A Decentralized Multirobot Spatiotemporal Multitask Assignment Approach for Perimeter Defense

Shridhar Velhal, Suresh Sundaram, *Senior Member, IEEE*, and Narasimhan Sundararajan, *Life Fellow, IEEE*

***Abstract*—This article provides a new decentralized approach to solve a perimeter defense problem (PDP). In a typical PDP, many intruders try to enter a territory, and a group of defenders operating both inside and on the perimeter protects the territory by capturing the intruders on the perimeter. The objective of the defenders is to detect and capture the intruders before they enter the territory. Defenders sense the intruders independently and compute their trajectories to capture all the intruders in a cooperative way. Each intruder is estimated to reach a specific location on the perimeter at a specific time, and this is considered as a spatiotemporal task to be handled by a defender. At any given time, the PDP is converted to a decentralized multirobot spatiotemporal multitask assignment (DMRST-MTA) problem. The cost of executing a task for a defender is defined by a composite cost function that includes both the spatial and temporal cost components. In this article, a modified decentralized consensus-based bundle algorithm is presented to solve the above spatiotemporal multitask assignment problem. The performance evaluation of the proposed approach is presented based on Monte Carlo studies, and the results show the effectiveness of the proposed approach under different scenarios. The robustness studies also show that the proposed approach is robust against uncertainties in the heading angles of the intruders. The performance comparison with the decentralized adaptive partitioning approach for the various arrival distributions of intruders clearly shows that DMRST-MTA is efficient.**

***Index Terms*—Consensus-based bundled auction algorithm, dynamic multitask assignment, perimeter defense problem (PDP), spatiotemporal task.**

I. Introduction

RAPIDLY evolving technologies in the autonomous operations of robots and associated developments in low-cost sensors have created significant interest among researchers in using them for various civil and military applications. Serious security, safety, and privacy issues arise when a freely operating robot intrudes and operates in a critical area (such as nuclear facilities, airports, chemical industries, ports, etc.) [1]. Recently, significant research efforts have been made to develop efficient territory protection systems to ensure the safety, security, and privacy of the protected area. A territory protection system mainly has to handle two critical aspects, namely, an early detection of the intruders and then neutralization of them.

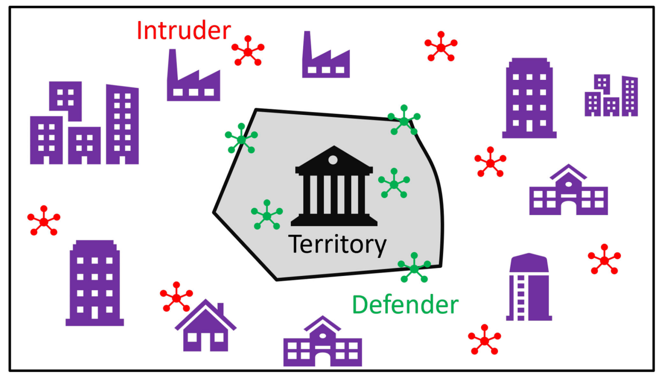


Fig. 1. Typical scenario for the protection of critical infrastructure.

Fig. 1 shows a typical scenario for protecting a critical infrastructure from a group of intruders. The surrounding area around any critical infrastructure is called a territory and is represented by a convex gray region in Fig. 1. A set of robots operating in the territory (called the defenders, illustrated in "green color" in Fig. 1) protects the territory by capturing the intruders before any intruder enters the territory. The intruders (represented in red) try to enter the territory. Defenders operate both inside the territory and on the perimeter. One should note that the number of intruders is not constant and can vary over time. In summary, the perimeter defense problem (PDP) is defined as a multiplayer problem, where a group of defenders cooperatively operates and captures the intruders on the perimeter with the aim of protecting the critical infrastructure inside the territory [2]–[6]. First, we present a brief review of some of the key existing works in the area of PDP.

A multirobot perimeter patrol problem for adversarial conditions was proposed in [7]. The nondeterministic patrol algorithm was employed for patrolling the territory in an adversarial setting. The path defense approach proposed in [8] provides a strategy for a one-to-one scenario, where one defender protects the target from one intruder while operating only on a predefined path using a fast-marching method. This constraint of predefined path limits the defender's movements.

In [9], a PDP was proposed to protect a straight-line boundary. Here, the path of a single defender was obtained to capture the

Manuscript received July 20, 2021; revised July 20, 2021 and December 15, 2021; accepted February 21, 2022. This article was recommended for publication by Associate Editor Jason O'Kane and Editor Paolo Robuffo Giordano upon evaluation of the reviewers' comments. *(Corresponding author: Suresh Sundaram.)*

The authors are with the Department of Aerospace Engineering, Indian Institute of Science, Bengaluru 560012, India (e-mail: velhalb@iisc.ac.in; vssuresh@iisc.ac.in; ensundara@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TRO.2022.3158198.

Digital Object Identifier 10.1109/TRO.2022.3158198

maximum number of intruders with constant velocities arriving at the boundary. Later, this was extended to a circular territory with radially incoming intruders in [10]. Recently, in [11], multiple defenders with an adaptive partitioning (AP) of the circular territory have been proposed to capture radially incoming intruders. These works are limited to the circular territory with radially traversing intruders.

In [2], a geometric method was proposed to compute a feasible region for defenders. Next, a "maximum-matching algorithm" was used to assign an intruder to a specific defender. Since the "maximum-matching" algorithm assumes that each defender plays a game independently, the solution may lead to the noncapture of some of the intruders for a general problem. In [3], a cooperative defense strategy was proposed to overcome the above noncapture nature of the maximum-matching algorithm. First, a local game region is defined to form a team of defenders who can then capture multiple intruders in that local region. Here, the defenders have to cooperate with other defenders in the local region to neutralize an intruder. The approach is valid only if the number of defenders is greater than the number of intruders. In [4], patroller robots operating outside the perimeter were introduced to sense the intruders early and share the information with the neighboring defenders. A centralized approach has been used here to assign a defender to an intruder. In addition, an analytical condition on the number of patrollers and defenders required to defend a territory and a bound on the interagent separation for better detection of intruders has been presented. In [12], the PDP was extended to the case where an aerial defender neutralizes a ground intruder. Furthermore, the aforementioned works employ only a centralized solution approach for solving the PDP and are not easily scalable.

Recently, a decentralized approach for the PDP with multirobot sensing and communication has been presented in [5]. The decentralized policy for communication and decision-making is learned using a centralized solution. The action space for the defenders is restricted to either a clockwise or an anticlockwise motion. The reinforcement-learning-based solution gets complicated when the defender moves in any direction and when the number of defenders is lower than that of the intruders. A complete and detailed review of works in the area of PDP was presented in [6]. It pointed out the limitations of the current literature, which include defender dynamics, sequential capture, fast intruders, and partial information. Based on the above review, it may be noted that there is still a strong need to develop a decentralized approach to overcome the aforementioned limitations for a general PDP problem.

In this article, we present a consensus-based decentralized multirobot spatiotemporal multitask assignment (DMRST-MTA) approach to solve the PDP. It uses a sequential capture approach to handle a varying number of intruders under partial observability conditions. The proposed method can handle a larger number of intruders using a smaller number of defenders. Defenders are equipped with sensors to detect the intruders, and the defenders cooperate with each other to capture the intruders. Here, the problem of capturing a varying number of intruders is first converted into a spatiotemporal multitask assignment (ST-MTA) problem. The estimated arrival times and locations of the intruders at any given time are used to define the spatiotemporal tasks. These spatiotemporal tasks are then assigned to the defenders using a decentralized ST-MTA approach.

The consensus-based bundle algorithm (CBBA) for the 1-D multiple-task assignment problem, presented in [13], is extended here to handle the 2-D spatiotemporal multiple-task assignment problem. The modified CBBA algorithm provides a decentralized scalable auction-based solution for one-to-many task assignment problem. The modified CBBA is used as a building block for DMRST-MTA, where every defender bids for neutralizing the observed intruders. The bidding cost is computed for every defender's chosen trajectory, and a composite loss function is then used to unite both the spatial and temporal components. A consensus is then formed based on partial communication among the defenders. The task assignment output defines a trajectory for each defender to neutralize their assigned intruders.

The performance of the proposed DMRST-MTA has been evaluated using simulations. Monte Carlo simulations have been conducted for analyzing the success rates by varying the speeds of defenders, sensing radii, and temporal separations between the intruders. As the speed of the defender increases, the success percentage of neutralizing the intruders also increases. The increasing sensing radius provides more planning time for a defender, and hence, this also increases the success percentage. Similarly, the temporal separations between the intruders increase the number of options to capture the intruders; hence, it increases the chances of success. A robustness study has been performed to analyze the effects of uncertainty in the heading angle of intruders. The simulation results show that the performance of the DMRST-MTA approach is robust, and the effect of the uncertainty is negligible. Furthermore, the proposed DMRST-MTA is compared with the AP method [11]. The results show that the DMRST-MTA has better success for protecting the territory.

The main contributions of this article can be summarized as follows.

1) Formalization of the PDP into a multirobot spatiotemporal multitask assignment (MRST-MTA) problem. The presence of both the spatial and temporal dimensions and dynamic environments makes the solution for DMRST-MTA challenging.
2) A composite loss function is defined to handle the spatiotemporal nature of the task. The formulated ST-MTA problem is solved using a modified consensus-based distributed multitask assignment algorithm.
3) The PDP has been extended to limiting cases where the number of defenders is lower than that of intruders.

The rest of this article is organized as follows. Section II provides a review of the territory protection problem and task assignment methods. Section III presents the mathematical formulation of the PDP as an ST-MTA problem. Section IV presents the details of the DMRST-MTA approach. Performance evaluation is presented in Section VI. Finally, Section VII concludes this article.

II. Related Works

The PDP is a part of the general territory protection problem. This section describes briefly existing works in solving the territory protection problem in the first part. Since the PDP is converted into an ST-MTA problem, a brief review of multitask assignment literature is also presented in the second part.

A. Territory Protection Problems

In general, territory protection problems include those problems where a team of defenders attempts to capture the intruders (where intruders are either approaching toward the territory or trying to escape a boundary of territory). The literature on the territory protection problems can be broadly classified into the following categories: a) pursuit–evasion problems; b) guarding problems; c) reach-and-avoid games; d) border defense problems; and e) PDPs. Some of these problems in territory protection (particularly in categories a–d) are formulated as multiplayer games, which can be converted further to decision-making problems in cooperative control of a multiplayer system. A short description of these problems along with the relevant works in these areas is given in the following.

1) Pursuit–Evasion Problem: One of the fundamental problems in multiplayer games is the problem of a pursuit–evasion game, where a pursuer tries to capture an evader and the evader tries to escape. In [14] and [15], a geometric approach (using Voronoi diagrams) for the cooperative solution of a team of pursuers to capture a single evader was presented. It was extended to handle multiple evaders using a distributed algorithm in [16]. The fundamentals of pursuit–evasion games and their challenges were described in the seminal work [17].

2) Guarding Problem: In this problem, a group of defenders attempts to protect the target from an intruder by operating outside the territory. A differential game between an attacker and a guard was presented in [18] and [19]. The approach provides an analytical solution based on the initial position of the attacker. In [20], a geometrical capture region was analytically computed based on the location of the guards and the territory. Using this approach, the guard can capture the attacker in the capture region irrespective of its strategy.

3) Reach–Avoid Game: Here, the defenders (pursuers) attempt to capture an intruder outside the territory before the intruder enters the territory. A multiplayer reach–avoid game is a numerically intractable problem, and Chen *et al.* [21] provided upper bounds for the number of attackers required to reach a target. By extending Hamilton–Jacobi methods, the reach–avoid problems with time-varying dynamics, targets, and constraints were addressed in [22]. Two defenders and one intruder game have been solved using barrier functions based on the shape of the territory, and the defenders' initial positions in [23] and [24]. For a multiplayer game, the problem has been solved in two parts: in the first part, the feasible reachable sets are computed using barrier functions, and in the second part, the defenders are assigned to the feasible intruders such that the score is maximized. In [25], the subspace guarding problem with two fast defenders and one attacker was analytically solved using barrier functions.

4) Border Defense Problem: This is the inverse problem of perimeter defense, where intruders are inside the territory and defenders need to capture intruders before they escape the territory. In [26], a solution for radially outward moving evaders was proposed using a vehicle routing algorithm. In [27] and [28], the border defense problem was solved using game theory based on an Apollonius circle. The intruder's aspect of this problem, where intruders tries to escape the territory, is referred to as confinement escape problem [29].

In summary, all the aforementioned works address different aspects of a territory protection problem. The main objective of this article is to explicitly address the PDP with partial observability in a decentralized fashion using a smaller number of defenders with a multitask assignment approach. Before getting into the details of the proposed approach, a brief review of works on task assignment methods is highlighted next.

B. Task Assignment Methods

One of the crucial challenges in using multirobot systems for real-world applications is to solve a complex task allocation/assignment problem in an unknown/uncertain environment. The objective here is to find an optimal strategy that will assign a set of tasks to the robots such that the multirobot system completes its mission successfully. The taxonomy of a task allocation problem and a detailed review of various task allocation schemes are given in [30] and [31]. Most of these works address the dynamic assignment of spatially located tasks using any one of the following strategies: market-driven strategies [32], game-theoretic strategies [33], Hungarian method [34], [35], and consensus-based task assignment methods [13], [36], [37]. Recently, the taxonomy of the multitask allocation problem with temporal constraints for a spatial task was presented in [38]. A heuristic approach to solve the spatial multitask allocation with temporal constraints for multiple robots has been proposed in [39]. Here, the temporal constraints are handled as soft constraints in task allocation. Even though these approaches address the multitask assignment with temporal constraints, they are not suitable for a PDP due to their soft-temporal constraints.

In this article, the decentralized PDP is converted into an ST-MTA problem, where tasks are dynamic in nature and need to be executed at specific times. In addition, to solve the above problem, the 1-D multiple-task allocation using a CBBA [13] has been extended to handle the spatiotemporal multiple-task assignment problem, and the same approach is described next.

III. ST-MTAs for the PDP

This section presents the setup for a PDP with distributed sensing and then shows how this can be converted into an ST-MTA problem.

A. Problem Definition

The PDP aims to activate a team of defenders with proper task assignments such that they protect a territory from invading intruders, by capturing the intruders before they enter the territory. Next, we describe a mathematical formulation of the PDP.

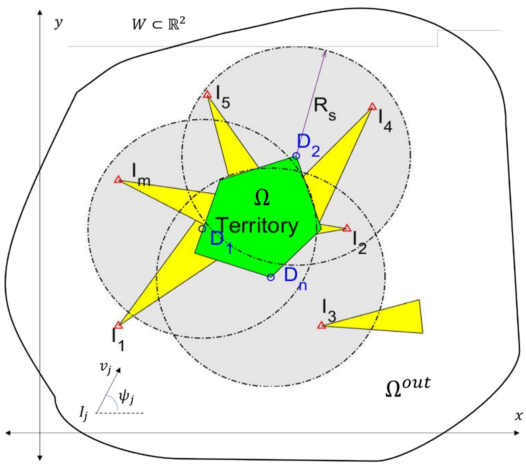


Fig. 2. PDP with the distributed sensing system.

Fig. 2 shows a typical scenario for a PDP with distributed sensing, where each defender can sense an intruder within its sensing range (R_s). The figure highlights the individual defender sensor coverage and also highlights the potential impact regions of the intruders. Consider a region $W \subset R^2$, a region of interest that includes the critical infrastructure and its neighborhood. The territory is approximated using a convex shape and is shown by the green color in Fig. 2 and is denoted as $\Omega, (\Omega \subset W \subset R^2)$. Let $\partial\Omega$ denote the territory's perimeter, and let $\Omega^{\text{out}} = (W - \Omega)$ denote the region outside the territory. The length of the perimeter is L, and any point on the perimeter is denoted by $\zeta \in [0, L)$.

Let us consider a set of defenders $\mathcal{D} = \{D_1, D_2, \ldots, D_N\}$ operating inside the territory. Defenders can communicate synchronously among themselves with limited bandwidth. Each defender is equipped with the same type of sensor to detect the intruder's position and velocity within their sensing radius. Let $D_i^s = (x_D^i, y_D^i)$ be the position of the ith defender and velocity V_D^i and heading angle ψ_D^i. Defenders are initialized inside the territory, i.e., $D^s(t_0) \in \Omega$. The defenders are assumed to operate only on and inside the boundary of the convex territory. The kinematic equations of the ith defender are given as

$$\dot{x}_D^i = V_D^i \cos(\psi_D^i), \qquad \dot{y}_D^i = V_D^i \sin(\psi_D^i) \tag{1}$$

$$\psi_D^i \in [0, 2\pi), \quad V_D^i \in [0, V_D^{\max}], \quad i = 1, 2, \ldots, N. \tag{2}$$

The number of intruders at any given t is a varying number. Let M_t be the number of intruders that appeared in Ω^{out} at time instant t. Let us consider a set of intruders $\mathcal{I} = \{I_1, I_2, \ldots, I_{M_t}\}$. Here, we assume that there is no coordination between the intruders and that they will not penetrate the territory simultaneously. Let $I_j^s = (x_I^j, y_I^j)$ be the position of the jth intruder and velocity V_I^j and heading angle ψ_I^j. Intruders are initialized outside the territory $I^s(t_0) \in \Omega^{\text{out}}$. The intruder's kinematic equations are given as

$$\dot{x}_I^j = V_I \cos(\psi_I^j), \qquad \dot{y}_I^j = V_I \sin(\psi_I^j) \tag{3}$$

$$\psi_I^j \in [0, 2\pi), \quad V_I \in [0, V_I^{\max}], \quad j = 1, 2, \ldots, M_t. \tag{4}$$

It is assumed that the intruders are nonagile, and the angular velocity of an intruder is bounded within $\pm\delta$ rad/s. Note that the number of intruders may be more than the number of defenders $(M_t > N)$. In addition, the defender's maximum velocity is more than that of the intruder $V_D^{\max} > V_I^{\max}$.

When the distance between the defender and the intruder is smaller than ϵ, the defender captures the intruder using a safety net mechanism. The capture condition is given as

$$C = \left\{ I_j \mid \exists D_i \ \text{ s.t. } \|I_j^s - D_i^s\|_2 \le \epsilon \ \& \ I_j^s \in \Omega^{\text{out}} \right\}. \tag{5}$$

In [2]–[5], it was assumed that a defender captures/neutralizes an intruder using a head-on collision mechanism. This capture process will destroy both the defender and the intruder. However, in this article, we propose that the defender employs a safety net to capture an intruder. Because of this, it is possible that a defender can capture multiple intruders. Hence, this allows us to use a smaller number of defenders to protect the territory from a large number of intruders.

From Fig. 2, we can see that D_1 can detect only I_m, whereas D_2 can detect I_2, I_4, and I_5. Since partial observability exists in detecting the targets, all the N defenders need to act cooperatively in a decentralized way to capture all the intruders. To put simply, the PDP's main objective is to detect the varying number of intruders with a distributed sensing mechanism and assign multiple tasks of capturing these intruders to different defenders in a decentralized fashion.

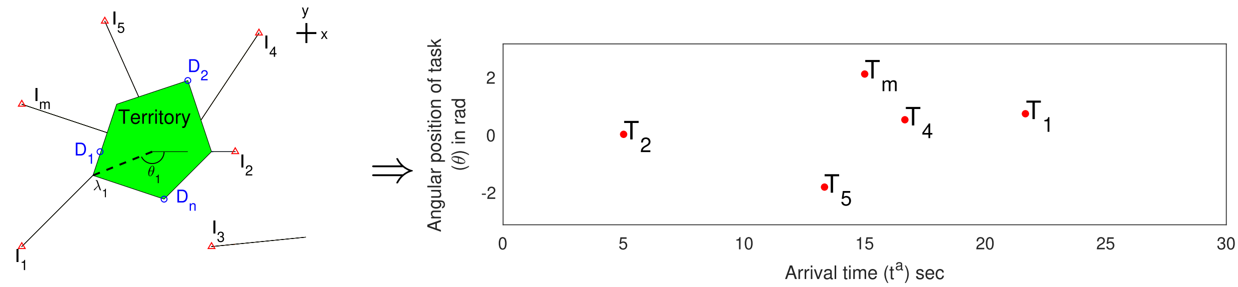
B. ST-MTA Problem

In this subsection, we show the conversion of the above PDP into an ST-MTA problem. Based on the sensor information, a defender will be able to identify potential intruders. First, the defender computes the intruder's velocity vector. Based on the velocity vector, the defender assumes that the intruder travels with the same velocity vector and determines the point at which the intruder intersects the perimeter. This intersecting point for the jth intruder is called as the point of arrival (λ_j). The time required for the jth intruder to reach the point of arrival is called as the arrival time (t_j^a) and is computed as

$$t_j^a = \frac{\|I_j^s - \lambda_j\|_2}{\|V_I^j\|_2}. \tag{6}$$

The defender has to neutralize the jth intruder. A neutralization task is defined for the defenders such that the defender should visit the arrival location (λ_j) at the arrival time (t_j^a). Thus, the PDP problem of neutralizing intruder is converted into a spatiotemporal task and is denoted by $T_j(\lambda_j, t_j^a)$.

Since the defenders know only the current positions and velocities of intruders, they will determine unique arrival points in the perimeter and the time of their arrival, as shown in Fig. 3(a). The corresponding spatiotemporal tasks are shown in Fig. 3(b). The point of arrival for the intruder j is denoted by λ_j. Now, this Cartesian point $\lambda_j(x, y)$ is converted to a polar coordinate form (r_j, θ_j) with the center at the centroid of the territory. The angular position θ_j uniquely denotes λ_j on the convex territory. For easy understanding, angular position (θ) is used to denote the task location in Fig. 3(b).



(a) (b)

Fig. 3. Representation of tasks in spatiotemporal space. (a) Estimated location of arrival of intruders for a given time t. (b) Spatiotemporal tasks for a given time instant t.

From Fig. 3(b), one can note that the task is available at a specific location (θ) at the time t^a. This means that the task has a strict requirement that it is available only at a specific location and at a specific time. If the defender reaches the location λ earlier, then it must wait for the task. If the defender reaches the location λ after the arrival time, the intruder will enter the territory, and this indicates that defense has failed. The intruder I_3 is not directed toward the territory, as shown in Fig. 3(a), and hence, the task T_3 is not defined in Fig. 3(b).

The intruder may change its heading angle during its movement, and then, the spatiotemporal task also will change. However, smooth heading variations by an intruder lead to a smooth variation in the spatiotemporal task. The philosophy used in our approach is that the defenders predict the spatiotemporal task $T_j(\lambda_j, t^a_j)$ and use a decentralized multitask assignment algorithm for assigning the specific spatiotemporal tasks to defenders. Based on its assigned task, a defender moves toward its first assigned task location. These steps are repeated at every time instant to handle the smooth variations in the intruder heading.

The set of spatiotemporal tasks is denoted as $\mathcal{T} = [T_1(\lambda_1, t^a_1), \ldots, T_j(\lambda_j, t^a_j), \ldots, T_{M_t}(\lambda_{M_t}, t^a_{M_t})]$. Note that $M_t > N$; defenders will be assigned to multiple tasks at any given time t. Let $\mu_i = \{T_a, T_b, T_c\}$ be the multiple tasks assigned to the ith defender. Based on the assigned task's arrival time and location, the defender will generate its motion commands to follow the trajectory (μ_i). The trajectory μ_i means that D_i executes tasks T_a, T_b, and T_c in a sequence.

IV. Decentralized Approach for MRST-MTA

In the previous section, the PDP was formulated as an MRST-MTA problem. In the section, we present a decentralized approach for the ST-MTA problem.

Using the sensor information, the defender robots can identify potential intruders who will attempt to enter the territory at location λ_j and at time t^a_j. Owing to the partial observability conditions, all the intruders may not be visible to a defender. The task of neutralizing the intruder is first converted into a spatiotemporal task $T_j(\lambda_j, t^a_j)$. For the sake of brevity, in the rest of this article, $T_j(\lambda_j, t^a_j)$ is denoted by T_j.

In order to protect a convex territory, the defenders will be assigned multiple spatiotemporal tasks in a decentralized fashion. This requires the defenders to navigate in a sequence to specific locations (λ) at specific times (t^a) to neutralize the intruders. The objective of a DMRST-MTA is to find an optimal sequence of defenders' motions such that the defenders collectively capture the intruders. Here, the optimal sequence implies the sequence that minimizes the total distance traveled by a team of defenders, and this sequence of defenders' motion is a trajectory to be followed by a defender to neutralize the intruders.

Let us consider that the defender D_i can detect only a few intruders at time t, denoted by the set $\mathcal{O}_i$. Let the total number of detected tasks in the set $\mathcal{O}_i$ be M^i_t. Note that the union of all such sets is equal to the total number of the tasks at any given time t, i.e., $\bigcup_{i\in\mathcal{I}} \mathcal{O}_i = \mathcal{T}$. The sequence in which a defender D_i executes the spatiotemporal tasks is μ_i, where $\mu_i \in \{\mathcal{O}_i \bigcup \varnothing\}^{M^i_t}$. Owing to the partial observability condition and overlapping search regions, some intruders are observed by many defenders. Hence, the defenders need to form a consensus among themselves to determine which intruders are assigned to which defender.

The multiple spatiotemporal tasks are assigned to defenders such that the overall cost to the team of defenders is minimum. In the next subsection, the costs for the spatiotemporal tasks consisting of both the spatial and temporal loss functions are described.

A. General Cost Function Formulation

Let the available tasks for the ith defender be $\mathcal{O}_i$. The defender D_i finds a locally optimal sequence (μ_i) to neutralize its tasks in $\mathcal{O}_i$. The current position of the defender D_i is D^s_i. At the beginning, the optimal sequence μ_i is set as $\varnothing$. To find the optimal sequence based on the available task $\mathcal{O}_i$, it is necessary to define a general cost function that combines both the spatial component and the temporal component of the task T_j, and the same is given as follows.

1) Spatial Loss Function $L^s_i(T_j)$: The spatial loss depends on the defender's position and the intruder's position. The spatial loss is the sum of the distances traveled by the defender to reach the arrival location of an intruder and a fraction of the distance traveled by the intruder from its current position to its arrival location. The distance traveled by a defender for neutralizing the task T_j depends on the distance that needs to be traveled by the defender D_i from the location $D^s_i(\mu_i, j)$ (location of the task previous to T_j on trajectory μ_i) to the neutralizing point λ_j. The spatial loss function is defined as

$$L_i^s(T_j) = \|\lambda_j - D_i^s(\mu_i, j)\|_2 + \eta\|(\lambda_j - I_j^s)\|_2 \quad (7)$$

where $\eta \in (0,1)$ is the scaling factor; the second component reflects the intruder's distance to reach the perimeter. For a trivial case, if the defender is already at the location λ_j, the first term becomes zero; then, the second term provides the possibility for the defender to neutralize another task and does not force it to wait for the task T_j. The defender D_i should be assigned to another intruder I_k if the intruder I_j is far away from the territory. If the intruder I_j is far away, the cost will be high, and the defender will be assigned to some other low-cost tasks. If the intruder I_j is closer to the perimeter, then the second term will be small, and the defender will be assigned to the task T_j.

For example, a defender D_i plans its trajectory $\mu_1 = \{T_1, T_2\}$; then, the spatial loss is computed as

$$L_i^s(T_1) = \|\lambda_1 - D_i^s\|_2 + \eta\|(\lambda_1 - I_1^s)\|_2 \quad (8)$$

$$L_i^s(T_2) = \|\lambda_2 - \lambda_1\|_2 + \eta\|(\lambda_2 - I_2^s)\|_2. \quad (9)$$

2) Temporal Loss Function $L_i^t(T_j)$: Before computing the temporal cost, the defender checks the feasibility of reaching the location λ_j within the arrival time of the task (intruder) (t_j^a). The time spent by the defender D_i for the previous task in a sequence μ_i is $t_i^a(\mu_i, j)$. The task is said to be feasible if and only if the following conditions are satisfied:

$$0 \leq \frac{\|\lambda_j - D_i^s(\mu_i, j)\|_2}{V_D^{\max}} < (t_j^a - t_i^a(\mu_i, j)) \quad (10)$$

T_j is feasible if (10) is satisfied. The time available is a very important aspect for execution of a task. Let us consider a task T_a at time 2 s, task T_b at time 3 s, and task T_c at time 4 s, and all three tasks are time feasible. Now, the time available for tasks T_b and T_c (in sequence $\{T_a, T_b, T_c\}$) is the same, but they should be executed in a time order; T_b should be executed prior to T_c. From this example, it is clear that few tasks can have same time availability, but they may arrive at different times. In such scenarios, the priority should be given to the tasks with lesser arrival times.

The temporal loss function should reflect the exact time of arrival and the time available for a defender to execute a task after executing the previous task in the trajectory. Hence, the temporal cost function is selected as a product of the time available for the defender and the arrival time of the intruder. Mathematically, the temporal loss function $L_i^t(T_j)$ defined for defender D_i to execute task T_j at time t_j^a is given as

$$L_i^t(T_j) = \begin{cases} t_j^a(t_j^a - t_i^a(\mu_i, j)), & \text{if task j is feasible} \\ \infty, & \text{if task j is infeasible} \end{cases} \quad (11)$$

where $t_i^a(\mu_i, j) = 0$ for the first task on a trajectory; as T_j is the first task in trajectory μ_i, defender D_i directly starts executing assigned to task T_j.

For example, a defender D_1 plans trajectory $\mu_1 = \{T_1, T_2\}$; then, the temporal loss is computed for feasible trajectory

$$L_1^t(T_1) = t_1^a(t_1^a - 0)$$

$$L_1^t(T_2) = t_2^a(t_2^a - t_1^a).$$

3) Composite Loss Function $L_i(T_j)$: A general composite loss function for a spatiotemporal task can be defined as

$$L_i(T_j) = f(L_i^s(T_j), L_i^t(T_j)). \quad (12)$$

The characteristic required by a composite loss function is to have a small value when either of the spatial or temporal loss is small. If the spatial loss value is large and the temporal loss value is small, then a composite loss should have a small value to reflect the urgency. If the temporal loss value is large and the spatial loss value is small, then the composite loss should be small so that priority will be given to the task that requires lesser distance to be traveled. The multiplication function in the composite loss function satisfies the above requirements. For the PDP problem, we have selected the product form for the composite loss function

$$L_i(T_j) = L_i^s(T_j) L_i^t(T_j). \quad (13)$$

It may be noted that the spatial and temporal loss functions are defined for only task T_j and its previous task and not for the entire trajectory. In addition, the loss function does not involve all the tasks carried out by the defender D_i. First, we compute the total loss function for the trajectory μ_i as the summation of all the composite losses of tasks in the trajectory

$$L_i(\mu_i) = \sum_{j \in \mu_i} L_i(T_j). \quad (14)$$

It should be noted that this total loss function for the trajectory does not consider the sequence in which these tasks are done; it just accumulates the loss values for each of the tasks present in the trajectory. Hence, a general cost function is defined for a task T_j in the trajectory μ_i. The general cost function $c_{ij}(\mu_i)$ is based on the effective position of the task T_j in the trajectory μ_i and is given by

$$c_{ij}(\mu_i) = \begin{cases} \min_{n \leq |\mu_i|} L_i^{\mu_i \oplus_n \{j\}} - L_i^{\mu_i}, & \text{if } j \in \mu_i \\ \infty, & \text{if } j \notin \mu_i \end{cases}. \quad (15)$$

$|\cdot|$ is the cardinality of the set, and $\mu_i \oplus_n \{j\}$ denotes that task T_j is added after the nth element trajectory μ_i. As the task T_j is added at any location, the new task's cost added to the trajectory is the difference between the total loss function of the new trajectory and the original trajectory. If task T_j is not in the trajectory, then the cost value is ∞.

B. Decentralized ST-MTA Problem

A spatiotemporal task requires that the defender should be at the arrival location of an intruder at its time of arrival. The team of defenders has to execute all the tasks, and the objective is to minimize the total cost required for executing the tasks. Each defender D_i plans its trajectory (μ_i) and executes its assigned tasks in sequence. Thus, (16) defines an integer programming problem for the trajectory computation. $C(\mu_i)$ denotes the cost of the trajectory of defender D_i, and a decision variable δ_{ij} denotes whether task T_j is in the trajectory of defender D_i. In this way, an optimization problem that minimizes cost is formulated to obtain the trajectory of the defender.

Algorithm 1: Local Trajectory Generation by Using the Composite Cost Function (CBBA for Defender i at Iteration q).

```
1:  input O^i, b_i(q-1), μ_i(q-1), y_i(q-1), z_i(q-1)
2:  Initialize b_i(q) = b_i(q-1);   μ_i(q) = μ_i(q-1);
    y_i(q) = y_i(q-1);   z_i(q) = z_i(q-1);
3:  conflict resolved = 0;
4:  while conflict resolved = 0 do   % Auction Algorithm;
5:    c_ij = min_{n≤|μ_i|} L_i^{μ_i ⊕_n {j}} - L_i^{μ_i},  ∀j ∈ O^i \ b_i(q);
6:    h_ij = 𝕀(c_ij < y_ij(q)),   ∀j ∈ O^i;
7:    J_i = argmin_j c_ij.h_ij;
8:    n_{i,J_i} = argmin_j L_i^{μ_i ⊕_n {j}};
9:    b_i(q) = b_i(q) ⊕_end J_i;
10:   μ_i(q) = μ_i(q) ⊕_{n_{i,J_i}} J_i;
11:   y_{i,J_i}(q) = c_{i,J_i};
12:   z_{i,J_i}(q) = i;
13:   Call Algorithm 2   % consensus algorithm;
14: end while
```

(Remark: minimization over all ∞ value is taken as ∞. All ∞ means that the task T_j is infeasible along path μ_i)

The spatiotemporal multiple-task assignment problem aims to assign each of the M_t tasks to the available N defenders such that a task is assigned to only one defender. A defender can execute only the detected tasks. A defender D_i has detected few tasks $\mathcal{O}_i$; then, it can execute the tasks present in $\mathcal{O}_i$. The cost of assigning a task T_j to the defender D_i is c_{ij}. The tasks are assigned such the total cost is minimized.

The task assignment problem is defined as

$$\min_{\delta_{ij}} \quad \sum_{i=1}^{N}\sum_{j=1}^{M_t} c_{ij}(\mu_i)\delta_{ij} \tag{16}$$

$$\text{such that} \quad \sum_{i=1}^{N} \delta_{ij} \leq 1 \qquad \forall j \in \mathcal{T} \tag{16a}$$

$$\delta_{ij} \in \{0,1\} \qquad \forall (i,j) \in \mathcal{D} \times \mathcal{T} \tag{16b}$$

$$\delta_{ij} = 0 \qquad \text{if} \quad j \notin \mathcal{O}_i. \tag{16c}$$

The constraint in (16a) enforces that the task is assigned only to one defender. Equation (16b) is a condition on the decision variable δ_{ij}, which states whether the defender D_i is assigned to the task T_j. A defender is allowed to bid only observable tasks. Hence, the constraint given in (16c) is added to enforce the assignments only to observable tasks, which, in turn, reduces the complexity. One can also relax this constraint and obtain a better solution at the cost of complexity.

This problem is complex as the cost is a function of the trajectory. This NP-hard problem is solved in two steps. In the first step, each defender D_i computes its own locally optimal trajectory μ_i. In the second step, using the locally optimal trajectory of each defender, a task is assigned to the defender who has the lowest cost.

B. Consensus-Based DMRST-MTA Approach

The DMRST-MTA approach is carried out in two steps. In the first step, the local cost-optimal sequence of tasks for every defender is computed. Each defender greedily computes its trajectory while minimizing the cost of locally (only for that individual) computed trajectory. Now, each defender had computed its own trajectory without the consideration of other defenders and their trajectories. Then, they share this information among themselves with the provided communication setup. The computed trajectories might have conflicts; hence, a conflict resolution is required. The defender with the lowest cost for the task is assigned to the task. The defenders modify their local cost-optimal trajectories based on the assigned tasks over consensus. The defenders will repeat their local trajectory computation and global trajectory generation steps until global trajectories become the same as the locally computed trajectories. Both these computations are given next, and they are also summarized as Algorithms 1 and 2

1) Algorithm 1: Computing the Local Optimal Trajectory: Let $\mathbf{y}_i \in \mathbb{R}_+^{M_t^i}$ be the vector of winning (smallest) bid values for all the spatiotemporal tasks available at time t. For defender D_i and task T_j, $\mathbf{y}_i$ is initialized as $y_{ij} = \infty$. The trajectory and bundle are initialized as null, i.e., $\mu_i = \mathbf{b}_i = \emptyset$. The vector indicating winning defender is initialized to zero, i.e., $\mathbf{z}_i = \mathbf{0}$. Without loss of generality, the winning bids ($\mathbf{y}_i$), winning defenders list ($\mathbf{z}_i$), trajectory (μ_i), and bundle ($\mathbf{b}_i$) of the $(q-1)$th iteration are used as initial condition at the qth iteration.

As a first step, the defender computes the costs for all the observed tasks from its initial position and bids for a task with minimum cost, as shown in line 5 of Algorithm 1. Then, the defender decides whether its own cost is less than the minimum bid for that task. $\mathbb{I}(\cdot)$ is an indicator function, which is unity for a true argument and zero otherwise. If the cost is smaller than the known bid value, the defender bids for the task. This is executed using line 7 of Algorithm 1. Then, the defender adds this task to trajectory, and thus, the initial trajectory is fixed. Each defender generates its own local cost-optimal trajectory to execute the tasks. Then, defenders freeze those local trajectories and communicate their trajectories with their associated costs. A global trajectory is computed using a consensus among all the defenders using Algorithm 2. After having a consensus, each defender computes its own trajectory starting from its new initial position (the position of the last task in the trajectory) and computes the cost for the observed tasks that are not in trajectory.

2) Algorithm 2: Global Trajectory Selection Algorithm: Algorithm 2 solves the MTA problem for a given fixed trajectory μ_i. As trajectories for each defender are fixed, the cost matrix (C) is constant. The given MTA problem reduces the fixed cost matrix to find a minimum cost for each task. Multiple defenders may bid for a single task, and this creates a conflict between the trajectories. The conflict between multiple defenders for a single task is resolved by finding a minimum bid for that task. After conflict resolution, the winning bids, the winning defender, and the trajectories for the defenders involved in the conflict will be

Algorithm 2 Computation of a Feasible Global Trajectories (Consensus by Defender i at Iteration q).

1: **input** $\mathbf{b}^k$, μ^k, $\mathbf{y}^k$, $\mathbf{z}^k$ (data received from defender k via synchronized communication);
2: **if** $z^k_{kj} = \emptyset$ **then**
3: $\quad y^k_{kj} = \infty$
4: **end if**
5: **if** $z^k_{kj} = k$ **then**
6: $\quad$ **if** $z^i_{ij} = k$ **then**
7: $\quad\quad$ Update
8: $\quad$ **else**
9: $\quad\quad$ **if** $y_{kj} < y_{ij}$ **then**
10: $\quad\quad\quad$ Update
11: $\quad\quad$ **end if**
12: $\quad$ **end if**
13: **end if**
14: **if** $z^k_{kj} = i$ **then**
15: $\quad$ **if** $z^i_{ij} = k$ **then**
16: $\quad\quad$ Reset
17: $\quad$ **end if**
18: **end if**
19: **if** $z^k_{pj} = p;\ p \neq \{i, k\}$ **then**
20: $\quad$ **if** $z^i_{ij} \neq p$ **then**
21: $\quad\quad$ **if** $y_{pj} (= y_{kj}) < y_{ij}$ **then**
22: $\quad\quad\quad$ Reset
23: $\quad\quad$ **end if**
24: $\quad$ **end if**
25: **end if**
26: **if** $\mathbf{z}_i = \mathbf{z}_k \quad \forall i, k \in \mathcal{I}$
27: $\quad$ conflict resolved $= 1$
28: **end if**
29: Update : $y_{ij} = y_{kj}, z_{ij} = z_{kj}$;
30: Reset : $y_{ij} = \infty, z_{ij} = \emptyset$;

modified. The winning defender will update the vectors for the winning task, as shown in line 29 of Algorithm 2. In addition, some of the defenders may have to remove the same task from their trajectories. As one task becomes invalid, the rest of the trajectory after that task becomes invalid, and hence, defenders must discard the trajectory from the first unassigned task.

Algorithm 2 gives the detailed steps involved in the global trajectory generation process. After computing the winning trajectory, winning bids, and winning defenders' list, each defender regenerates a locally optimal trajectory. Then, the defenders share their bid values, and the global trajectory is then computed. This procedure is repeated until all local trajectories are unchanged in the global computation. In a nutshell, the local trajectories are modified until they become globally valid.

C. Convergence of the Algorithm

The loss function for a trajectory $L_i(\mu_i)$ is defined in (14). The composite loss function for a trajectory $(L_i(\mu_i))$ is a monotone function; the function value increases with a newly added task. Here, a newly added task in the trajectory μ_i increases the loss function exactly by the loss value of the individual task. Now, from the definition of the cost function given in (15) for task T_j in the trajectory μ_i, the monotone condition can be written as

$$c_{ij}(\mu_i) \leq c_{ij}(\mu_i \oplus_n j).$$

The characteristic of the above condition is similar to the condition presented by [13, eq. (7)]. Note that DMRST-MTA minimizes the cost, whereas Choi *et al.* [13] maximize the cost. Since the spatiotemporal cost is a monotone function, the convergence proof given in [13] will hold for DMRST-MTA.

VI. Performance Evaluation of DMRST-MTA

This section presents the results of a detailed performance evaluation of the DMRST-MTA approach using simulations. First, we present the simulation results for a typical scenario that clearly illustrates the working of the proposed approach for an increasing number of intruders. Next detailed Monte Carlo study results are presented to evaluate the performance of the proposed approach under different scenarios. Finally, the performance of DMRST-MTA is compared with the decentralized AP approach.

The PDP considered here is the protection of a synthetic convex territory region shown in Fig. 2. The vertices of the considered territory are (20,0), (0,−20), (−30,−10), (−20,20), and (10,30). For the simulation study, the number of defenders (N) is set to 3, and the number of intruders (M_t) at time t is limited by 6. Any defender can neutralize an intruder within a neutralizing distance (r), and this is set at 5 m. The velocity of the intruders (V_I) is selected as 3 m/s. The maximum velocity of all the defenders ($V_D^{\max}$) is constrained at 4.5m/s.

A. Simulation Study

At any given time t, a total of six spatiotemporal tasks are available. The results for a single initial condition are presented here. The defender's assignments of the intruders at different times are shown in Fig. 4.

Fig. 4(a) shows the initial condition. The defender's sensing region is shown in a shaded region, and a defender can be assigned to the observed intruder. Fig. 4(a) shows that three defenders can sense only four intruders. As only four intruders are sensed, three defenders create their trajectories to capture the observed intruders. Here, we can see that D_3 is unassigned.

Fig. 4(b) shows the scenario at 4 s. Defenders can sense all six intruders. The defenders are assigned to the intruders in a sequence; D_1 will neutralize I_1 first, and then, it will go for I_3. First, the assigned intruder is shown by an $*$ mark, the second by a diamond, and the third by a square. The defender D_1 senses I_1 and I_3; D_2 senses I_4, I_5, and I_6; and D_3 senses I_1, I_2, I_3, and I_4. Fig. 5 shows the sensed intruder by each defender, and the assigned intruders are shown by the dark color. Defender D_2 can sense intruders I_4, I_5, and I_6. From these three sensed tasks, two tasks $\{T_5, T_6\}$ are assigned to D_2. Defender D_2 executes tasks by following trajectory $\mu_2 = \{T_5, T_6\}$. The sequence of tasks can also be observed in the time axis of the assigned tasks, where T_5 comes before T_6. The spatiotemporal figure shows the time dimensions of the task, and one can

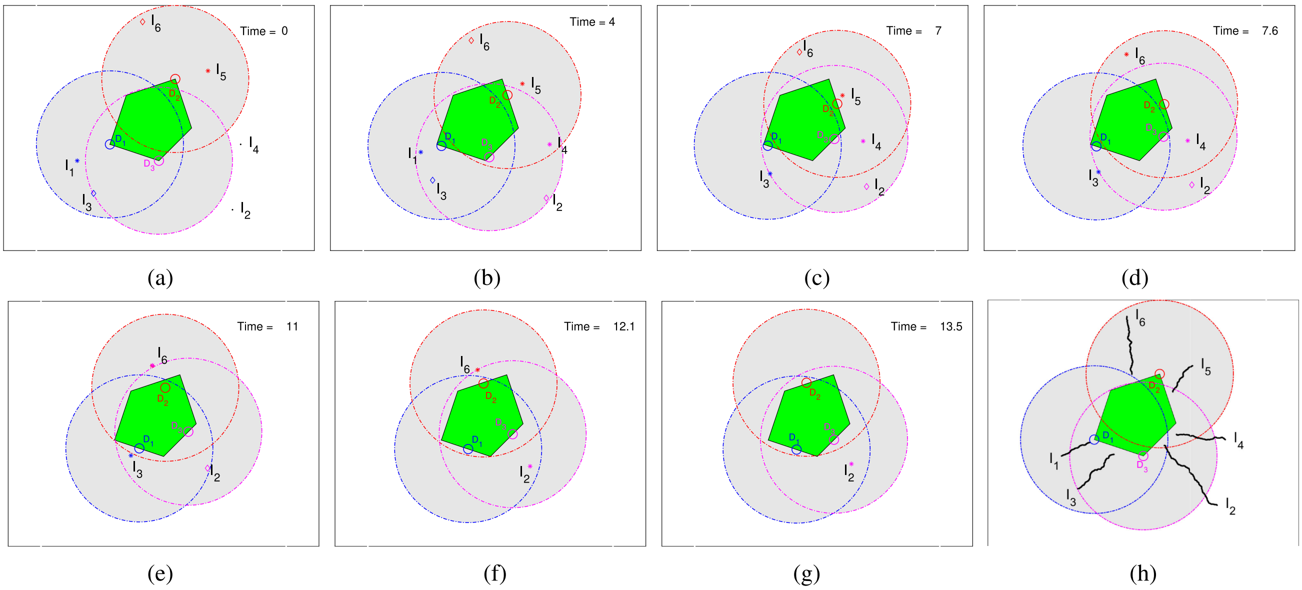


Fig. 4. Snapshots of the intruder tasks assigned to the defenders, at different time instants. Defenders D_1, D_2, and D_3 are represented by blue, red, and magenta colors, respectively. The intruders assigned to an defender are colored with the color of defender. The path selected by defender is shown by symbols in sequence : $*$, $\diamond$, $\square$. (a) $t = 0$ s. (b) $t = 4$ s. (c) $t = 7$ s. (d) $t = 7.6$ s. (e) $t = 11$ s. (f) $t = 12.1$ s. (g) $t = 13.5$ s. (h) Intruders path.

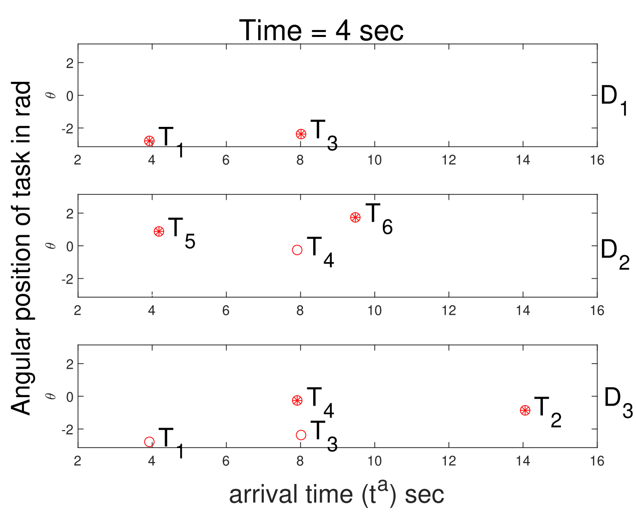


Fig. 5. Spatiotemporal task assignment in spatiotemporal dimension. The spatial location is denoted by angular position with respect to the center of the territory. $\circ$ shows the observed tasks, and assigned tasks are shown by $*$.

observe that the defenders will execute the assigned tasks as per increasing order of time.

Fig. 4(c) shows the scenario at 7 s, where D_1 has captured the intruder I_1, and there are only five remaining intruders. Even though the number of intruders is changed, the same algorithm gives the same previously planned assignment. D_1 is assigned to I_3, and the remaining assignments are unchanged. Fig. 4(d) shows the scenario at 7.6 s, where D_2 has captured I_5 and is now assigned only to I_6. Fig. 4(e) shows the scenario at 11 s, where D_3 has captured the intruder I_4. Now, there are three intruders and three defenders. The assignment is the same as planned. Fig. 4(f) and (g) shows the capturing of the remaining intruders. The intruders are randomly changing the heading angles; the path of each intruder is shown in Fig. 4(h).

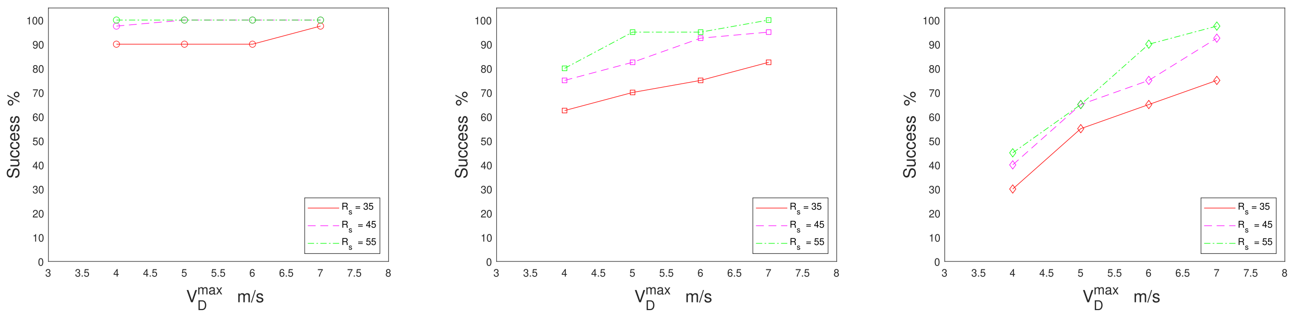
From the simulation results, one can observe that a smaller number of defenders can protect the territory against the larger and varying number of intruders. Defenders can also neutralize intruders who undergo small heading variations. The success of the defense depends on the speed ratio, interintruder separation distance, and the sensing range of the defenders. The effects of these parameters are studied next.

B. *Performance Evaluation Using Monte Carlo Simulations*

Monte Carlo simulations are performed by varying the speed limits of the defenders, sensing radii of the defenders, and interintruder time separations. The velocity of the intruder is kept constant, and the maximum velocity of the defender is varied. A total of 15 intruders are considered in a single run, and they come from random positions over a time interval. At a time instant, at most six intruders are considered. Here, 500 randomized simulations have been conducted for each of the Monte Carlo study. When all the intruders are captured in a simulation run, it is called success. The success percentage is computed as

$$\text{Success (in \%)} = \frac{\text{\# successful simulations}}{\text{\# total simulations}} \times 100.$$

Fig. 6 shows the result that as the defender's speed limit increases, the success rate to capture the intruders increases. The success rate is approaching 100% as some randomized simulations may have infeasible tasks due to insufficient temporal separations between consecutive tasks. Fig. 6(a)–(c) shows that for a given intruder velocity, as the defender's velocity increases, the success rate improves. The simulations have been



(a) (b) (c)

Fig. 6. Variations in defender speed limit $N = 3$, $M_t = 6$. (a) $V_I = 1$m/s. (b) $V_I = 2$m/s. (c) $V_I = 3$m/s.

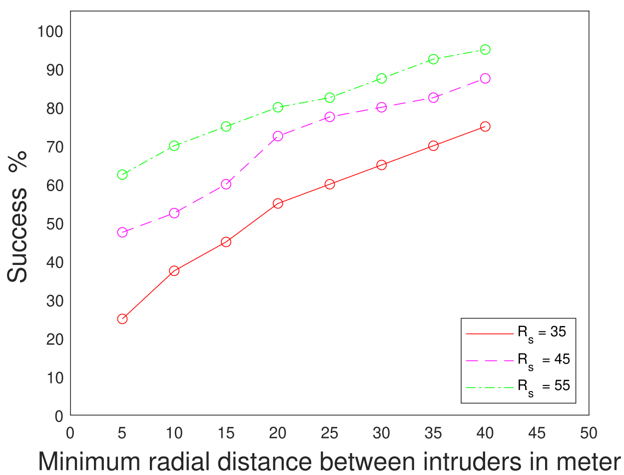


Fig. 7. Variation in interintruder separation distance $N = 3$, $M_t = 6$, $V_I = 3\,\text{m/s}, V_D^{max} = 6\,\text{m/s}$.

conducted with different intruder velocities. For the lower intruder velocity, the success rate is higher. For higher intruder velocity, the defender needs even higher velocity to capture multiple intruders. A higher sensing range helps the defender to detect the intruders earlier and plan accordingly. Fig. 6 shows that the success rate increases with increasing sensing range.

The next set of randomized simulations were carried out by varying the interintruder distance separations. We have used the same setup as used for the number of intruders, except that the newly added intruders have minimum separation in their arrival times. This interintruder distance provides feasibility for a smaller number of defenders to capture more intruders. Fig. 7 shows that when the interintruder distance separation increases, the success rate increases. If the interintruder separation is large, then the defenders can easily capture a large number of defenders. This helps for strategy making for an intruder; if the interintruder separation is small, then the intrusion rate increases. In addition, these simulations were carried out by also varying the sensing radius. As the sensor radius increases, the planning time for the defender increases. The defender can coordinate properly within the defender team, and hence, its success rate improves. Fig. 7 shows that, here also, the success rate increases with increasing sensing radius.

TABLE I
SUCCESS RATE OF THE DEFENSE WITH VARIATIONS IN VARIANCE AND SPEED OF DEFENDERS. $V_I = 2\,\text{m/s}$, $N = 3$, AND $M_t = 6$

σ in degrees $\rightarrow$ V_D in m/sec $\downarrow$	0	5	10	15
4	90.8%	90%	89.8%	89.2%
5	96 %	95.6%	95.4%	95%

C. Robustness Study

The proposed DMRST-MTA mainly uses the heading direction of the intruder for the computation of the spatiotemporal tasks. In order to understand the effect of measurement uncertainty of the intruder heading angles on the PDP, we have performed simulations by adding noise to the measurements in the heading angles of the intruders.

A zero-mean Gaussian noise is added to the heading direction of the intruders. The variance of the noise is proportional to the distance of the intruder from the boundary. When intruders are far away from the boundary, the measurements will not be accurate, resulting in higher variances. As the intruder comes closer to the boundary, sensors become more accurate, and the variance decreases. We have considered that the variance varies linearly with respect to the distance; the value of σ is highest at a distance of 55 m (sensing radius of the defender) and zero at a distance of 5 m. The parameters considered for the simulations are $R_s = 55\,\text{m}$ and $V_I = 2\,\text{m/s}$. Table I shows the success rate of the perimeter defense with uncertainties in the heading angle measurements.

Next, a Brownian noise is added to the measured heading direction of the intruders. The simulations with the Brownian noise scenario results in a success rate of 90.5% for $V_D = 4\,\text{m/s}$ and 94.6% for $V_D = 5\,\text{m/s}$.

From the results, one can clearly see that the effect of the Gaussian noise in the measurement of the heading angle of the intruder and the effect of the Brownian noise in the measurements on the success rates are negligible.

D. Performance Comparison

The decentralized AP method was proposed in [11] for protecting a circular territory against radially incoming intruders.

TABLE II
AVERAGE PERCENTAGE AND STANDARD DEVIATION OF CAPTURED INTRUDERS OVER 100 RUNS FOR NAIVE, AP, AND DMRST-MTA APPROACHES

	Uniform	von Mises	Bimodal	Time-varying
Naive	64.9 ± 3.2	58.8 ± 3.6	64.8 ± 3.2	58.2 ± 3.1
AP	64.9 ± 3.2	72.4 ± 3.1	67.8 ± 2.9	65.6 ± 4.1
DMRST-MTA (Ω)	$\mathbf{65.6 \pm 4.5}$	$\mathbf{77.7 \pm 4.5}$	$\mathbf{68.8 \pm 4.8}$	$\mathbf{71.2 \pm 4.8}$
DMRST-MTA ($\partial\Omega$)	$\mathbf{64.3 \pm 4.1}$	$\mathbf{77.7 \pm 3.6}$	$\mathbf{69.8 \pm 4.0}$	$\mathbf{70.0 \pm 5.2}$

Defenders compute adaptive partitions by estimating the distributions of the intruders, and then, defenders follow a path to capture a maximum number of intruders in that partition. The proposed DMRST-MTA approach is capable of protecting a convex territory with partial observability against multiple intruders. For a fair comparison of both approaches, we have considered a restrictive case for DMRST-MTA to match the conditions given in [11]. We have considered the same simulation setup, where the territory is a unit circle, and intruders are radially moving toward the territory with a speed of 0.5 m/s and a speed ratio of 1. The intruders are spawn according to a Poisson process with an arrival rate of 4 over a period of $T = 30$ s.

The studies are conducted for the following distributions: 1) uniform distribution; 2) von Mises distribution with a mean = 0 and $\kappa = 2$; 3) bimodal distribution, which is generated by combining two von Mises distributions with a mean of 0 and a mean of 2.5; and 4) time-varying distribution, which is generated by changing the mean of the von Mises distribution at every 10 s to a mean whose values are $\{0, \pi, 3\pi/2\}$.

The results for the different distributions are given in Table II. The results with the proposed DMRST-MTA, where the defender can operate inside and on the perimeter, are given in the row of DMRST-MTA (Ω). For comparison purpose, the defenders are restricted to operate only on perimeter, and these results are shown in the row of DMRST-MTA ($\partial\Omega$). From Table II, one can see that the proposed DMRST-MTA approach performs better in all cases compared to both naive and AP approaches. For the uniform and bimodal distribution, the success is improved by 1%, while for von Mises and time-varying distribution, the success is improved by 5% compared to the adaptive portioning approach.

Overall, based on results, one can say that the DMRST-MTA is a scalable solution to neutralize more intruders with a smaller number of defenders. It can also handle partial observability and is robust to uncertainty in the estimation of the spatiotemporal tasks.

VII. CONCLUSION

In this article, a new DMRST-MTA approach was presented for solving the PDP. In the problem studied here, a smaller number of defenders with partial observability among themselves were defending a convex territory against a large and varying number of intruders. An intruder was estimated to reach the perimeter at a specific location and at a specific time based on the velocity of the intruder. A spatiotemporal task was defined for a defender to reach that specific location at that specific time to neutralize that intruder. The PDP was converted into an MRST-MTA problem. A modified CBBA, based on the composite loss function and the general cost function, was used to solve the problem of assigning defenders to multiple intruders in a scalable decentralized fashion. Monte Carlo simulation studies clearly showed that the speed ratio, arrival rate, and sensor radius significantly influence the mission success. If the speed ratio is higher (3 or more), then the success rate is higher and can reach 100%. As the interintruder radial distance increases, the success rate increases linearly. If the sensing radius is greater than the maximum distance between any two points in the convex territory, then defenders achieve better performances. The DMRST-MTA approach shows 1–5% improvements in success percentage for different arrival distributions of intruders compared to the decentralized AP approach. The proposed DMRST-MTA approach is robust against uncertainties in the heading direction of the intruders. Further work is needed to develop a method to find an optimal utilization of a dynamic number of defenders to handle any number of intruders and also handle agile intruders.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable suggestions, which improved the quality of this article.

REFERENCES

[1] A. Solodov, A. Williams, S. Al Hanaei, and B. Goddard, "Analyzing the threat of unmanned aerial vehicles (UAV) to nuclear facilities," *Secur. J.*, vol. 31, no. 1, pp. 305–324, 2018.
[2] D. Shishika and V. Kumar, "Local-game decomposition for multiplayer perimeter-defense problem," in *Proc. IEEE Conf. Decis. Control*, 2018, pp. 2093–2100.
[3] D. Shishika, J. Paulos, and V. Kumar, "Cooperative team strategies for multi-player perimeter-defense games," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2738–2745, Apr. 2020.
[4] D. Shishika, J. Paulos, M. R. Dorothy, M. A. Hsieh, and V. Kumar, "Team composition for perimeter defense with patrollers and defenders," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 7325–7332.
[5] J. Paulos, S. W. Chen, D. Shishika, and V. Kumar, "Decentralization of multiagent policies by learning what to communicate," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 7990–7996.
[6] D. Shishika and V. Kumar, "A review of multi agent perimeter defense games," in *Proc. Int. Conf. Decis. Game Theory Secur.*, 2020, pp. 472–485.
[7] N. Agmon, S. Kraus, and G. A. Kaminka, "Multi-robot perimeter patrol in adversarial settings," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 2339–2345.
[8] M. Chen, Z. Zhou, and C. J. Tomlin, "A path defense approach to the multiplayer reach-avoid game," in *Proc. 53rd IEEE Conf. Decis. Control*, 2014, pp. 2420–2426.
[9] S. L. Smith, S. D. Bopardikar, and F. Bullo, "A dynamic boundary guarding problem with translating targets," in *Proc. IEEE Conf. Decis. Control*, 2009, pp. 8543–8548.
[10] S. Bajaj and S. D. Bopardikar, "Dynamic boundary guarding against radially incoming targets," in *Proc. IEEE 58th Conf. Decis. Control*, 2019, pp. 4804–4809.
[11] D. G. Macharet, A. K. Chen, D. Shishika, G. J. Pappas, and V. Kumar, "Adaptive partitioning for coordinated multi-agent perimeter defense," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 7971–7977.
[12] E. S. Lee, D. Shishika, and V. Kumar, "Perimeter-defense game between aerial defender and ground intruder," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 1530–1536.
[13] 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, Aug. 2009.
[14] E. Bakolas and P. Tsiotras, "Optimal pursuit of moving targets using dynamic Voronoi diagrams," in *Proc. 49th IEEE Conf. Decis. Control*, 2010, pp. 7431–7436.

[15] A. Von Moll, D. Casbeer, E. Garcia, D. Milutinović, and M. Pachter, "The multi-pursuer single-evader game," *J. Intell. Robot. Syst.*, vol. 96, no. 2, pp. 193–207, 2019.
[16] A. Pierson, Z. Wang, and M. Schwager, "Intercepting rogue robots: An algorithm for capturing multiple evaders with multiple pursuers," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 530–537, Apr. 2017.
[17] R. Isaacs, *Differential Games: A Mathematical Theory With Applications to Warfare and Pursuit, Control and Optimization*, North Chelmsford, MA, USA: Courier Corporation, 1999.
[18] E. Garcia, D. Casbeer, K. D. Pham, and M. Pachter, "Cooperative aircraft defense from an attacking missile using proportional navigation," in *Proc. AIAA Guid., Navigat., Control Conf.*, 2015, Art. no. 0337.
[19] E. Garcia, D. W. Casbeer, A. Von Moll, and M. Pachter, "Cooperative two-pursuer one-evader blocking differential game," in *Proc. Amer. Control Conf.*, 2019, pp. 2702–2709.
[20] L. Liang, F. Deng, Z. Peng, X. Li, and W. Zha, "A differential game for cooperative target defense," *Automatica*, vol. 102, pp. 58–71, 2019.
[21] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via pairwise outcomes," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1451–1457, Mar. 2017.
[22] J. F. Fisac, M. Chen, C. J. Tomlin, and S. S. Sastry, "Reach-avoid problems with time-varying dynamics, targets and constraints," in *Proc. 18th Int. Conf. Hybrid Systems: Comput. Control*, 2015, pp. 11–20.
[23] R. Yan, Z. Shi, and Y. Zhong, "Reach-avoid games with two defenders and one attacker: An analytical approach," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1035–1046, Mar. 2019.
[24] R. Yan, Z. Shi, and Y. Zhong, "Task assignment for multiplayer reach–avoid games in convex domains via analytical barriers," *IEEE Trans. Robot.*, vol. 36, no. 1, pp. 107–124, Feb. 2020.
[25] R. Yan, Z. Shi, and Y. Zhong, "Guarding a subspace in high-dimensional space with two defenders and one attacker," *IEEE Trans. Cybern.*, early access, doi: 10.1109/TCYB.2020.3015031.
[26] P. Agharkar, S. D. Bopardikar, and F. Bullo, "Vehicle routing algorithms for radially escaping targets," *SIAM J. Control Optim.*, vol. 53, no. 5, pp. 2934–2954, 2015.
[27] E. Garcia, D. W. Casbeer, A. Von Moll, and M. Pachter, "Multiple pursuer multiple evader differential games," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2345–2350, May 2020.
[28] J. L. Salmon, L. C. Willey, D. Casbeer, E. Garcia, and A. V. Moll, "Single pursuer and two cooperative evaders in the border defense differential game," *J. Aerosp. Inf. Syst.*, vol. 17, no. 5, pp. 229–239, 2020.
[29] N. Mohanty and S. Sundaram, "Scaffolding reflection in reinforcement learning framework for confinement escape problem," 2020, *arXiv:2011.06764*.
[30] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1495–1512, 2013.
[31] A. Khamis, A. Hussein, and A. Elmogy, "Multi-robot task allocation: A review of the state-of-the-art," in *Cooperative Robots Sensor Networks*. New York NY, USA: Springer, 2015, pp. 31–51.
[32] E. G. Jones, M. B. Dias, and A. Stentz, "Learning-enhanced market-based task allocation for oversubscribed domains," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2007, pp. 2308–2313.
[33] R. Cui, J. Guo, and B. Gao, "Game theory-based negotiation for multiple robots task allocation," *Robotica*, vol. 31, no. 6, pp. 923–934, 2013.
[34] S. Chopra and M. Egerstedt, "Spatio-temporal multi-robot routing," *Automatica*, vol. 60, pp. 173–181, 2015.
[35] S. Chopra, G. Notarstefano, M. Rice, and M. Egerstedt, "A distributed version of the Hungarian method for multirobot assignment," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 932–947, Aug. 2017.
[36] L. Brunet, H.-L. Choi, and J. How, "Consensus-based auction approaches for decentralized task assignment," in *Proc. AIAA Guid. Navigat. Control Conf. Exhib.*, 2008, Art. no. 6839.
[37] M. P. Fanti, A. M. Mangini, G. Pedroncelli, and W. Ukovich, "A decentralized control strategy for the coordination of AGV systems," *Control Eng. Pract.*, vol. 70, pp. 86–97, 2018.
[38] E. Nunes, M. Manner, H. Mitiche, and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints," *Robot. Auton. Syst.*, vol. 90, pp. 55–70, 2017.
[39] S. A. Nelke, S. Okamoto, and R. Zivan, "Market clearing-based dynamic multi-agent task allocation," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 1, 2020, Art. no. 4.

Shridhar Velhal is working toward the Ph.D. degree with the Department of Aerospace Engineering, Indian Institute of Science, Bengaluru, India.

His research interests include dynamics and control of autonomous systems, with a particular focus on multiagent systems, territory protection games, spatiotemporal tasks, and multitask allocation.

Suresh Sundaram (Senior Member, IEEE) received the Ph.D. degree in aerospace engineering from the Indian Institute of Science, Bengaluru, India, in 2005.

He is currently an Associate Professor with the department of Aerospace Engineering, Indian Institute of Science. From 2010 to 2018, he was an Associate Professor with the School of Computer Science and Engineering, Nanyang Technological University, Singapore. His research interests include flight control, unmanned aerial vehicle design, machine learning, optimization, and computer vision.

Narasimhan Sundararajan (Life Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Illinois at Urbana–Champaign, Urbana, IL, USA, in 1971.

From 1971 to 1991, he was with the Vikram Sarabhai Space Centre, Indian Space Research Organization, Trivandrum, India. Since 1991, he had been a Professor (Retd.) with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently a Technical Consultant in WIRIN Project with the Indian Institute of Science, Bengaluru, India. His current research interests include spiking neural networks, neuro-fuzzy systems, and optimization with swarm intelligence.