+1} = \Pi_{\geq 0}\left(\bar{D}^{k'} - \frac{1}{\rho}W_1^{k'}\right) \tag{37}$$

$$D_2^{k'+1} = \Pi_{1_S^\top D_2 = 1_M^\top}\left(\bar{D}^{k'} - \frac{1}{\rho}W_2^{k'}\right). \tag{38}$$

E. Performance Analysis

We seek the convergence of either global or local optimality as in the works in [40], where an explicit proof of two block ADMM can be found. However, the convergence of multiblock ADMM is still an open problem even when the objective function is strictly convex or concave. However, the multiblock ADMM continues to function and exhibits good convergence even in most nonconvex optimization situations. Strong convexity has been demonstrated to be a fundamental requirement for convergence in [50], which is unsuitable for our linear objective function. On the contrary, we propose a penalty component and its tunable parameters to ensure that each objective function is still convex in order to guarantee the performance of our approaches. We also explain the underlying principles and the relevance of the parameter values, as indicated in Section V.

V. Numerical Simulations

We conduct numerical simulations in this part to show how well our algorithms perform in adversarial decision making for assignment tasks. Consider a group of ten robots that can identify five targets using their onboard sensors. Each target is detectable by two separate types of sensors, each of which can simultaneously be disabled. This configuration makes sense in practical applications like a target detection mission using the cameras and infrared sensors.

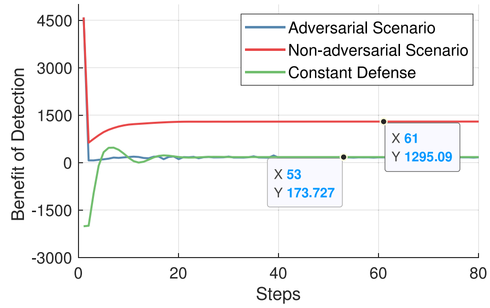


Fig. 3. Benefit of detection in three scenarios.

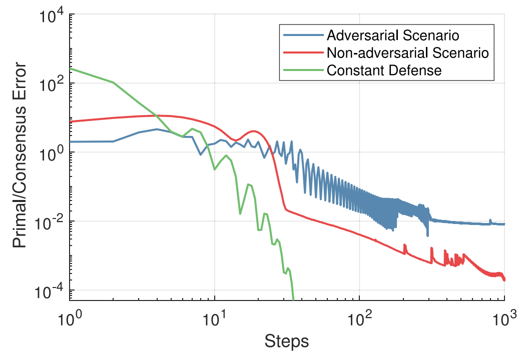


Fig. 4. Primal/consensus error $\sum_{i=1}^{A} \|X_i^k - \bar{X}^k\|$ in three scenarios.

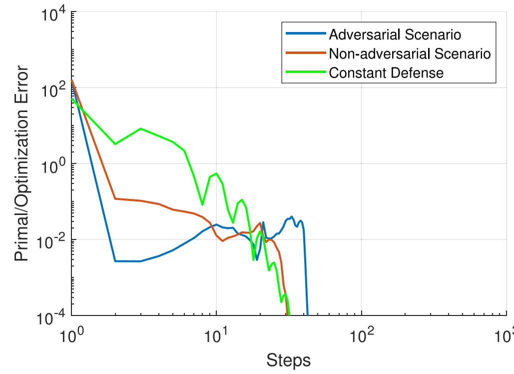


Fig. 5. Dual/optimization error $A\rho^2\|\bar{X}_i^{k+1} - \bar{X}^k\|$ in three scenarios.

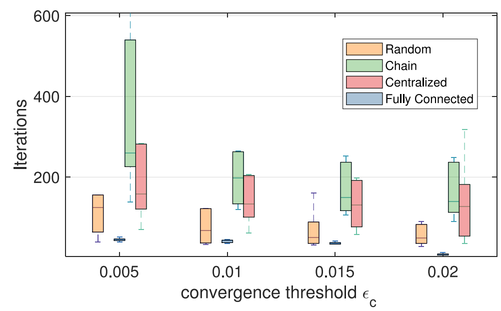


Fig. 6. Computational iterations in the main loop using different communication topologies.

We intend to answer the following two questions: 1) Does our algorithm provide an optimal solution in adversarial scenarios? and 2) How should our algorithm be applied in real applications? The first question is related to the comparison with centralized methods in nonadversarial scenarios. It should be highlighted that, in contrast to the best assignment solutions with no interference, the maximal benefit of detection is lower in adversarial settings. As a result, we additionally display the outcomes of the current target selection interference. In order to respond to the second question, we validate our algorithm by applying different conditions, such as parameter selection and communication topology.

A. Convergence and Optimality

To validate the performance of our algorithms in adversarial scenarios, we test on three scenarios, including: 1) *Adversarial Scenario:* the targets choose the best interference according to the assignment action of MAS; 2) *Nonadversarial Scenario:* all targets cannot avoid being detected by onboard sensors; and 3) *Constant Interference:* the interference decision of targets is fixed. We solve the decision-making problems using our method in Algorithm 2 for adversarial scenario. In comparison, we use the distributed decision-making methods proposed in [29] in nonadversarial scenario and constant interference. The performance criterion is defined as the benefit of detection and the convergence error versus the optimization iterations. The results are shown in Figs. 3–5.

It is worth noting that the benefit of detection also represents the dual residual. The consensus error can be explained as the primal residual of the global consensus constraints. The convergence of both errors shows that we have attained the optimality of the detection benefit function $f(X, D)$. The convergence of the primal residual, or the achievement of global consistency, is indicated by the consensus error's continuous reduction. The consensus error is defined as $\sum_{i=1}^{A} \|X_i^k - \bar{X}^k\|$ in Fig. 4, where $\bar{X}^k$ is the average value of all X_i^k at iteration k. In fact, X_i^k contains the local assignment action and the corresponding estimation of the neighbors. The average value $\bar{X}^k$ can be viewed as the true value of the global assignment action. Suppose that the consensus error equals to zero, it means that the estimation of assignment action is identical to the true values for all neighbors. The consensus error in Fig. 4 shows that the estimation of the neighbors is continuously approaching the global convergence threshold condition.

We observe that our algorithm converges to the optimal solution with acceptable error bounds and convergence rate. The lower convergence rate occurs because the defense policy of the targets is also updated to minimize the benefit of detection, which inevitably causes convergence oscillation. There exists a sharp decrease in the benefit of detection compared to nonadversarial scenarios because of the interference. We also show the results under constant but optimal interference of targets, where the optimality is verified by identical convergence. Compared to the distributed optimization methods in [29], our algorithm is able to resolve the decision-making problem that is focused on the target assignment in an adversarial setting.

B. Influence of Communication Topology

Each robot interacts with all of its neighbors throughout the optimization phase to obtain the temporary primal and dual

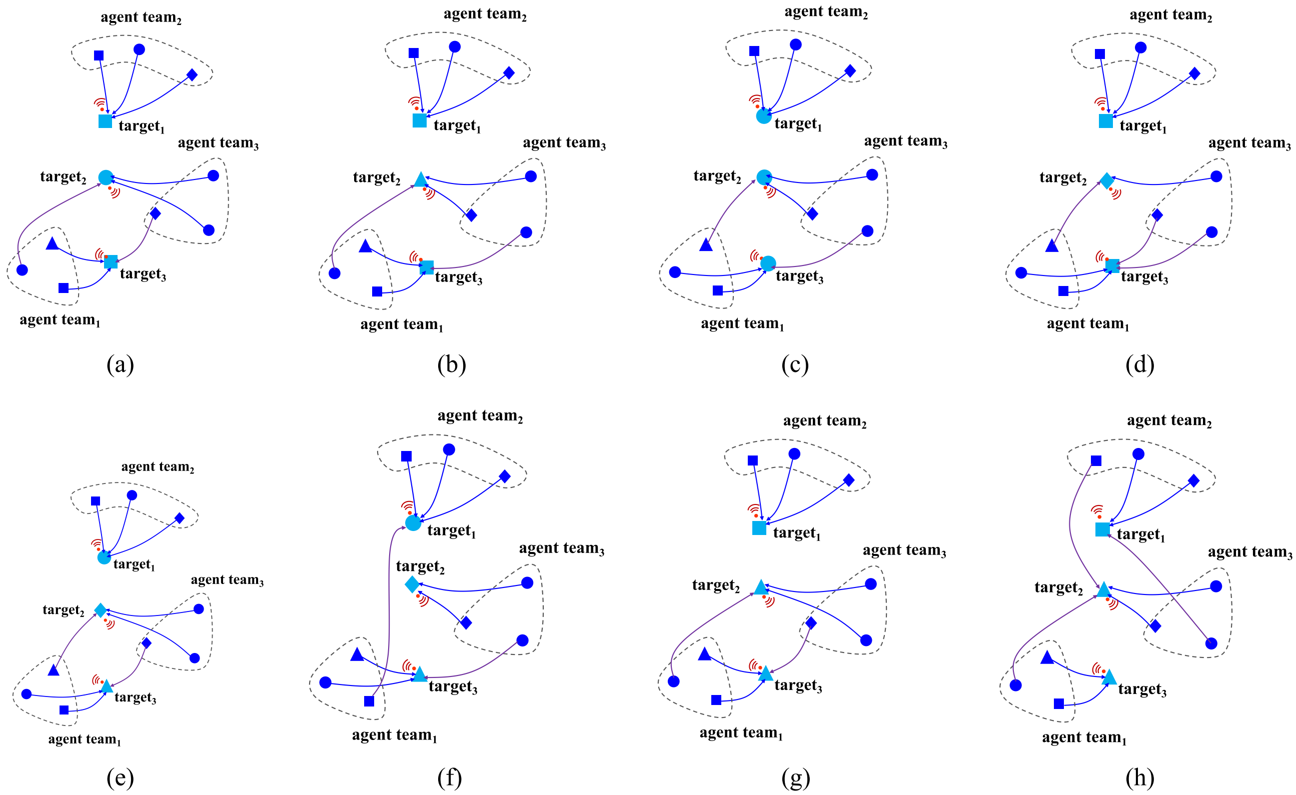


Fig. 7. We compare eight different choices of Λ and show the assignment solutions. The purple arrows represent the actions of modification compared to the initial assignment with constant defense. The shapes of the targets are also determined by the optimal defense policy. (a) $\lambda = (1, 1, 1, 1)$. (b) $\lambda = (1, 100, 1, 1)$. (c) $\lambda = (1, 1, 100, 1)$. (d) $\lambda = (1, 1, 1, 100)$. (e) $\lambda = (1, 10, 10, 10)$. (f) $\lambda = (1, 100, 100, 100)$. (g) $\lambda = (10, 10, 1, 1)$. (h) $\lambda = (100, 100, 1, 1)$.

variables as well as all errors. Therefore, a more straightforward communication architecture will result in a slower rate of convergence. We evaluate our algorithm's performance using four distinct communication topologies.

1) *Random:* We generate the communication topology randomly and the expectation of edges is set to be 25.
2) *Chain:* Robots are connected in a linear chain.
3) *Centralized:* Only one robot is selected to be able to communicate with other robots.
4) *Fully Connected:* Every pair of robots is connected.

We can conclude from Fig. 6 that our approach performs well with any connected communication topology, but that the convergence rates vary. Under the fully connected topology, we attain the faster convergence rate because information is effectively spread among robots. Chain topology presents the worst convergence rate since updated information needs to spread from one end to the other.

C. Influence on Parameters

MAS with diverse sensors outperforms homogeneous ones in practical applications because all agents work in concert to diversity the detection. Thus, we also provide the performance of our algorithm for various detection ability settings in Fig. 7. We conclude from the findings that the targets are likely to neutralize stronger sensors. The assignment action of MAS is also influenced by the decay coefficients related to the distance between the agents and targets in extreme circumstances where all targets have the same interference decisions.

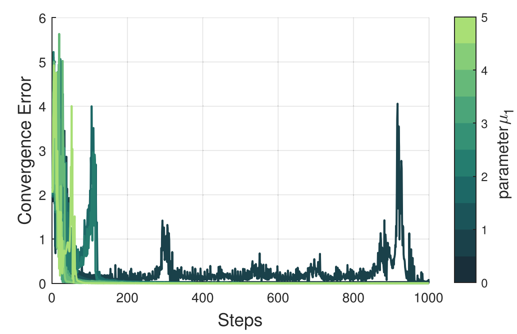


Fig. 8. Consensus errors under different parameters of μ.

The selection of parameter μ is crucial because they penalize the assignment solutions X_i that locate within $(0, 1)$ and push all elements of X_i toward $\{0, 1\}$. On the contrary, the convergence cannot be guaranteed if μ is set too large to violate the convex conditions of the proximal operator. Here, we show the relationship of μ and the convergence error of each X_i in Fig. 8. We found that the failure of convergence occurs if $\mu = 0$. The main reason is that multiple optimal solutions exist if each element is not restricted to $\{0, 1\}$. And the convergence rate is rather higher with larger μ.

D. Computational Complexity

The main process of our algorithm is composed of the updates of X_i^{k+1}, D^{k+1}, and Y_{ij}^{k+1}. The number of required iterations is mainly decided by the communication topology

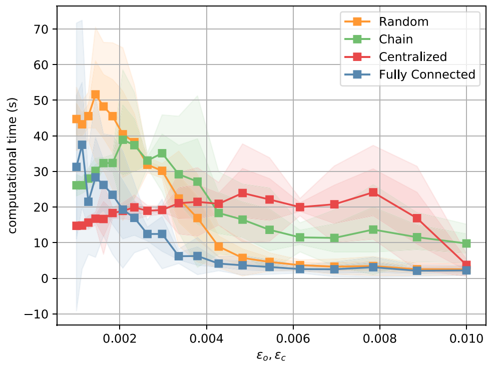


Fig. 9. Average computational time when increasing the error bounds ϵ_c and ϵ_o. We set $\epsilon_c = \epsilon_o$ under all scenarios. The comparison is performed on a platform with Intel i7 10700F CPU (2.9 GHz). We also show the variance generated by the random initialization of the algorithms.

and the stopping criteria, which is decided by the primal and dual error bounds, i.e., ϵ_c and ϵ_o. The optimization steps of X_i^{k+1} and D^{k+1} both have the computational complexity of $\mathcal{O}(AM)$ and $\mathcal{O}(MS)$ caused by the Frobenius norm as shown in Algorithms 3 and 4 separately. We test the computational complexity of our algorithms by showing the computational time with respect to different primal and dual error bounds under four different topologies in Fig. 9. It can be learned that the computational time is most decided by the topology, which coincides with the required iterations. The error bounds will also influence the computational time directly. As shown in Fig. 9, we give the computational time with setting $\epsilon_c, \epsilon_o \in [10^{-1}, 10^{-3}]$. In real applications, we recommend ϵ_c as $[0.05^2 MA, 0.1^2 MA]$ such that to achieve a tradeoff between an acceptable average error and computational time.

The computational time of all topologies will less than 10 s when ϵ_c and ϵ_o are both below than 0.01. It means that the difference of assignment actions of all agents is less than 0.01, which is a strict criterion. The convergence can be achieved within 1 s except for chain topology, which is a rare condition in real applications. It is also worth noting that the centralized topology has fewer computational iterations which are shown in Fig. 6 but larger computational time in Fig. 9. This is because more steps are required to achieve the optimal X_i^{k+1} and D^{k+1} in the central agent.

VI. Conclusion

In this article, an adversarial decision-making method for MAS is proposed to reduce the impact of uncertain defense policy in the process of observing a team of targets. We formulate the adversarial decision-making problem for target assignment as a max–min problem and solve it in parallel using multiblock ADMM, which is particularly practical for deployment in the distributed MAS. It can be concluded from the simulation results that the multiblock ADMM can be modified and used in our adversarial decision-making problem even without an explicit guarantee of convergence. We can empirically ensure the effectiveness of multiblock ADMM by tuning the consistency penalty term and preserving the convexity of the objective function in each local block. It also offers the promising potential for the multiagent decision-making applications considering the convergence rate in our simulation. Multiblock ADMM provides a promising tool for the distributed optimization with a split objective function, such as a linear function in every part without the strong convex assumptions. Besides, a balance between the cost and the benefit of communication among MAS is supposed to be designed carefully such as the topology, which influences the coordination of MAS. Currently, we assume that the updated information of each robot is transmitted synchronously. How to handle asynchronous systems and communication delays will be our future work.

References

[1] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Trans. Cogn. Devel. Syst.*, vol. 10, no. 3, pp. 514–529, Sep. 2018.

[2] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *Int. J. Robot. Res.*, vol. 38, nos. 2–3, pp. 316–337, 2019.

[3] Y. Xu and Z.-G. Wu, "Data-based collaborative learning for multi-agent systems under distributed denial-of-service attacks," *IEEE Trans. Cogn. Devel. Syst.*, early access, May 6, 2022, doi: 10.1109/TCDS.2022.3172937.

[4] L. Jin, S. Li, H. M. La, X. Zhang, and B. Hu, "Dynamic task allocation in multi-robot coordination for moving target tracking: A distributed approach," *Automatica*, vol. 100, pp. 75–81, Feb. 2019.

[5] S. C. Pinto, S. B. Andersson, J. M. Hendrickx, and C. G. Cassandras, "Optimal multi-agent persistent monitoring of the uncertain state of a finite set of targets," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, 2019, pp. 4280–4285.

[6] A. Mir and A. Khachane, "Sensing harmful gases in industries using IoT and WSN," in *Proc. 4th Int. Conf. Comput. Commun. Control Autom. (ICCUBEA)*, 2018, pp. 1–3.

[7] R. K. Ramachandran, P. Pierpaoli, M. Egerstedt, and G. S. Sukhatme, "Resilient monitoring in heterogeneous multi-robot systems through network reconfiguration," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 126–138, Feb. 2022.

[8] J. Zhang, L. Jin, and C. Yang, "Distributed cooperative kinematic control of multiple robotic manipulators with improved communication efficiency," *IEEE Trans. Mechatronics*, vol. 21, no. 1, pp. 149–158, Feb. 2022.

[9] X. Hou, H. Zhao, C. Wang, and H. Liu, "Knowledge driven indoor object-goal navigation aid for visually impaired people," *Cogn. Comput. Syst.*, vol. 4, no. 4, pp. 329–339, 2022.

[10] H. Tang, R. Yan, and K. C. Tan, "Cognitive navigation by neuro-inspired localization, mapping, and episodic memory," *IEEE Trans. Cogn. Devel. Syst.*, vol. 10, no. 3, pp. 751–761, Sep. 2018.

[11] H. Zhao, H. Wang, W. Wu, and J. Wei, "Deployment algorithms for UAV airborne networks toward on-demand coverage," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2015–2031, Sep. 2018.

[12] G. H. Tzeng and C.-H. Chen, "Multiobjective decision making for traffic assignment," *IEEE Trans. Eng. Manag.*, vol. 40, no. 2, pp. 180–187, May 1993.

[13] F. Khemakhem, H. Ellouzi, H. Ltifi, and M. B. Ayed, "Agent-based intelligent decision support systems: A systematic review," *IEEE Trans. Cogn. Devel. Syst.*, vol. 14, no. 1, pp. 20–34, Mar. 2022.

[14] V. M. Lo, "Heuristic algorithms for task assignment in distributed systems," *IEEE Trans. Comput.*, vol. 37, no. 11, pp. 1384–1397, Nov. 1988.

[15] A. Salman, I. Ahmad, and S. Al-Madani, "Particle swarm optimization for task assignment problem," *Microprocess. Microsyst.*, vol. 26, no. 8, pp. 363–371, 2002.

[16] L. Luo, N. Chakraborty, and K. Sycara, "Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 19–30, Feb. 2015.

[17] G. Neville, A. Messing, H. Ravichandar, S. Hutchinson, and S. Chernova, "An interleaved approach to trait-based task allocation and scheduling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 1507–1514.

[18] N. A. M. Subha and M. N. Mahyuddin, "Distributed adaptive cooperative control with fault compensation mechanism for heterogeneous multi-robot system," *IEEE Access*, vol. 9, pp. 128550–128563, 2021.

[19] C. Hajaj and Y. Vorobeychik, "Adversarial task assignment," 2018, *arXiv:1804.11221*.

[20] N. Seenu, R. M. K. Chetty, M. Ramya, and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Ind. Robot Int. J. Robot. Res. Appl.*, vol. 47, no. 6, pp. 929–942, 2020.

[21] L. Zhou and P. Tokekar, "Sensor assignment algorithms to improve observability while tracking targets," *IEEE Trans. Robot.*, vol. 35, no. 5, pp. 1206–1219, Oct. 2019.

[22] 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.

[23] E. Montijano, D. Tardioli, and A. R. Mosteo, "Distributed dynamic sensor assignment of multiple mobile targets," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2019, pp. 4921–4926.

[24] E. Suslova and P. Fazli, "Multi-robot task allocation with time window and ordering constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 6909–6916.

[25] A. Ray, A. Pierson, H. Zhu, J. Alonso-Mora, and D. Rus, "Multi-robot task assignment for aerial tracking with viewpoint constraints," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 1515–1522.

[26] M. Chen and D. Zhu, "A workload balanced algorithm for task assignment and path planning of inhomogeneous autonomous underwater vehicle system," *IEEE Trans. Cogn. Devel. Syst.*, vol. 11, no. 4, pp. 483–493, Dec. 2019.

[27] W. Du, Y. Tang, S. Y. S. Leung, L. Tong, A. V. Vasilakos, and F. Qian, "Robust order scheduling in the discrete manufacturing industry: A multiobjective optimization approach," *IEEE Trans. Ind. Informat.*, vol. 14, no. 1, pp. 253–264, Jan. 2018.

[28] F. Lagum, I. Bor-Yaliniz, and H. Yanikomeroglu, "Strategic densification with UAV-BSs in cellular networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 384–387, Jun. 2018.

[29] R. N. Haksar, O. Shorinwa, P. Washington, and M. Schwager, "Consensus-based ADMM for task assignment in multi-robot teams," in *Proc. Int. Symp. Robot. Res.*, vol. 20, 2019, pp. 35–51.

[30] F. Cacciamani, A. Celli, M. Ciccone, and N. Gatti, "Multi-agent coordination in adversarial environments through signal mediated strategies," 2021, *arXiv:2102.05026*.

[31] W. Luo, J. Lü, K. Liu, and L. Chen, "Learning-based policy optimization for adversarial missile-target assignment," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 7, pp. 4426–4437, Jul. 2021.

[32] B. Reily and H. Zhang, "Team assignment for heterogeneous multi-robot sensor coverage through graph representation learning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 838–844.

[33] H. Park and S. Hutchinson, "Worst-case performance of a mobile sensor network under individual sensor failure," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2013, pp. 895–900.

[34] R. Schwartz and P. Tokekar, "Robust multi-agent task assignment in failure-prone and adversarial environments," 2020, *arXiv:2007.00100*.

[35] X. Yi, A. Zhu, S. X. Yang, and C. Luo, "A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 974–983, Apr. 2017.

[36] X. Ren and Y. Mo, "Secure detection: Performance metric and sensor deployment strategy," *IEEE Trans. Signal Process.*, vol. 66, no. 17, pp. 4450–4460, Sep. 2018.

[37] F. Yang and N. Chakraborty, "Algorithm for multi-robot chance-constrained generalized assignment problem with stochastic resource consumption," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2020, pp. 4329–4336.

[38] M. Rudolph, S. Chernova, and H. Ravichandar, "Desperate times call for desperate measures: Towards risk-adaptive task allocation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 2592–2597.

[39] Z. Wang, S. Lian, Y. Zhang, X. Cui, R. Yan, and H. Tang, "Towards lossless ANN-SNN conversion under ultra-low latency with dual-phase optimization," 2022, *arXiv:2205.07473*.

[40] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[41] S.-S. Park, Y. Min, J.-S. Ha, D.-H. Cho, and H.-L. Choi, "A distributed ADMM approach to non-myopic path planning for multi-target tracking," *IEEE Access*, vol. 7, pp. 163589–163603, 2019.

[42] Y. Li, H. Zhao, H. Liu, S. Lu, and Y. Hou, "Research on visual-tactile cross-modality based on generative adversarial network," *Cogn. Comput. Syst.*, vol. 3, no. 2, pp. 131–141, 2021.

[43] Y. Li, H. Zhao, H. Liu, S. Lu, and Y. Hou, "Research on visual-tactile cross-modality based on generative adversarial network," *Cogn. Comput. Syst.*, vol. 3, no. 2, pp. 131–141, 2021.

[44] L. Zhou, Y. Yue, M. Zhong, and F. Jin, "High-speed railway rolling stock scheduling based on ADMM decomposition algorithm," in *Proc. 2nd Int. Conf. Ind. Artif. Intell. (IAI)*, 2020, pp. 1–6.

[45] J. Lin, Y. Li, and Q. Peng, "Joint power allocation, base station assignment and beamformer design for an uplink SIMO heterogeneous network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2014, pp. 434–438.

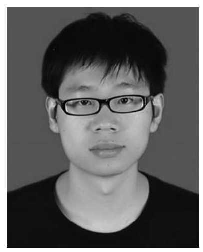
[46] H. Chen, O. Dogru, S. K. Varanasi, X. Yin, and B. Huang, "Distributed process monitoring for multi-agent systems through cognitive learning," *IEEE Trans. Cogn. Devel. Syst.*, early access, Oct. 12, 2022, doi: 10.1109/TCDS.2022.3214424.

[47] B. Reily, T. Mott, and H. Zhang, "Adaptation to team composition changes for heterogeneous multi-robot sensor coverage," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2021, pp. 9051–9057.

[48] Z. Sun et al., "An evolutionary algorithm with constraint relaxation strategy for highly constrained multiobjective optimization," *IEEE Trans. Cybern.*, early access, Mar. 11, 2022, doi: 10.1109/TCYB.2022.3151974.

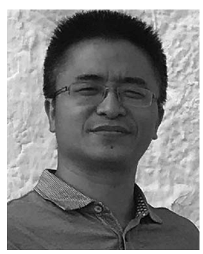
[49] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.

[50] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the ADMM with multi-block variables," *SIAM J. Optim.*, vol. 25, no. 3, pp. 1478–1497, 2015.



Yang Li received the B.Eng. degree in automatic control and the Ph.D. degree in armament science and technology from Northwestern Polytechnical University, Xi'an, China, in 2011 and 2021, respectively.

He is currently a Postdoctoral Fellow with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. His current research interests include the decision making for multiagent system.



Huaping Liu (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2004.

He is a Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include robot perception and learning.

Dr. Liu was a recipient of the National Science Fund for Distinguished Young Scholars. He served as the Area Chair of Robotics Science and Systems for several times. He is currently a Senior Editor of *International Journal of Robotics Research*.

Fuchun Sun received the Ph.D. degree from Tsinghua University, Beijing, China, in 1997.

He is currently a Full Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include intelligent control and robotics.

Prof. Sun was a recipient of the National Science Fund for Distinguished Young Scholars. He serves as the Editor-in-Chief for *Cognitive Computation and Systems* and an Associate Editor for a series of international journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Mechatronics*, and *Robotics and Autonomous Systems*.

Zhihong Chen received the Ph.D. degree from Beijing Jiaotong University, Beijing, China, in 2015.

He is currently a Senior Engineer with China Aerospace Science and Technology Corporation, Beijing. His research focuses on kinematic and dynamical modeling and trajectory tracking controller designing for a mobile robot.