Research Article

A Dynamic Territorializing Approach for Multiagent Task Allocation

Mohammad Islam , Mehdi Dadvar , and Hassan Zargarzadeh

Electrical Engineering Department, Lamar University, Beaumont, TX 77710, USA

Correspondence should be addressed to Hassan Zargarzadeh; hzargarzadeh@lamar.edu

Received 29 November 2019; Revised 25 March 2020; Accepted 11 April 2020; Published 13 May 2020

Academic Editor: Marcio Eisencraft

Copyright © 2020 Mohammad Islam et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In this paper, we propose a dynamic territorializing approach for the problem of distributing tasks among a group of robots. We consider the scenario in which a task comprises two subtasks—detection and completion; two complementary teams of agents, hunters and gatherers, are assigned for the subtasks. Hunters are assigned with the task of exploring the environment, i.e., detection, whereas gatherers are assigned with the latter subtask. To minimize the workload among the gatherers, the proposed algorithm utilizes the center of mass of the known targets to form territories among the gatherers. The concept of center of mass has been adopted because it simplifies the task of territorial optimization and allows the system to dynamically adapt to changes in the environment by adjusting the assigned partitions as more targets are discovered. In addition, we present a game-theoretic analysis to justify the agents' reasoning mechanism to stay within their territory while completing the tasks. Moreover, simulation results are presented to analyze the performance of the proposed algorithm. First, we investigate how the performance of the proposed algorithm varies as the frequency of territorializing is varied. Then, we examine how the density of the tasks affects the performance of the algorithm. Finally, the effectiveness of the proposed algorithm is verified by comparing its performance against an alternative approach.

1. Introduction

Fair and efficient distribution of tasks in an area-coverage problem among the agents in a multirobot system is a common objective and has been widely considered in the literature. Multirobot systems when utilized in an area-coverage problem possess the advantage of being able to complete the global mission in a shorter period and offer improved robustness against single robot failure [1]. In addition, a team of agents is likely to offer superior performance since they can be distributed over different parts of the operating environment and often carry out dissimilar tasks if the system consists of heterogeneous robots [2]. Furthermore, in situations in which the completion of a single task requires decomposition into a sequence of subtasks, multirobot systems offer more flexibility as compared to a system consisting of a single robot with multiple sensors for accomplishing the assignment [3]. Consequently,

multirobot systems have been gaining popularity and are expected to take more imperative roles in applications that require fast response and pose a high level of risks for humans [4], e.g., applications that require surveillance where it is essential to visit every section of the environment regularly to inspect the frequency of anomalies and intruders may be dangerous for humans, as well as for tasks that are monotonous and repetitive [5]. However, the problem of multirobot task allocation (MRTA) gives rise to several challenges that have been investigated in the past two decades [6–8]. Distributing the workload fairly, so that a specific agent is not overloaded, remains a challenging problem, and it is imperative to develop techniques that can potentially solve the workload balance assignment issue. Moreover, in a dynamic setting where the environment is initially unknown to the agents, i.e., the number and locations of the tasks are unknown, the workload balancing assignment becomes even more challenging.

In this paper, we consider the workload balancing assignment issue in a nature-inspired problem proposed in [9, 10], called “hunter and gatherer.” More specifically, the dynamic MRTA problem is transfigured into a scenario where each task is composed of two sequential sub-tasks—exploration and completion. Hence, there are two sets of teams of robots: hunters, who are quickly able to explore the unknown environment and detect the locations of the tasks, and gatherers, who are a team of heavy-duty dexterous robots assigned with the completion of the detected tasks. The two sets of robots bring about heterogeneity in the system, as they possess different cognitive skills and speed profiles. Previous studies have shown that heterogeneous systems are well suited for several real-world problems, such as the urban search and rescue (USAR) [11, 12], agricultural field operations [13], environmental monitoring [14], and surveillance [5, 14, 15]. In terms of a real-world setting for the hunter and gatherer approach, we can consider the USAR in a disaster site where several victims have been stranded in unknown locations and need immediate rescue operations. For such an operation, the hunters could be a group of lightweight unmanned aerial vehicles (UAVs) as they would offer agility and provide faster exploration and their mission would be to search the site and locate the victims. On the other hand, the gatherers could be a group of unmanned ground vehicles (UGVs) whose mission would be to rescue the victims once their locations have been discovered by the UAVs. Both sets of robots should be equipped with multiple cameras such as both infrared camera and visible light which enables object detection and obstacle avoidance. However, a practical implementation of our proposed framework has been kept for future research and does not fall within the scope of this work. We encourage the readers to review [16, 17], which considers practical implementation of similar operations and offers insight into solutions related to exploration, localization, and mapping of UGVs and UAVs.

Balancing the workload amongst the agents in this work is based on environment partitioning through locational optimization. The setting considered consists of a mission composed of a set of spatially distributed tasks; i.e., they are directly associated with a fixed location in the environment. To balance the workload, we propose a dynamic territorializing algorithm that utilizes the center of mass of the known targets to form territories amongst the gatherers. As the hunters explore the environment and discover new targets, they pass this information to a central planner (CP) who uses the locations of the discovered targets to form partitions amongst the gatherers. Note that the terms territorializing and partitioning essentially have the same meaning and will be used interchangeably throughout the rest of this paper. There are several advantages of forming territories amongst the gatherers, which has been discussed in Section 4.1. Apart from being advantageous in balancing the workloads, the formation of territories offers advantages in terms of reduced overlapping of areas covered by the gatherers. The concept of center of mass has been adopted because it simplifies the task of locational optimization and allows the system to dynamically adapt to changes in the

environment by adjusting the assigned partitions as more targets are discovered.

The rest of this paper is organized as follows. Section 2 provides a summary of the related work. In Section 3, we provide a schematic model of the system and introduce the problem statement. In Section 4, we present the dynamic territorializing algorithm that is employed to form territories amongst the gatherers. Section 5 provides a Nash equilibrium analysis that verifies why it is reasonable for a gatherer to stay within its assigned territory. In Section 6, statistical analysis on simulation results is presented, followed by the conclusion in Section 7.

2. Related Work

In this section, we focus on the relevant related work in the literature. More specifically, we have structured it into two different sections. The first section presents several studies that have focused on the problem of task allocation as a primitive, global, and prominent problem in the context of MRTA. The subsequent section highlights some of the research studies carried out towards the work-balancing assignment problem in MRTA.

The field of multirobot research is not newly developed, and many architectonics have been considered over the years to tackle various aspects of dynamic problems in MRTA. In [18], the authors provided a potential framework to analyze or predict the behavior of a multirobot system, focusing on MRTA. The authors provided three axes—single-task robots (ST) versus multitask robots (MT), single-robot tasks (SR) versus multirobot tasks (MR), and instantaneous assignment (IA) versus time-extended assignment (TA); they also analyzed six MRTA architectures based on the combinations of these three axes to demonstrate how relevant theories from operations research and combinatorial optimization can be used for analysis and deeper comprehension of the prevailing approaches to task allocation. A mathematical model of a general dynamic task allocation was presented in [19], where the objective was to achieve an intended task allocation without explicit communication and global knowledge. Nonetheless, in some cases, capturing the communication amongst the agents played an imperative role; to this end, Liemhetcharat and Veloso [20] introduced a novel weighted synergy graph model to capture new interactions amongst the agents. Market-based approaches utilizing an auction mechanism for task allocation have been used to solve ST-SR and ST-MR problems [21, 22]. Although market-based approaches can essentially meet the practical demands of robot teams for both distributed and centralized approaches, their scalability is limited by the computation and communication needs that arise from increasing auction frequency, bid complexity, and planning demands [23]. Common applications of MRTA also include environment exploration and mapping. In [24], a decentralized cooperative exploration strategy based on a sensor-based random graph method was proposed, which utilized cooperation and coordination mechanisms to avoid conflicts and improve efficiency. The problem of exploring an unknown environment with a team

of robots was modelled as an ST-SR problem in [25], where the selection of targets by the robots was dependent on distance and utility. In recent years, deep-learning techniques have been utilized to a great deal in the MRTA community for exploration, path-planning, and cooperation between several robotics systems [26]. For example, a deep Q-network algorithm was proposed in [27] which focused on improving the learning efficiency. A deep learning cooperative (DL-Cooper) method utilizing the cloud robotic architecture for trail following task was proposed in [28].

Workload-balancing assignment and partitioning of the environment are both an emerging field of research in robotics systems. In the context of MRTA, several studies have contributed towards providing feasible solutions to efficiently assign tasks to the agents. In [29], the authors investigated the problem of fairly dividing a single global task among a group of heterogeneous robots. The task distribution problem was formulated as a fair subdivision problem, and a centralized algorithm was presented to evaluate the allocations for each robot. Each robot could define their preference over parts of the task related to their sensing capabilities and speed profiles. A multirobot coordination approach for informative sampling based on environment partitioning was proposed in [30], where a central planner directed robots to different partitions of the environment which is formed according to the effort needed to explore each region. The system utilized a priori knowledge of the environment in this work. In [31], to balance the workloads for service vehicles over a geographic territory, the authors provided a fast algorithm, based on infinite-dimensional optimization formulation. The proposed algorithm divides the operating region into compact, connected territories of equal area with a vehicle depot assigned to each of them. The authors in [32] proposed an area decomposition algorithm to reduce the spatial interference between the robots in the operating environment. The algorithm divides the working environment into cells which are dynamically assigned to the robots. Since each robot operates in its cell, the spatial interference is reduced, and more time is allocated towards a domain task. In addition to the studies mentioned, workload balancing, or the environment partitioning problem, has also been considered in multirobot patrol, which is a fundamental application of a multirobot system. For example, in [3], a Multilevel Subgraph Patrolling (MSP) algorithm based on balanced graph partitioning was proposed to deal with the assignment of a local patrolling task. In this work, it is assumed that the environment is known and the proposed method assigns exclusive regions to each agent to ensure that work redundancy is reduced and collision between the operating robots is eliminated. In [33], a distributed self-organized graph partitioning approach was proposed that can partition a graph into nonoverlapping subgraphs without the presence of a centralized entity. The proposed self-organized autonomous algorithm required less synchronization and only local information since it required no central entity. To detect the number of incoming visitors in an area, the authors in [34] utilized a territorial approach to partition the environment into territories. Additionally, to address the

challenging issue of workload balancing, dynamic partitioning strategies that utilized information about visitor trends were proposed. An area-partitioning method for cooperative cleaning robots in an environment consisting of obstacles was proposed in [35]. The proposed method partitions the area based on a model of dirt accumulation in a bottom-up manner.

When there are synchronization and precedence constraints (SP) for specifying ordering constraints for TA problems, the MRTA problem is often referred to as a TA:SP problem. In the previous work, the authors aimed to provide solutions for an MT-MR-TA:SP problem, which is a ubiquitous problem in a wide variety of fields, such as USAR and agricultural field operations. To the best of our knowledge, the workload balancing issue, or the environment partitioning problem, considered in the literature, so far do not fully capture the dynamics of the “hunter and gatherer” approach proposed in the previous work. Hence, this work serves as an extension in terms of the workload-balancing assignment issue via environment partitioning (territorializing) in the context of an MT-MR-TA:SP problem.

3. Preliminaries and Assumptions

3.1. Schematic Model. To motivate the proposed work, consider the schematic model shown in Figure 1. Some tasks are spread over the field, labelled as “Target” in the figure. In this work, we define the tasks as getting to the locations of the targets and picking them up. Hence, each task further consists of two sequential subtasks. The locations of the tasks are initially unknown, and hence, it is imperative to locate them first. For this purpose, an agile group of robots known as hunters explore the unknown environment; essentially, this is the first subtask for a specific target. The exploration is indicated by the blue trails of the hunters, as labelled in the figure, and the specific regions they have explored, as marked by lime-green color. As the hunter keeps exploring the environment, it can identify the location of the nearby targets. A Central Planner (CP) receives the locations of the discovered targets. The CP is a powerful data center whose primary goal is to collect the locations of the discovered targets and assist the gatherers in assigning their territories based on the information received from the hunters. Gatherers are another team of robots; they are heavy-duty robots capable of accumulating the targets to complete the second subtasks. Upon receiving the locations of the discovered targets from the CP, the gatherers use a “shortest path-planning algorithm” [36] to get to the nearest available location first. Besides, the CP is responsible for performing other heavy computations for the proposed algorithm, which will be discussed later.

3.2. Problem Statement. Suppose there are m tasks which are randomly distributed over a given field F of area Φ unit square. Let $T = \{T_1, T_2, \dots, T_m\}$ represent the set of tasks. For any given task, it is essential to discover its location

FIGURE 1: A schematic model of the system showing how the hunters and gatherers perform their respective operations while being assisted by a Central Planner (CP).

before it can be completed; after the discovery of its location, when an agent travels to the known location of the task, it is assumed that the task is complete. Let $A = \{A^h, A^g\}$ represent the set of two teams, where $A^h = \{a_1^h, \dots, a_{n_h}^h\}$ and $A^g = \{a_1^g, \dots, a_{n_g}^g\}$ represent the set of hunters and gatherers, respectively; the notations k and l will be used to refer to the k^{th} hunter and l^{th} gatherer, respectively, such that $1 \leq k \leq n_h$ and $1 \leq l \leq n_g$. Since T represents the set of tasks, we can represent each task as a combination of the subtasks, i.e., $T_i = \{t_i^h, t_i^g\}$, where t_i^h represents hunting and t_i^g represents gathering and $1 \leq i \leq m$. The reward and cost of accomplishing a task $t_i^h(t_i^g)$ for $a_i^h(a_i^g)$ are denoted by $c_{k,i}^h(c_{l,i}^g)$ and $r_{k,i}^h(r_{l,i}^g)$, respectively. Territorializing involves dividing F amongst the gatherers in a way such that each agent receives a specific share of the total field with some tasks lying within the boundaries of the assigned territory. This will ensure that the gatherers only cover a specific portion of the entire area, thus balancing the workload and the distance covered per completed task. Hence, this work is associated with solving the workload-balancing assignment issue.

Motivated by the notion of the cake cutting problem [37], we use the concept of fair division theory [38] to ensure that the territory assigned to an agent is fair. Fair division deals with the issue of distributing resources among a certain number of interested agents such that each agent receives a fair share. Let $A^g = \{a_1^g, \dots, a_{n_g}^g\}$ receive $\{\varphi_1, \dots, \varphi_{n_g}\}$ unit square of area, and the number of tasks available for a_l^g in their assigned region is $\{t_1^g, \dots, t_{\kappa_l}^g\}$, respectively, where κ_l is the total number of tasks available in φ_l . Furthermore, let us assume that $\zeta_l = \{(x_1, y_1), \dots, (x_p, y_p)\}$ represents the set of coordinates contained within the area φ_l , where $p = |\zeta_l|$, and $|\cdot|$ represents the cardinality of a set and each coordinate corresponds to a square unit. Using the following notations, the territorializing problem for the l^{th} gatherer can be represented by the following optimization problem:

$$\max \left(\varphi_l + \sum_{i=1}^{\kappa_l} r_{l,i}^g \right), \quad (1)$$

$$\text{subject to } \varphi_l + \sum_{i=1}^{\kappa_l} r_{l,i}^g \cong \frac{\Phi + \sum_{i=1}^m r_{l,i}^g}{l} \pm \delta, \quad (2)$$

$$(x_l^g, y_l^g) \in \zeta_l, \quad (3)$$

$$\zeta_l \cap \zeta_{l \neq 1} = \emptyset, \quad (4)$$

$$\zeta_1 \cup \zeta_2 \dots \cup \zeta_l = F, \quad (5)$$

where (x_l^g, y_l^g) is the initial location of a_l^g after territorializing. The objective of (1) is to maximize the area of the territories and the number of tasks available within that region, defined by $\varphi_l + \sum_{i=1}^{\kappa_l} r_{l,i}^g$, which is the share received by a_l^g , i.e., the territory φ_l received and the reward $r_{l,i}^g$, where $i = 1, \dots, \kappa_l$, associated with each available target in φ_l . In the optimization problem, (2) ensures that all the gatherers receive shares which are approximately equal; subsequently, the next constraint ensures that initial locations of the gatherers lie within their assigned territory. Then, the following constraint ensures that the assigned territories do not share any common region, and finally, the last constraint makes sure that all parts of F have been allocated. Note that, for the purpose of area optimization, we assume that the unit for rewards is equivalent to the unit of area, which is unit square and it is a scalar constant. In addition, to determine how many times the optimization is to be performed, we will define two more parameters—density of targets and workload, which will be discussed later in Section 4.4.

Since it is assumed that the gatherers are heavy-duty robots, they are less agile in motion. Consequently, having them operate within a certain territory will speed up the time required to complete a certain number of tasks in a given mission; simultaneously, it will also minimize the workload as compared to one where there are no certain territories defined for the agents and they are free to move anywhere within the field. Nonetheless, it is nontrivial to divide the region under operation amongst the gatherers, because it is imperative to ensure that the approach utilized to form territories is by no means “unfair” to any of the gatherers.

3.3. Assumptions. We consider the following assumptions throughout the paper:

- (1) Each task has a fixed location, and initially, their locations are unknown to all the agents.
- (2) Agents of a specific team are identical.
- (3) Hunters travel at a faster speed compared to the gatherers, i.e., hunters are lightweight, while gatherers are heavy-duty robots.
- (4) The cost of accomplishing a task for both sets of agents is proportional to the distance covered.
- (5) All agents are rational; they intend to maximize their expected utility.

- (6) A task is assumed to be partially or fully complete when an agent reaches the location of the target. Particularly, for a specific task, when the hunter reaches the location of a task, it is partially complete; when one of the gatherers reaches the location after being discovered, the task is assumed to be fully complete.
- (7) The initial distribution of the unknown targets is assumed to be uniform over the field as grid shape; the territories are also formed based on this assumption, as shown in an example in Figure 2.

For a dynamic environment, where the locations of tasks are not initially known, it is imperative to devise appropriate methodologies capable of fairly dividing the area on which the agents operate at any given point of time. In the following section, we present the proposed method which can dynamically form territories amongst the gatherers.

4. The Proposed Method

4.1. Territorializing amongst the Gatherers. The underlying principle behind territorializing amongst a certain type of agents relies on how many of them are operating; i.e., the number of those agents determines the number of territories. Note that the A^h 's are agile in their movement, which enables them to quickly explore the environment and detect the location of tasks, while A^g 's are involved with specific movement from one target location to another. Owing to their differences in characteristics, it can be claimed that the gatherers are the heavy-duty agents, and employing a certain number of gatherers will be more expensive than employing the same number of hunters; therefore, it is desirable to have $n_h \geq n_g$. If we were to territorialize amongst the hunters, it would mean that the number of territories would be more than n_g and it would not be possible to ensure that there is at least one agent ready for the completion of a discovered task in a specific territory. Although the formed territories could again be partitioned together to have their number equal to n_g , it would further complicate the task and introduce redundancy. On the other hand, territorializing amongst the gatherers is more suitable, since it would ensure that there is one gatherer always ready for the competition of task in a territory and allow specific hunters to be allocated to a territory if needed, ensuring that the gatherers do not have to wait too long or sit idle for tasks to be discovered.

4.2. Using Center of Mass to Form Territories. The center of mass of a system is the average position of all parts of the system, weighted according to their masses. In order to dynamically divide the area among the gatherers in a suitable time period, we will use the concept of center of mass to identify the location (center) from which the territories can be formed. For the ease of discussion, consider an ideal scenario with 3 gatherers as seen in Figure 2, where F has 12 tasks, distributed uniformly over the field. The field consists of grids, where each grid corresponds to a coordinate (x, y) , where x and y are integers. Essentially, each target occupies one grid space in F . It is

FIGURE 2: Dividing the area under operation from the center among 3 gatherers when the targets are uniformly distributed over the space. Each coordinate corresponds to a grid; hence, each target occupies one grid space.

straightforward to ascertain from Figure 2 that the center of mass will be located exactly at the center, given that the weights of the tasks are equal. Owing to this orientation, using simple geometry, 3 territories, OGFC, OGDE, and OEABF, can be formed, each consisting of 4 tasks for each of the gatherers. Since each of the gatherers receives an equal number of tasks distributed over an equal space, it is justifiable to surmise that the division is fair. Nonetheless, this still does not clarify the benefit of utilizing center of mass as the point from which the territories are formed. Consider another situation as portrayed by Figure 3, where the upper right section of the field has a higher concentration of targets. Under such circumstances, the center of mass can be obtained using

$$O(x, y) = \left(\frac{\sum_{i=1}^m w_i x_i}{\sum_{i=1}^m w_i}, \frac{\sum_{i=1}^m w_i y_i}{\sum_{i=1}^m w_i} \right), \quad (6)$$

where O is the center of mass of the system and w_i and (x_i, y_i) are the weights and positions of i^{th} target, respectively. Owing to the densely populated region on the upper right envelope of the field, the center of mass will shift towards it. This provides a more efficient method of partitioning, as it ensures that the territory with the highest density of targets has the least area, increasing the possibility of more targets being discovered within the other territories as the hunters continue exploring. This is evident in Figure 3; as points A , B , C , and D are fixed, during territorializing, the position of the center of mass at O and of points E , F , and G change. It is seen that the region of most densely populated targets is enclosed by $OEBF$, and hence, it has the least area. Note that the region least densely populated with targets has the maximum area, $OGDAE$. Similarly, the territory $OGCF$, which is slightly more densely populated, has a slightly smaller area in comparison to $OGDAE$. Essentially, territorializing utilizes the area and target density to ensure that the partitioning is approximately fair.

FIGURE 3: An example demonstrating the advantages of utilizing the proposed method to form territories. The center of mass lies closest to the region populated more densely with available targets.

Algorithm 1 illustrates the territorializing procedure for A^g . The algorithm utilizes the optimization constraints introduced in (1) to form the territories for A^g . First, the center of mass O from which the territories are to be formed is calculated. Then, the points are initialized. Note that any points on the boundaries of field F can be utilized as the initial points, but for the purpose of our analysis, we initialize the points on the boundaries with the assumption that all targets are uniformly distributed and then adjust the points accordingly. Afterwards, the share received for each agent is calculated. If the share received is equivalent to $l^{-1}(\Phi + \sum_{i=1}^m r_{l,i}^g) \pm \delta$, then the algorithm stops calculating the territory for the respective agent; otherwise, the algorithm adjusts the points accordingly and checks if the condition in line 8 is met. The algorithm repeats until the condition is met for all the gatherers. Note that the proposed algorithm is not restricted to only 3 gatherers; this number has been used as an example only for the ease of analysis. It can be utilized for a higher number of A^g ; although the scalability of the proposed algorithm in terms of the number of gatherers is beyond the scope of this paper, it will be considered as a potential scenario for our future work.

4.3. Exploration and Path-Planning. In this section, the exploration for A^h and path-planning for A^h and A^g are discussed. A^h utilizes the algorithm introduced in [39] to explore the unknown environment, an algorithm based on the concept of frontiers, regions on the boundary between open space, and unexplored space. Note that the operating environment is spatially visualized as Cartesian grids containing cells, and at a given time, an agent can visit a specific cell. An example of the frontier-based exploration has been illustrated for a hunter in Figure 4. Initially, the hunter starts exploring from the center of the environment as shown in Figure 4(a). Information about the immediate nearest unexplored grids becomes available which is indicated by grids colored black. For example, if the center of the map is at (50,

50) in Figure 4(a), then information about the following nearest grids becomes available: (50, 51), (49, 51), (49, 50), (49, 49), (50, 49), (51, 49), (51, 50), and (51, 51). Initially, the hunter moves upward, so its next position will be at (50, 51). At this location, the newly discovered grids are (49, 52), (50, 52), and (51, 52); information about the other nearest grids around it has already been discovered when the hunter was at the center. In Figure 4(b), the trail of the hunter after some time is portrayed. As the hunter continues to explore the environment, the grids colored black indicates that these regions have been explored; on its way if any location of the target is discovered, they are marked by a blue flag as shown in the same figure.

For the purpose of this research, where the exploration of the same environment takes place in a repeated manner and there exists at least a minimum number of agents for exploring the environment, some modifications to the algorithm were made in order to provide enhanced coverage of the environment. Particularly, as A^h continues to explore the environment, we assign ages Λ^{age} to the explored regions (cells), since these agents often must come back to a previously explored region. We also set an expiration time τ^{expired} which determines for A^h whether or not to revisit an available region, depending on the value of the age and the expiration time. Let $\Psi_k = \{\psi_1, \dots, \psi_{n_{\text{cells}}}\}$ represent the list of cells available for a_k^h to visit, where n_{cells} is the number of cells. If the age of all the cells in Ψ_k is zero, then a_k^h picks a cell randomly from the list. If the age of all the cells in Ψ_k is greater than zero, then a_k^h only keeps those ψ_i 's for which $\Lambda_i^{\text{age}} > \tau^{\text{expired}}$ and then picks ψ_i for which Λ_i^{age} is maximum. Finally, if the age of some cells (but not all) in the list equals zero, then a_k^h must first eliminate ψ_i for which $\Lambda_i^{\text{age}} \neq 0$ and then pick randomly from the new list. Algorithm 2 summarizes the modified frontier-based exploration for A^h . Essentially, these agents prioritize the cells that have not been explored before, and when they must revisit explored locations, the expiration time is utilized to prioritize the cells that were explored earlier.

For path-planning, both A^h and A^g utilize the A^* search algorithm [40], which plans the shortest multidestination temporary path.

4.4. Utility, Expected Utility, and Workload. To analyze the behavior of the agents, it is imperative to define several parameters that influence their decision-making over the course of a mission. Particularly, we first introduce instantaneous utility, which represents the profit any agents can make upon the completion of a task. Then, we introduce expected utility for A^g , which is necessary because it will be used later on in Section 5 to validate why these agents do not have any incentive to move away from their assigned territory in situations in which the closest target available to them is from another territory. Finally, we want to be able to measure the performance of the proposed algorithm in terms of a suitable parameter. Since the purpose of territorializing is to ensure that the workload is balanced amongst the gatherers, we introduce a function for it. Workload-balancing or job-scheduling problems on

```

(1) function territorialize ( $O, \varphi_l, \xi_l$ )
(2)    $O \leftarrow$  calculate the center of mass
(3)    $\xi_l \leftarrow$  initialize the points
(4)   flag  $\leftarrow$  false
(5)   for each  $a_l^g$ 
(6)     while flag = false do
(7)        $\varphi_l + \sum_{i=1}^{k_l} r_{l,i}^g \leftarrow$  calculate the share received by  $a_l^g$ 
(8)       if  $\varphi_l + \sum_{i=1}^{k_l} r_{l,i}^g \cong l^{-1} (\Phi + \sum_{i=1}^m r_{l,i}^g) \pm \delta$  then
(9)         results  $\leftarrow [\xi_l, \varphi_l]$ 
(10)        flag  $\leftarrow$  true
(11)        break for
(12)      else
(13)        adjust the respective points in  $\xi_l$ 
(14)      end if
(15)    end while
(16)  end for
(17) return results

```

ALGORITHM 1: Territorializing optimization for A_g .

FIGURE 4: Portraying the exploration of the hunter: (a) the initial position of the agent and (b) trail of the hunter after some time as it explores the environment.

identical machines have been thoroughly studied in the literature, and to the best of our knowledge, any measure of dispersion used in statistical practice can be used as a performance criterion for workload balancing [41]. Hence, the Normalized Sum of Squares for Workload Deviations (NSSWD) criteria proposed in [41, 42] have been utilized in this work to quantify the workload at a given instance.

Definition 1 (instantaneous utility function for a_l^g). The utility function $u(t_i^g)$ for a_l^g in accomplishing a task t_i^g is defined as the profit made by a_l^g in completing that task. Specifically, it is the cost of completing the task subtracted from the reward:

$$u(t_i^g) = r_{l,i}^g - c_{l,i}^g, \quad (7)$$

where $c_{l,i}^g$ is proportional to the Euclidean distance between a_l^g and t_i^g , i.e., if (x_l^g, y_l^g) is the location of a_l^g and (x_i, y_i) is the location of t_i^g , then

$$c_{l,i}^g \propto \sqrt{(x_l^g - x_i)^2 + (y_l^g - y_i)^2}. \quad (8)$$

For brevity, we have defined $u(t_i^g)$ for a_l^g ; $u(t_i^h)$ for a_k^h can be defined in a similar manner.

Definition 2 (expected utility). The expected utility associated with a gatherer a_l^g , in territory φ_l , is defined as the total

```

(1) for  $k = 1: n_h$ 
(2)  $\Psi_k = \{\psi_1, \dots, \psi_{n_{\text{cells}}}\} \leftarrow$  list of cells for  $a_h^k$ 
(3)  $\{\Lambda_1^{\text{age}}, \dots, \Lambda_{n_{\text{cells}}}^{\text{age}}\} \leftarrow$  obtain the age of cells for  $a_h^k$ 
(4) if  $\forall i (\Lambda_i^{\text{age}} = 0), i = 1, \dots, n_{\text{cells}}$  then
(5)   pick randomly from  $\Psi_k$ 
(6) elseif  $\forall i (\Lambda_i^{\text{age}} \neq 0), i = 1, \dots, n_{\text{cells}}$  then
(7)   for  $i = 1: n_{\text{cells}}$ 
(8)     if  $\Lambda_i^{\text{age}} > \tau^{\text{expired}}$  then
(9)       keep  $\psi_i$  in  $\Psi_k$ 
(10)    else
(11)      discard  $\psi_i$  from  $\Psi_k$ 
(12)    end if
(13)  end for
(14)  pick for  $\psi_i$  which  $\Lambda_i^{\text{age}}$  is maximum
(15) else
(16)  eliminate  $\psi_i, \forall i (\Lambda_i^{\text{age}} \neq 0), i = 1, \dots, n_{\text{cells}}$ 
(17)  pick randomly from  $\Psi_k$ 
(18) end if
(19) end for

```

ALGORITHM 2: Modified frontier-based exploration for A_h .

expected reward, proportional to the total number of targets present in the partition, divided by the number of players present in the partition, such that

$$U(\varphi_l, a_l^g) = E\left(\frac{\kappa_l}{n_g^{\varphi_l}}\right), \quad (9)$$

where $n_g^{\varphi_l}$ and κ_l are the number of gatherers and tasks, respectively, in territory φ_l .

Definition 3 (workload). The workload $W(t)$ at any time t is defined as the extent to which the number of targets gathered by A^g varies from the mean number of targets accumulated by the agents:

$$W(t) = \frac{\sqrt{\sum_{l=1}^{n_g} (\Gamma_l^g - \mu)^2}}{\mu}, \quad (10)$$

where μ is the average number of tasks completed by A_g and Γ_l^g is the total tasks completed by a_l^g in time t . Note that the denominator in (4) ensures that the equation is normalized, as the workload is calculated over different instances of time.

5. Nash Equilibrium Analysis for Gatherers

By proposing the territorializing optimization algorithm in the hunter and gatherer scheme, an imperative question arises: do the gatherers retain their own assigned territories? In fact, the negative answer to the question contradicts the stability of the proposed territorializing solution. To investigate the problem from the explained perspective, we apply a Nash equilibrium (NE) analysis by which we can determine whether any gatherer is motivated to deviate from its assigned territories and encounter other gatherers' territories. Basically, according to the basic definition of NE, if we can prove that the proposed solution is a NE, then no

agent would have a motivation to deviate from the proposed solution, i.e., the optimized territorializing.

To have a NE analysis, we initially need to formulate the problem from a game theory perspective. To that end, we define the problem in the following manner:

- (i) The gatherers, $A^g = \{a_1^g, \dots, a_{n_g}^g\}$, are the set of players in the game; essentially, there are n_g players in the game.
- (ii) The l^{th} player, a_l^g , in A^g has a finite set of pure strategies defined by $S_l = \{\varphi_1, \varphi_2, \dots, \varphi_{n_g}\}$, where $1 < l < n_g$ and φ_l is the territory initially assigned to a_l^g . Note that each player must choose a territory as its action from S_l . Having said that, for instance, if a_2^g chooses φ_2 , it means that it has decided to retain the initial territorializing. Furthermore, if a_2^g chooses φ_l , where $l \neq 2$, then the agent has decided to deviate from the initial partitioning. An example with 3 gatherers where a_2^g must deal with such decision-making is illustrated in Figure 5.
- (iii) The expected utility associated with a player, denoted as a_l^g , in territory φ_l is defined as total expected reward, proportional to the total number of targets present in the partition, divided by the number of players present in the partition, as has been defined in Definition 2.
- (iv) The structure of the game played by the agents is defined as a normal-form game, where each player takes an action simultaneously and the payoffs are then determined based on the defined utility function.

Definition 4 (normal-form game amongst the gatherers). Let $G = \langle A^g, S, U \rangle$ define a normal-form game amongst the gatherers, where $S = \{S_1, S_2, \dots, S_{n_g}\}$ is an n_g -tuple of pure

strategy sets one for each player and $U = \{U_1, U_2, \dots, U_1, \dots, U_{n_g}\}$ is an n_g -tuple of payoff functions in terms of expected utility function, essentially $U_i = U(\varphi_i, a_i^g)$.

Remark 1. For a set of assigned territory $\{\varphi_1, \varphi_2, \dots, \varphi_{n_g}\}$, at each iteration, the CP can monitor the number of tasks completed by A_g and the number of tasks that they will be complete after some iterations, i.e., by sorting out the number tasks available in all the partitions by their distance the CP can determine how many iterations it will take for a gatherer to complete a certain set of tasks from the total number that is available in their region. Essentially, the CP keeps monitoring this at every iteration and waits for the number of tasks $\{\kappa_1, \kappa_2, \dots, \kappa_{n_g}\}$ in $\{\varphi_1, \varphi_2, \dots, \varphi_{n_g}\}$ to have an equal distribution before assigning new territories. Therefore, if a set of territory is assigned at t' , the CP can determine, in advance, the time t'' when $\{\kappa_1, \kappa_2, \dots, \kappa_{n_g}\}$ reaches equal distribution.

Theorem 1. Let an n_g -tuple s represent an association of strategies to players called the pure strategy profile, $s = \{s_1, s_2, \dots, s_{n_g}\}$, such that $s_1 \in S_1, s_2 \in S_2, \dots, s_{n_g} \in S_{n_g}$. If each player in the set A_g chooses the assigned territory as its action, i.e., a_i^g chooses φ_i , then the pure-strategy profile obtained in that circumstance is a NE solution of the game, i.e., the strategy profile $s^* = \{s_1^*, s_2^*, \dots, s_{n_g}^*\} = \{\varphi_1, \varphi_2, \dots, \varphi_{n_g}\}$ is a NE.

Proof. Recall the definition of expected utility function $U(\varphi_i, a_i^g) = E(\kappa_i/n_g^{\varphi_i})$. We know that the strategy profile in which each agent is making its best response to other agents is a NE. Essentially, we need to prove $U_i(s_i^*, s_{-i}^*) \geq U_i(s_i, s_{-i}^*)$, $\forall s_i \in S_i$. According to the proposed territorializing algorithm, there is an initial partitioning where the number of partitions equals the number of gatherers. As per the explanation in Remark 1, the number of targets in each partition converges to an equal distribution for the gatherers. When that happens, the κ_i in the utility function associated with φ_i will be equal for all agents. In such a situation, the best action for each agent is not to be in the same partition as another agent. In other words, since the number of targets in all partitions is equal, the more the agents present in a partition, the less the reward they accumulate based on the expected utility function. Therefore, $U_i(s_i^*, s_{-i}^*) \geq U_i(s_i, s_{-i}^*)$, $\forall s_i \in S_i$; hence, s^* is a NE.

Consequently, as the number of tasks in each partition converges to equal distribution, the NE of the formulated game will converge to an equal distribution for gatherers in partitions as well. Henceforth, it can be surmised that it is justifiable for the gatherers to stay within their assigned territory.

6. Simulation Results

To validate the efficacy of the proposed work, simulation results have been presented in this section. First, we compare how the performance of the dynamic territorializing

algorithm varies, as the number of times territorializing is performed is varied. Second, we investigate the effect of varying the number of targets in the field F . Then, we compare the proposed algorithm with an alternative approach, where F has no territories for the gatherers. The comparison is quantized in terms of measure of the mission accomplishment time and the effect on workload $W(t)$. The simulation platform has been developed in MATLAB. All missions have been executed for $n_g = 3$ and $n_h = 6$. The mission environment consists of $e_x \times e_y$ grid of squares, where $e_x = e_y = 100$. The hunters are set to move at a speed twice that of the gatherers, and all agents are assigned a random initial starting position in the field. In addition, all missions have been executed in an interminable mode in which each time a gatherer completes a task, another task is distributed randomly in the environment.

For the alternative approach utilized in Section 4.4, n_h and n_g remain the same. The same algorithms are utilized for the exploration and path-planning of the agents. However, no territories are formed for the gatherers and they have the freedom to collect the nearest available target from any section of the operating environment. Since the presence of a CP is no longer needed, it is assumed that there is an online board where the locations of the targets explored are revealed to the gatherers. In addition, the gatherers are assigned tokens so that when they happen to be equidistant from a target, the gatherer with the lower token number can be assigned with the task.

6.1. Effect of the Frequency of Territorializing on Workload. To analyze and evaluate whether there is any effect on the workload when the number of times territorializing is performed, we perform 1000 simulations for several instances, i.e., $N_s = 1000$. Specifically, we define a parameter frequency, f_{N_s} , which is the number of times territorializing is performed over a given N_s . For each frequency, $N_s = 1000$ is repeated 50 times, and the average of workload response is plotted against iteration as shown in Figure 6. As f_{1000} is set to 2, 3, 4, and 5, it can be deduced from the figure that all the plots follow the same trend, and the workload decreases with respect to the number of iterations. Particularly, when territorializing is performed 5 times over the 1000 simulations, i.e., $f_{n_s} = 5$, the workload converges to a value of 0.1988, as indicated by the black curve on Figure 6. The values of $W(t)$ in the plots for $f_{n_s} = 3$ and $f_{n_s} = 4$ converge to a value of 0.2073, and for $f_{n_s} = 2$, it reaches a value of 0.2477.

6.2. Performance of the Proposed Algorithm with respect to the Density of Targets. In this section, we look at how the performance of the proposed algorithm varies as the density, ρ , of the targets change. Particularly, we set $N_s = 1000$ and repeat the simulations 50 times, and we set ρ to 0.0005, 0.001, 0.0015, and 0.002. Like in the previous section, we plot $W(t)$ against the number of iterations to observe the convergence of the curve. It can be seen from the plot in Figure 7 that the value of workload tends to converge to a lower value as the density of the targets or

FIGURE 5: The available actions for a_2^g in the normal-form game. If the agent chooses φ_2 (territory 2), it means that it has decided to retain the initial territorializing. Otherwise, where $l \neq 2$, the agent has decided to deviate from the original partition.

FIGURE 6: Comparing the performance of the proposed algorithm in terms of the workload, $W(t)$, as the frequency of territorializing varies.

tasks increases. In addition, the number of iterations taken to converge to these values is also decreasing as the density is increasing. From this plot, it is evident that the proposed algorithm is likely to perform better when the density of target in F is higher.

6.3. Effectiveness of Territorializing on Mission Accomplishment Time. To demonstrate the efficacy of territorializing

FIGURE 7: Comparing how the performance of the proposed algorithm varies as the density of the targets in F varies. Performance is measured in terms of the workload $W(t)$.

amongst the gatherers, we compare the mission accomplishment time of the proposed algorithm against the alternative method. Particularly, mission accomplishment time is the total time it takes for the agents to complete a certain number of tasks. The simulation is run for numerous instances, setting different goals for the agents in each instance, where the goal of the agents is to complete a certain number of tasks $|T|$. We set $|T| = \{50, 100, 150, 200\}$ for the two algorithms; for each instance, the simulation is run 50 times so that we have 50 samples for both the algorithms for a given setting. To analyze the data, a paired T -test has been performed. The results are summarized in Table 1. The hypothesis has been carried out in the following manner: $H_0: \mu^{\text{proposed}} - \mu^{\text{alternative}} = D_0$, $H_0: \mu^{\text{proposed}} - \mu^{\text{alternative}} \neq D_0$, $D_0 = 0$, $n_t = 50$, $\text{dof} = 49$, and $\alpha = 0.05$. Here, μ^{proposed} and $\mu^{\text{alternative}}$ are the means of the proposed algorithm and alternative approach, respectively, n_t is the number of simulations, and dof is the degree of freedom. As summarized in Table 1, it can be seen that for all the instances, $p < \alpha$, from which it can be deduced that the null hypothesis $H_0: \mu^{\text{proposed}} - \mu^{\text{alternative}} = D_0$ cannot be retained. In fact, the positive value of t indicates that the alternative approach has a greater average mean time as compared to the proposed algorithm. In addition, to visualize this difference in performance, a boxplot has been plotted in Figure 8. It can be seen from the boxplot that in each of these five settings, the proposed algorithm is able to complete the number of given tasks at a faster rate compared to the alternative approach (the green boxplot lies below the purple box in all five instances). Particularly, for $|T| = \{50, 100, 150, 200, 250\}$, the alternative approach takes 397.4, 715, 1075, 1371, and 1705 iterations, respectively, whereas the proposed algorithm takes 349.9, 631.9, 925.7, 1241, and 1520 iterations, respectively.

TABLE 1: Paired *T*-test.

Γ_c	dof	p	t	95% CI	
				CI _L	CI _U
50	49	$8.15e-8$	6.29	32.3939	62.7661
100	49	$6.47e-13$	9.65	65.78	100.38
150	49	$1.75e-16$	12.22	124.32	173.28
200	49	$9.24e-11$	8.21	97.96	161.48
250	49	$1.34e-13$	10.12	148.39	221.89

FIGURE 8: Investigating the effectiveness of the proposed algorithm against an alternative approach. Each of the five boxplots shows that the proposed algorithm outperforms the alternative method. The means are marked on each of the boxes with a black dot.

FIGURE 9: Comparing the performance of the alternative approach with the proposed algorithm in terms of the measure of workload $W(t)$.

6.4. Validating the Functionality of the Proposed Algorithm against an Alternative Approach. In order to analyze whether there is any effect of territorializing on the workload, the alternative approach and the proposed method are simulated for 1000 iterations. In each iteration, the workload is calculated using (4). For the proposed algorithm, the territorializing is performed every 200 iterations; i.e., it is performed 5 times. For both the approaches, the 1000-iteration simulation is run 50 times, and the average is plotted as shown in Figure 9. It can be observed from the plot that for the alternative approach, the workload tends to converge to a value of 0.33; comparatively, for the proposed algorithm, the workload almost converges to a much lower value of 0.19. The curve for the proposed algorithm also lies below the alternative throughout the entire plot, confirming that the latter approach outperforms the former one.

7. Conclusion

In this work, a dynamic territorializing approach for multi-agent task allocation is proposed for a “hunter and gatherer” scheme developed in the previous work. The proposed algorithm utilizes the center of mass of the targets discovered by the hunters and employs it as the point from which the territories are to be formed for the gatherers. The advantage of using the center of mass is that it lies closer to the region that is more densely populated. A game-theoretic analysis is provided to justify why it is reasonable for the agents to stay within their assigned territory. Specifically, as the number of tasks in each territory converges to equal distribution, the Nash equilibrium of the formulated game will converge to an equal distribution for gatherers in partitions as well. Furthermore, numerical results have been provided to validate the effectiveness of the proposed method. Particularly, the proposed algorithm is compared with an alternative approach in terms of mission accomplishment time and workload. It is seen that the proposed algorithm performs considerably better than the alternative approach where no territories between the agents exist. In addition, the numerical analysis shows the effect of varying two parameters on the workload—the frequency of the performance of territorializing and the density of targets in the operating environment.

Future work will consider the scalability of the algorithm by varying the number of hunters and gatherers in the operating environment. To capture a more practical scenario, we would also like to implement the algorithms in the setting utilized in [43], where the targets are normally distributed at random instead of having a uniform random distribution. Moreover, we also plan to extend the centralized algorithm to a distributed one, in which the presence of a centralized planner will not be necessary.

Data Availability

Readers are encouraged to contact the authors for any data and source code.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Lamar University via internal grants. Readers are encouraged to contact the authors for any data and source code.

References

- [1] K. Cheng and P. Dasgupta, "Multi-agent coalition formation for distributed area coverage: analysis and evaluation," in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, vol. 3, pp. 334–337, Toronto, Canada, August 2010.
- [2] G. Dudek, M. R. Jenkin, E. Milios, and D. Wilkes, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.
- [3] D. Portugal and R. Rocha, "MSP algorithm: multi-robot patrolling based on territory allocation using balanced graph partitioning," in *Proceedings of the 2010 ACM symposium on applied computing*, pp. 1271–1276, Sierre, Switzerland, March 2010.
- [4] A. Jevtic, A. Gutiérrez, D. Andina, and M. Jamshidi, "Distributed bees algorithm for task allocation in swarm of robots," *IEEE Systems Journal*, vol. 6, no. 2, pp. 296–304, 2011.
- [5] T. Theodoridis and H. Hu, "Toward intelligent security robots: a survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1219–1230, 2012.
- [6] B. P. Gerkey and M. J. Mataric, "Multi-robot task allocation: analyzing the complexity and optimality of key architectures," in *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*, vol. 3, pp. 3862–3868, Taipei, Taiwan, September 2003.
- [7] L. Luo, N. Chakraborty, and K. Sycara, "Provably-good distributed algorithm for constrained multi-robot task assignment for grouped tasks," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 19–30, 2014.
- [8] L. Huang, Y. Ding, M. Zhou, Y. Jin, and K. Hao, "Multiple-solution optimization strategy for multirobot task allocation," in *Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Miyazaki, Japan, December 2018.
- [9] M. Dadvar, S. Moazami, H. R. Myler, and H. Zargarzadeh, "Multiagent task allocation in complementary teams: a hunter-and-gatherer approach," *Complexity*, vol. 2020, Article ID 1752571, 15 pages.
- [10] M. Dadvar, S. Moazami, H. R. Myler, and H. Zargarzadeh, "Exploration and coordination of complementary multi-robot teams in a hunter and gatherer scenario," 2019, <https://arxiv.org/abs/1912.07521>.
- [11] R. Murphy, *Disaster Robotics (Intelligent Robotics and Autonomous Agents Series)*, The MIT Press, Cambridge, MA, USA, 2014.
- [12] H. Kitano, S. Tadokoro, I. Noda et al., "Robocup rescue: search and rescue in large-scale disasters as a domain for autonomous agents research," in *Proceedings of the IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028)*, vol. 6, pp. 739–743, Tokyo, Japan, October 1999.
- [13] A. Bechar and C. Vigneault, "Agricultural robots for field operations: concepts and components," *Biosystems Engineering*, vol. 149, pp. 94–111, 2016.
- [14] M. Dunbabin and L. Marques, "Robots for environmental monitoring: significant advancements and applications," *IEEE Robotics & Automation Magazine*, vol. 19, no. 1, pp. 24–39, 2012.
- [15] J. N. K. Liu, M. Wang, and B. Feng, "iBotGuard: an Internet-based intelligent robot security system using invariant face recognition against intruder," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 35, no. 1, pp. 97–105, 2005.
- [16] J. Peterson, W. Li, B. Cesar-Tondreau et al., "Experiments in unmanned aerial vehicle/unmanned ground vehicle radiation search," *Journal of Field Robotics*, vol. 36, no. 4, pp. 818–845, 2019.
- [17] H. Liu, W. Wang, Z. He et al., "The design of air-space integrative calamity information analysis and rescue system," in *Proceedings of the 2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pp. 1997–2001, Shenyang, China, June 2015.
- [18] B. P. Gerkey and M. J. Mataric, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.
- [19] K. Lerman, C. Jones, A. Galstyan, and M. J. Mataric, "Analysis of dynamic task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 225–241, 2006.
- [20] S. Liemhetcharat and M. Veloso, "Weighted synergy graphs for effective team formation with heterogeneous ad hoc agents," *Artificial Intelligence*, vol. 208, pp. 41–65, 2014.
- [21] S. Sariel and T. R. Balch, "Efficient bids on task allocation for multi-robot exploration," in *Proceedings of the FLAIRS Conference*, pp. 116–121, Melbourne Beach, FL, USA, May 2006.
- [22] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas, "Distributed multi-robot task assignment and formation control," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pp. 128–133, Pasadena, CA, USA, May 2008.
- [23] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: a survey and analysis," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1257–1270, 2006.
- [24] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli, "The sensor-based random graph method for cooperative robot exploration," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, pp. 163–175, 2009.
- [25] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [26] M. Vaidis and M. J.-D. Otis, "Toward a robot swarm protecting a group of migrants," *Intelligent Service Robotics*, vol. 13, no. 2, pp. 1–16, 2020.
- [27] L. Lv, S. Zhang, D. Ding, and Y. Wang, "Path planning via an improved DQN-based learning policy," *IEEE Access*, vol. 7, pp. 67319–67330, 2019.
- [28] M. Geng, Y. Li, B. Ding, and H. Wang, "Deep learning-based cooperative trail following for multi-robot system," in *Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Rio de Janeiro, Brazil, July 2018.
- [29] J. C. G. Higuera and G. Dudek, "Fair subdivision of multi-robot tasks," in *Proceedings of the 2013 IEEE International*

- Conference on Robotics and Automation*, pp. 3014–3019, Karlsruhe, Germany, May 2013.
- [30] N. Fung, J. Rogers, C. Nieto, H. I. Christensen, S. Kemna, and G. Sukhatme, “Coordinating multi-robot systems through environment partitioning for adaptive informative sampling,” in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pp. 3231–3237, Montreal, Canada, May 2019.
 - [31] J. G. Carlsson, E. Carlsson, and R. Devulapalli, “Balancing workloads of service vehicles over a geographic territory,” in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 209–216, Tokyo, Japan, November 2013.
 - [32] D. Drenjanac, S. D. K. Tomic, L. Klausner, and E. Kühn, “Harnessing coherence of area decomposition and semantic shared spaces for task allocation in a robotic fleet,” *Information Processing in Agriculture*, vol. 1, no. 1, pp. 23–33, 2014.
 - [33] B. Wiandt, V. Simon, and A. Kokuti, “Self-organized graph partitioning approach for multi-agent patrolling in generic graphs,” in *Proceedings of the IEEE EUROCON 2017-17th International Conference on Smart Technologies*, pp. 605–610, Ohrid, Macedonia, July 2017.
 - [34] S. Hoshino and K. Takahashi, “Dynamic partitioning strategies for multi-robot patrolling systems,” *Journal of Robotics and Mechatronics*, vol. 31, no. 4, pp. 535–545, 2019.
 - [35] S. Vourchteang and T. Sugawara, “Area partitioning method with learning of dirty areas and obstacles in environments for cooperative sweeping robots,” in *Proceedings of the 2015 IIAI 4th International Congress on Advanced Applied Informatics*, pp. 523–529, Okayama, Japan, July 2015.
 - [36] H. Ortega-Arranz, D. R. Llanos, and A. Gonzalez-Escribano, “The shortest-path problem: analysis and comparison of methods,” *Synthesis Lectures on Theoretical Computer Science*, vol. 1, no. 1, pp. 1–87, 2014.
 - [37] Y. Chen, J. K. Lai, D. C. Parkes, and A. D. Procaccia, “Truth, justice, and cake cutting,” *Games and Economic Behavior*, vol. 77, no. 1, pp. 284–297, 2013.
 - [38] M. Dall’Aglio, “The Dubins–Spanier optimization problem in fair division theory,” *Journal of Computational and Applied Mathematics*, vol. 130, no. 1–2, pp. 17–40, 2001.
 - [39] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA’97. “Towards New Computational Principles for Robotics and Automation”*, vol. 97, p. 146, Monterey, CA, USA, July 1997.
 - [40] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Pearson Education Limited, Malaysia, 2016.
 - [41] A. Cossari, J. C. Ho, G. Paletta, and A. J. Ruiz-Torres, “A new heuristic for workload balancing on identical parallel machines and a statistical perspective on the workload balancing criteria,” *Computers & Operations Research*, vol. 39, no. 7, pp. 1382–1393, 2012.
 - [42] H. González-Vélez and M. Cole, “Adaptive statistical scheduling of divisible workloads in heterogeneous systems,” *Journal of Scheduling*, vol. 13, no. 4, pp. 427–441, 2010.
 - [43] I. Tkach and Y. Edan, “Extended examples of single-layer multi-sensor systems,” in *Distributed Heterogeneous Multi Sensor Task Allocation Systems*, pp. 49–79, Springer, Berlin, Germany, 2020.