IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 16 July 2024, accepted 23 July 2024, date of publication 29 July 2024, date of current version 25 September 2024.
Digital Object Identifier 10.1109/ACCESS.2024.3434645

RESEARCH ARTICLE

CoLoSSI: Multi-Robot Task Allocation in Spatially-Distributed and Communication Restricted Environments

ISHAQ ANSARI[1], ABUBAKR MOHAMMED[2], YAQOOB ANSARI[1], MOHAMMED YUSUF ANSARI[3], SAQUIB RAZAK[4], AND EDUARDO FEO FLUSHING[2]
[1]Department of Electrical and Computer Engineering, Texas A&M University at Qatar, Ar-Rayyan, Qatar
[2]Department of Computer Science, Carnegie Mellon University in Qatar, Ar-Rayyan, Qatar
[3]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA
[4]Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109, USA
Corresponding author: Ishaq Ansari (m.ansari1@qatar.tamu.edu)

ABSTRACT In our research, we address the problem of coordination and planning in heterogeneous multi-robot systems for missions that consist of spatially localized tasks. Conventionally, this problem has been framed as a task allocation problem that maps tasks to robots. However, all previous work assumes that tasks are atomic procedures. In this work, we relax this assumption and adopt a non-atomic model of tasks that enables robots to accomplish mission tasks incrementally over disjoint periods, precisely to account for the possibility of having a task serviced by numerous individual contributions over time. We propose a cooperative, load-balancing task allocation and scheduling algorithm based on sequential single-item auctions (CoLoSSI) that explicitly considers the non-atomicity of tasks, promotes synergies between agents, and enables cooperation while maintaining computational tractability. We also propose a fully distributed variant of CoLoSSI that tackles sparse, communication-restricted scenarios. Computational and simulation results confirm the efficacy of the proposed approaches for generating good-quality mission plans with low computational effort.

INDEX TERMS Cooperative robotics, distributed algorithms, load balancing, multi-robot systems, non-atomic task model, search and rescue robotics, sequential single-item auctions, task allocation.

I. INTRODUCTION

Mobile multi-robot teams have emerged as versatile, adaptable, and robust solutions for a variety of real-world applications, as evidenced by their growing prominence in areas such as search and rescue, environmental monitoring, and area patrolling [1]. These applications often involve time-extended missions over large areas, where the efficient coordination of heterogeneous teams (robots with diverse sensory-motor capabilities) is essential. Such teams, consisting of robots with different capabilities, must work together efficiently to avoid conflicts, minimize redundant efforts, and effectively achieve their objectives.

In large-scale operations, the efficiency and success of multi-robot teams rely heavily on their ability to handle two primary challenges: optimal task distribution and maintaining effective communication in potentially restricted environments. These operations typically demand extensive area coverage, prolonged mission duration, and the management of complex task distributions. The core challenge lies in the distribution of tasks among the team members to ensure the overall mission is completed in the shortest time possible, referred to as minimizing the makespan (i.e., time needed for mission completion). This need for efficient task allocation is universal in connectivity-restricted

The associate editor coordinating the review of this manuscript and approving it for publication was Faissal El Bouanani.

or unrestricted environments. However, in communication-restricted settings, there is a heightened risk of robots losing contact with their teammates by moving out of the communication zone, leading to redundant task allocation and unnecessary navigation. Consequently, this results in a significant increase in the overall makespan of the mission. As a result, maintaining connectivity within the team and forming larger, more cohesive groups becomes crucial for productive and efficient mission execution.

To address these challenges effectively, it is crucial to understand and optimize the fundamental problem of Multi-Robot Task Allocation (MRTA) in these complex environments. MRTA is a fundamental problem in multi-robot systems, focusing on efficiently assigning tasks to robots in a team to optimize overall performance. MRTA algorithms aim to distribute tasks among robots while considering various factors such as robot capabilities, task requirements, and environmental constraints.

This paper explores the coordination of mobile multi-robot teams equipped with range-limited radio interfaces for extended missions across expansive regions. Emphasizing teams of autonomous robots with diverse capabilities, we construct a mobile multi-hop wireless network essential for intra-team communication, particularly where dedicated infrastructure is absent and disconnection risks are high. Our focus is on optimizing makespan for spatially distributed, non-atomic tasks through a task allocation and scheduling model that assigns tasks considering limited communication, where robots rely on partial information. Existing coordination schemes, both implicit [2] and explicit [3], usually depend on robust communication channels, yet falter in large-scale operations like post-disaster scenarios with compromised infrastructure. Addressing this, we propose the CoLoSSI algorithm, a cooperative, load-balancing task allocation and scheduling method grounded in the concept of sequential single-item auctions. CoLoSSI advances the traditional auction model [4] by providing high-quality solutions efficiently, crucial for teamwork and reduced computational demands [5]. Our approach not only manages the distribution of tasks effectively but also maintains operational effectiveness, ensuring that the multi-robot teams' coordination is resilient to the communication challenges inherent in large-scale, unpredictable environments.

The algorithm is a variant of the classical sequential single-item auction approach [4]; therefore, it inherits its simplicity and low computational cost and also provides good quality solutions [5].

We implement CoLoSSI in a closed-loop, fully distributed manner, with periodic recomputation of mission plans. Robots within the communication range collaborate to develop joint plans using CoLoSSI. In this manner, they continuously update each other over the wireless network as they execute their tasks. This iterative planning process adapts to environmental uncertainties and unexpected developments. This is a crucial feature in dynamic scenarios that mimic real-world applications such as environmental monitoring or post-disaster response.

Moreover, we identify and address specific challenges in communication-restricted environments by introducing extensions to the CoLoSSI algorithm. These extensions focus on generating mission plans that optimize wireless networking conditions by considering factors like connectivity, obstacle presence, and spatiotemporal dynamics. The objective is to increase the team's performance by improving network connectivity. Our connectivity-aware variant, Co-CoLoSSI, integrates these considerations into the auction-based algorithm, prioritizing communication efficiency during plan execution.

Our evaluation demonstrates that CoLoSSI yields high-quality mission plans with reduced computational requirements compared to optimal solutions obtained using the CPLEX mathematical solver. Additionally, simulations of Co-CoLoSSI in communication-restricted environments show that enhanced communication capabilities significantly improve decision-making quality. Co-CoLoSSI proves particularly useful when there is no reliable communication infrastructure or when communication patterns are unpredictable. Its main advantages are its computational efficiency, dynamic adaptability, and practical use in real-world applications.

Distinguishing our work from prior research, we address a critical aspect of MRTA that has been relatively unexplored in the studies mentioned above: the impact of network connectivity on team performance. Unlike approaches that impose specific connectivity requirements or rely on dedicated relay agents [6], we introduce computationally efficient strategies to enhance communication and team coordination. Recent studies by [3], [7], and [8] have demonstrated that the performance of auction algorithms can significantly degrade under varying communication qualities within a team. Our work aims to mitigate this performance degradation through innovative strategies that improve network connectivity throughout the mission.

These strategies range from a straightforward yet impactful technique based on mission replanning intervals to more sophisticated approaches centered around managing spatiotemporal relations within the team. Importantly, as demonstrated with our CoLoSSI approach, these strategies can seamlessly integrate into any baseline auction-based decentralized method. This adaptability ensures that our contributions can be widely applied across various MRTA scenarios, further advancing the field's capabilities in real-world applications where communication constraints are a significant concern.

This paper introduces several novel contributions to multi-robot systems, particularly in enhancing communication and coordination within multi-robot teams operating in communication-restricted environments. The novelties introduced in this work are as follows:

- Introduction and validation of the **CoLoSSI algorithm** with a significantly larger dataset, showcasing its effectiveness in coordinating non-atomic tasks among multi-robot teams.
- Proposal of various strategies aimed at improving inter-agent communication and facilitating the formation of larger collectives, which help decrease the overall mission makespan.
- Extensive evaluation of strategies to enhance communication within robot teams, including exploring different replanning intervals and meetup tactics to optimize team coordination.
- Enhancement of the **CoLoSSI algorithm** with the **Co-CoLoSSI algorithm**, incorporating rigorously tested strategies to improve performance in communication-restricted settings.

Following these advancements, it is pertinent to mention that a preliminary version of the CoLoSSI algorithm was presented in previous work [9]. The current paper significantly extends upon this foundation by conducting experiments on a larger number of problem instances and those of higher dimensionality. An in-depth explanation of the CoLoSSI algorithm and a thorough evaluation of the model in various communication-restricted scenarios are provided. This work also includes a comparative analysis with the state-of-the-art sequential single-item auctioning algorithm, thereby demonstrating the adaptability and robustness of our proposed algorithms under dynamic conditions.

Following this introduction, Section II reviews related work, setting the context for our research. Section III presents a model for cooperative mission planning with heterogeneous teams, elaborating on the spatial task allocation and scheduling problem in heterogeneous mobile multi-robot teams (STASP-HMR). In Section IV, we introduce CoLoSSI, our cooperative and load-balancing task allocation and scheduling solution, utilizing sequential single-item auctions. Section V discusses distributed planning in communication-constrained environments, highlighting the challenges and necessities of such environments. Section VI introduces Co-CoLoSSI, a connectivity-aware variant of our algorithm tailored for communication-restricted environments. Section VII presents the empirical setup for evaluating the proposed algorithms. The results and discussion about CoLoSSI and Co-CoLoSSI, demonstrating their effectiveness and efficiency, are detailed in Section VIII. Finally, Section IX concludes the paper, summarizing our findings and contributions.

To support reproducibility and facilitate further research, the code used for this study is available in a GitHub repository: https://github.com/IshaqAnsari2001/CoLoSSI/tree/main.

II. RELATED WORK

In the context of coordination in the robotics domain, MRTA problems map robots to tasks with different assumptions, restrictions, and domain-dependent issues. In this section, we first discuss the MRTA literature in scenarios where network communication is taken for granted. In these works, the primary focus is on issues such as computational scalability, the nature of the tasks, and the task methodology utilized to make the decisions. Next, we discuss the approaches with a particular focus on the communication aspects of multi-robot missions. Table 1 summarizes the most related approaches to our work.

A. MULTI-ROBOT TASK ALLOCATION

A foundational classification of MRTA works was introduced in [10], establishing connections with mathematical optimization models like vehicle routing, assignment, and set problems. Building upon this work, [11] expanded the taxonomy to encompass interconnected utilities and constraints, while [12] categorized constraints on the schedules of the robots.

These classifications have enabled the formalization of MRTA scenarios and brought to light the NP-hard nature of MRTA problems. Consequently, devising computationally feasible solution approaches with performance guarantees proves inherently challenging in realistically complex scenarios. Several MRTA studies have proposed distributed approaches to address scalability concerns and cooperative game theory [19]. Among these, market-based methods [23], [24] leverage negotiation techniques such as auctions and have demonstrated success across various domains [14], [25]. Recently, deep neural networks have revolutionized several domains, including medical imaging [26], [27], [28], [29], [30], signal processing [31], [32], [33], [34], drug discovery [35], [36], [37], and machine vision [38]. Subsequently, graph neural networks have also been employed in literature for decentralized multi-robot goal assignment [18].

In this work, we propose a market-based method for the *spatial task allocation and scheduling problem in heterogeneous multi-robot teams*, or STASP-HMR in short [39]. According to the initial categorization of [10], the STASP-HMR covers both *single-task robots* (ST) and *multi-task robots* (MT), and considers *single-robot tasks* (SR), for a *time-extended assignment* (TA). According to the extended taxonomies, our problem also fits the class of MRTA with cross-schedule dependencies [11] and MRTA with precedence constraints [12].

It is noteworthy that all previous market-based MRTA approaches, to the best of our knowledge, have been formulated under the restrictive assumption that tasks are atomic procedures [5], [13], [14], [17], [20], [24], [40], necessitating uninterrupted agent effort for task completion. Market-based approaches, particularly auction-based approaches, have recently been shown to be effective in scenarios atomic tasks [14], [41], such as pick-and-deliver [42], [43] sensing tasks [15], and tasks that are announced stochastically [16]. These approaches, however, cannot be directly used to tackle the STASP-HMR without compromising their computational efficiency: accounting for non-atomic tasks would require

TABLE 1. Summary of related MRTA approaches.

Research Work	Task Model	Network Assumptions / Constraints	Method	Mission Objective
[5]	Atomic	Restricted comm. range	SSI	Minimize energy usage, completion time
CBBA [13]	Atomic	None	Consensus-based auctions	Maximize generic global reward
TeSSI [14]	Atomic, Temporal Dependencies	None	SSI + Temporal Netwoks	Makespan
[15]	Atomic tasks	Infrastructure	Periodic task auctions	Minimize energy
[16]	Atomic, stochastic start time and location	None	SSI auctions	Minimize waiting time
MM-CBGA [17]	Atomic, Composite	None	Consensus-based auctions	Maximize generic global reward
DGNN-GA [18]	Atomic	Restricted comm. range	Decentralized Graph Neural Network	Minimize global generic cost
[19]	Atomic	None	Game-theory based algorithm	Multi-objective
DEMiR-CF [20]	Coalition tasks with precedence constraints	None	Generic framework	Constraint-free, minimum cost allocation
CA-CBBA [21]	Atomic	Limited bandwidth	Consensus-based auctions	Maximize generic global reward
CBBA-Relays [6]	Atomic	Restricted comm. range, dedicated relays	Consensus-based auctions	Maximize generic global reward, ensure network connectivity
[22]	Atomic	Restricted comm. range	Distributed Optimization	Minimize global generic cost

decomposing each task into a bundle of atomic tasks for the agents to bid on, causing an immense increase in the number of tasks available for bidding, and a significant impact on the tractability of the computation. Our approach explicitly considers the non-atomic nature of tasks and integrates fractional task bidding into the auctioning process. As a result, it enables the calculation of - potentially more effective - solutions where tasks are completed incrementally over separate time intervals, allowing different agents to contribute effort during these periods.

B. MULTI-ROBOT TASK ALLOCATION IN COMMUNICATION-RESTRICTED ENVIRONMENTS

The majority of research in MRTA assumes the presence of a reliable communication network [8], [24]. Other works have addressed communication challenges in fully connected topologies. For instance, in [21], authors propose a variation of the Consensus-Based Bundle Algorithm (CBBA) [13] that takes into account bandwidth limitations and message collisions. Assuming a fully connected network, they propose a modified medium-access layer protocol that improves the utilization of the shared medium based on the nature of the messages sent by the task allocation algorithm. In [15], an information sensing mission employing UAVs takes advantage of stable wireless connections provided by prevalent urban networks like 5G and 6G. Their approach aims to enhance data exchange efficiency through periodic task auctions while managing the transmission costs associated with sensing tasks. Only a limited number of studies have explored situations in which the network topology undergoes continuous changes, affecting the potential for communication among agents [3], [6]. It is noteworthy that the majority of works in this domain predominantly restrict their treatment of communication issues to evaluation without actively seeking to enhance the network topology to address these challenges, e.g., [5].

Most MRTA works addressing challenges in communication-restricted environments generally adopt one of two distinct approaches. The first prevalent strategy is the assignment of dedicated network provisioning tasks to a subset of agents, as observed in various studies such as [44] and [45]. A second approach involves imposing rigid communication constraints, such as maintaining close proximity among robots. Examples include establishing permanent communication paths between a base station and a group of robots [46] and ensuring global connectivity among the robots [47].

Both approaches to addressing communication challenges in restricted environments come with distinct drawbacks. The strategy involving dedicated network provisioning tasks can lead to resource inefficiencies, particularly when the number of available robots is limited [48]. On the other hand, approaches imposing rigid communication constraints, such as maintaining close proximity at all times, sacrifice adaptability. These constraints limit flexibility in mission execution, impacting their overall effectiveness in dynamic and complex environments. Balancing these drawbacks is crucial when devising strategies for effective communication in robotic scenarios.

C. GLOBAL PERSPECTIVES AND RECENT ADVANCEMENTS IN MRTA

The field of MRTA has seen significant advancements globally, with researchers from various countries contributing to its evolution and demonstrating its impact across different sectors. In China, [49] developed a large-scale 3D printing system using multiple collaborative robots. Their printer task-optimized scheduling algorithm and robot interference avoidance strategy resulted in an efficiency improvement exceeding 73% compared to general printing methods, showcasing the potential of MRTA in advanced manufacturing. In Japan, [50] proposed hybrid evolutionary algorithms to optimize layouts for multi-robot cellular manufacturing systems. Their work, which included cooperative tasks among robots, evaluated layout area, operation time, and robot manipulability as key design criteria. The study found that a hybrid genetic algorithm with particle swarm optimization (GA+PSO) performed best, demonstrating the effectiveness of MRTA in complex industrial scenarios. Research from Korea by [51] focused on agricultural applications, developing a multi-robot tractor system for fieldwork. Their system, which could form various spatial patterns (I, V, or W), showed impressive efficiency improvements, particularly in large fields. For instance, with seven robots using the W-pattern, efficiency reached 84.9% for fields 500 meters in length, highlighting MRTA's potential to revolutionize agricultural practices.

These studies underscore the impact of MRTA innovations on technological progress across different sectors and geographical regions. The choice of market-based methods for MRTA, as employed in our study and several others [5], [13], [17], is based on their proven effectiveness and scalability. Market-based approaches, particularly auction-based methods, have shown promising results in various scenarios. For example, [15] applied these techniques to sensing tasks, while [16] explored their effectiveness in scenarios where tasks are announced stochastically. Recent years have witnessed a surge in MRTA research, with notable contributions expanding the field's scope and capabilities. [18] employed graph neural networks for decentralized multi-robot goal assignment, demonstrating the potential of advanced machine learning techniques in MRTA. Reference [19] explored game-theory-based algorithms for multi-objective MRTA, while [22] focused on distributed optimization techniques for scenarios with restricted communication range.

From a global perspective, MRTA research has become increasingly collaborative and diverse. This is evident from the variety of approaches and applications seen in recent literature, spanning from manufacturing and agriculture to complex optimization problems. Our work builds upon these foundations, addressing the specific challenges of the STASP-HMR while contributing to the broader landscape of MRTA innovation.

III. A MODEL FOR COOPERATIVE MISSION PLANNING WITH HETEROGENEOUS TEAMS

In this section, we briefly describe the *spatial task allocation and scheduling problem in heterogeneous mobile multi-robot teams*, or STASP-HMR in short. The STASP-HMR has been formally introduced in our previous work [39]. For completeness' sake, we include some definitions here.

A. DEFINITIONS AND TERMINOLOGY

We assume the mission has been decomposed into a set $\mathcal{T}$ of spatially distributed, location-dependent tasks. A task corresponds to the execution of a particular action at a specific location (or portion) of the environment. Tasks can be *non-atomic*, implying that any task can be accomplished incrementally over separate time periods when different agents devote a certain effort. Tasks in $\mathcal{T}$ are not necessarily spatially disjoint, and we assume there are no temporal dependencies related to the tasks (e.g., execution time-windows, or deadlines).

A set of resources A (*mobile agents*) is available for the mission. Each agent can perform different tasks to some extent (including the case where an agent is not suited to deal with a task) and with an *agent-specific time efficiency*.

The goal of *mission planning* consists in completing all the tasks while minimizing the completion time (e.g., makespan). To this end, three sub-problems must be solved:

- *Agent task allocation*: assign all tasks to specific agents within the team, based on their current status and their efficiency in dealing with the tasks;
- *Task routing:* for each agent, define the sequence for dealing with the assigned tasks, and, therefore, for moving from one spatial location to another;
- *Time scheduling:* appoint the duration of the services provided to each one of the selected tasks by each agent.

Note that minimizing the makespan requires these sub-problems to be jointly solved.

The diversity in the agent team A is captured through *task efficiency models* that, given $a \in A$ and $i \in \mathcal{T}$, define the efficiency (i.e., amount of work accomplished over time) with which agent a services task i. The intuition is that any progress on the completion of a task depends on the time spent on it, in particular, we consider linear task efficiency relations. For a given task i, we say that agent a is more efficient than agent b if, devoting the same amount of time to task i, a would complete more of i's workload.

We consider time uniformly discretized into a sequence of intervals of length $\Delta_{\mathcal{T}}$. Henceforth, all decisions concerning time are *discrete*: an agent can only spend an integral number of time steps servicing a task. We assume that task efficiency models are known for each agent in A, and remain constant during a mission. Efficiency models are represented by *performance functions* $\varphi_k(i)$ that precisely indicate the decrease in workload of task $i \in \mathcal{T}$ when agent $k \in A$ executes it for one $\Delta_{\mathcal{T}}$ time interval. We also assume that the efficiency models are additive: when agents $A' \subseteq A$ work simultaneously on the same task i, the decrease in workload of i is equal to $\sum_{k\in A'} \varphi_k(i)$. Additive and linear efficiency models are computationally simple and effectively capture the nature of various robotic tasks such as search and coverage [48].

Finally, the *completion map* $C : \mathcal{T} \mapsto [0, 1]$ is used as a mean of expressing the fractional *residual workload* of each task. This concept is particularly relevant in staged planning scenarios to identify the *current completion level* of tasks whose workload has been partially addressed in previous planning stages. For instance, a value of $C(i) = 0$ indicates that task i has already been completed in the past. Therefore no further effort from the agents is required. If an agent attempts to further deal with a completed task, it will amount to a waste of time and resources. Servicing $p \cdot 100$ percent of the workload of task i decreases its required completion $C(i)$ by p.

B. PROBLEM FORMULATION

Based on the above concepts and assumptions, the *spatial task allocation and scheduling problems in heterogeneous multi-robot teams*, or STASP-HMR, in short, is formally stated as follows. Given a set A of heterogeneous agents, characterized by their task performance functions φ, a set of assignable tasks $\mathcal{T}$, a set of initially accessible tasks, the STASP-HMR consists in determining a mission plan – joint plans for the activities of the agents in the environment – that completes all tasks and minimizes the makespan.

A solution to the STASP-HMR is a mission plan that is represented by a set of tasks a_k; task routes p_k, and schedules s_k for each agent k in team A. We denote a mission plan as $\mathcal{P} = \{\mathcal{P}_k \mid k \in A\}$, where $\mathcal{P}_k = < a_k,\ p_k,\ s_k >$ denotes the plan (i.e., assigned tasks, route, and schedule) corresponding to agent k. The sets $a_k \subseteq \mathcal{T}$ indicate the tasks that are assigned to agent k, with $\bigcup_{k \in A} a_k = \mathcal{T}$. Each route p_k is the sequencing of tasks a_k. Schedules $s_k a_k \mapsto \mathbb{N}$ are time assignments that define the total time each of the selected tasks will receive. All routes p_k must start with a task that belongs to the set $\mathcal{T}_0$ of initially accessible tasks (e.g., the locations of the mission's control centers).

The STASP-HMR can be formalized as a Mixed-Integer Linear Program (MILP). Interested readers may find more details on the MILP formulation in [39].

IV. COLOSSI: COOPERATIVE AND LOAD-BALANCING TASK ALLOCATION AND SCHEDULING BASED USING SEQUENTIAL SINGLE-ITEM AUCTIONS

CoLoSSI [9] implements sequential single-item auctions. The algorithm randomly selects one agent who plays the auctioneer role while the other agents are the bidders. In practice, the same agent can play the auctioneer and bidder simultaneously, or the auctioneer could be a central entity. Although it can be a weak point, the auctioneer makes communication among robots easier and keeps track of all individual plans and the completion level of tasks. Additionally, auction methods distribute the task of determining robot bids, as explained below, enabling the auctioning process to continue even if a robot leaves the mission or fails to communicate.

When the auction process begins (Algorithm 1), the auctioneer broadcasts all available tasks and the completion map to all participating agents. Then, the initial phase of CoLoSSI, named *Initial Task Allocation and Scheduling (ITAS)*, occurs in a distributed manner. Each agent independently computes the bid value for each task (using the function *GenerateWeightedBid*) and submits its minimum bid to the auctioneer (using the function *SendBid*). The auctioneer then assigns the task with the minimum bid to the corresponding agent and communicates its decision to the agents. Agents use the function *ReceiveWinner* to learn about the round winner. If multiple equivalent minimum bids are submitted, ties are broken randomly to determine the round winner.

When an agent is assigned a task, it calculates the most efficient route to reach it and includes it in its plan using the function *UpdatePlan*. This step is necessary because tasks are located in different areas, and the agent may need to go through other task locations to get to the assigned task. Once the optimal route is determined, all tasks are added to the end of the plan, with minimal effort assigned to them. The updated plan is communicated to the auctioneer, using the function *Notify*, who distributes the updated state of the mission to all other agents (function *Consensus*).

After each round, at least one task is assigned to an agent, and After $|\mathcal{T}|$ rounds, all the tasks have been assigned to at least one agent and are scheduled for completion. During the auction process, the auctioneer maintains a copy of the completion map to track the completion of the tasks as the rounds are performed. The auctioneer also distributes this knowledge of the completion map to all the agents participating in the auction after each round using the function *UpdateCompletion*.

During the auctions, the agents determine the schedule to allow them to perform the tasks. They allocate the maximal time to complete the residual workload for the won tasks. As mentioned above, agents perform minimal work for tasks that belong to the path to a task that has been won. It is important to note that in some applications, this minimal amount of work could still be helpful for the mission. For instance, in search and rescue missions, the agents could keep searching for the target while traveling to the location of their assigned search tasks.

The bidding algorithm of CoLoSSI takes into account multiple factors. Let C^j be the completion map at the start of the j-th auction round. Let a'_k be the set of tasks awarded to agent k so far ($a'_k \subseteq a_k$). The bid that agent $k \in A$ computes for task $i \in \mathcal{T}$ is a linear combination of the following factors.

- Expected time required from the agent to complete the residual workload of the task: $C^j(i)/\varphi_k(i)$
- The number of tasks the agent has been awarded to the agent: $|a'_k|$.
- The time required to reach the task by traversing through other tasks (i.e., number of traversal tasks required), denoted as $dist(p_k, i)$.
- The duration of the agent's current plan: $\sum_{i \in a_k} s_k(i)$.

$$bid(i) = k_1 C^j(i)/\varphi_k(i) + k_2 |a'_k| \\ + k_3 dist(p_k, i) + k_4 \sum\nolimits_{i \in a_k} s_k(i)$$

where $k1$, $k2$, $k3$ and $k4 \in \mathbb{R}$ represent the weights of each factor in the bidding function (the higher the value, the more

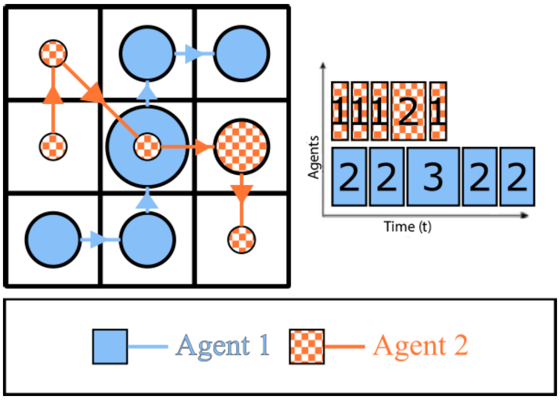

FIGURE 1. This is an illustration of a potential allocation of tasks between two agents. The grid on the left shows which tasks have been allocated to which agent and the graph on the right indicates the number of time units each agent will spend on each task.

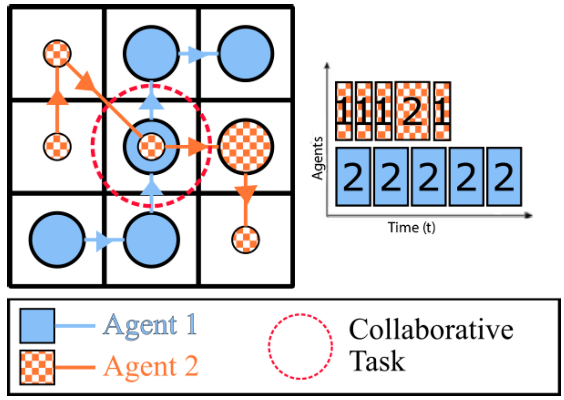

FIGURE 2. After the initial makespan generation, it is still possible that tasks are allocated more time than needed due to future agent traversals. These tasks (e.g. task circled in red) are potential opportunities for collaboration between agents.

importance is given to the factor). Although all the factors in the bidding function play a crucial role in determining each task's winner, some factors should have more influence on the bidding than others. In our experiments, the order of priority of these factors is as follows:

$$k_1 \geq k_2 \geq k_3 \geq k_4 \tag{1}$$

Algorithm 1 ITAS on Agent k

Input: $\mathcal{T}, A, C, \varphi, \mathcal{P}$
Output: Mission plan $\mathcal{P}$
$C^{\mathcal{P}} \leftarrow C$
while not $\mathit{MissionComplete}(C^{\mathcal{P}})$ **do**
  for $i \in \mathcal{T}$ **do**
    $b_{ik} \leftarrow \mathit{GenerateWeightedBid}(i, k)$
  end for
  $t \leftarrow \arg\min_{i \in \mathcal{T}} b_{ik}$
  $\mathit{SendBid}\ ((t, b_{tk}))$
  $(r, t) \leftarrow \mathit{ReceiveWinner}()$
  if $r == k$ **then**
    $\mathcal{P}_k \leftarrow \mathit{UpdatePlan}(t, k)$
    $\mathit{Notify}(\mathcal{P}_k)$
  else
    $\mathcal{P} \leftarrow \mathit{Consensus}()$
  end if
  $C^{\mathcal{P}} \leftarrow \mathit{UpdateCompletion}(\mathcal{P}, \varphi)$
end while

A. IMPROVING AGENTS' SCHEDULES BY DISCOVERING COOPERATIVE ACTIONS

Once the sequential auctions are completed, the resulting plans can have overlapping actions that could be leveraged to boost cooperation and synergies among the agents. For instance, consider the allocation represented by Fig. 1. In this case, agent 1 won the task located in the center of the grid during the bidding. Later, agent 2 won a task located in the bottom-right of the grid that involved a traversal through the central task. During that traversal, the second agent spends minimal time on the central task that could count toward completion. This means that the first agent's preliminary time commitment to the central task did not account for the work the second agent could do over that task during the traversal.

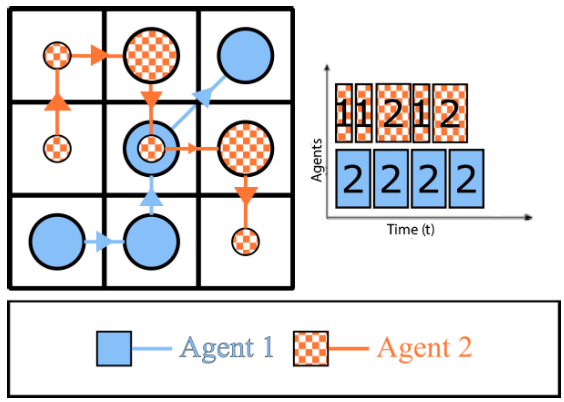

FIGURE 3. Example of final mission plan after workload balancing. Note the redistribution of the tasks and time units among the agents.

This issue is rectified during the second stage of CoLoSSI, named *Iterative Collaborative Refinement* (ICR). During this stage, the agent with the largest makespan iterates over its schedule to find tasks that were assigned more time units than necessary. The agent uses the auctioneer's knowledge of the completion map and the plans of the other agents to determine which tasks are already being contributed to by other agents. From these tasks, if the agent can reduce their time commitment without hampering the completion of the task, then the extra time units are removed from the agent's schedule, reducing their overall makespan. It is important to note that if the removal of time units will result in the task being left incomplete, then no changes are made to the agent's time commitment to the task. The process repeats until the maximum makespan remains unchanged. An example of this is shown in Fig. 2.

Algorithm 2 ICR on Agent k

Input: $\mathcal{T}, A, C, \varphi, \mathcal{P}$
Output: Updated mission plan $\mathcal{P}$

$T, r \leftarrow MaximumMakespanAgent(\mathcal{P})$
$T_{prev} \leftarrow \infty$
$C^{\mathcal{P}} \leftarrow UpdateCompletion(\mathcal{P}, \varphi)$
while $T < T_{prev}$ **do**
    $T_{prev} = T$
    if $r == k$ **then**
        $\mathcal{P}_k \leftarrow RefinePlan(k, C^{\mathcal{P}}, \mathcal{P})$
        $Notify(\mathcal{P}_k)$
    else
        $\mathcal{P} \leftarrow Consensus()$
    end if
    $T, r \leftarrow MaximumMakespanAgent(\mathcal{P})$
    $C^{\mathcal{P}} \leftarrow UpdateCompletion(\mathcal{P}, \varphi)$
end while

Algorithm 3 MBR on Agent k

Input: $\mathcal{T}, A, C, \varphi, \mathcal{P}$
Output: Updated mission plan $\mathcal{P}$

$T, r \leftarrow MaximumMakespanAgent(\mathcal{P})$
$T_{prev} \leftarrow \infty$
$C^{\mathcal{P}} \leftarrow UpdateCompletion(\mathcal{P}, \varphi)$
while $T < T_{prev}$ **do**
    $T_{prev} = T$
    if $r == k$ **then**
        $\mathcal{P} \leftarrow BalancePlan(k, A, C, \varphi, \mathcal{P})$
        $Notify(\mathcal{P})$
    else
        $\mathcal{P} \leftarrow Consensus()$
    end if
    $C^{\mathcal{P}} \leftarrow UpdateCompletion(\mathcal{P}, \varphi)$
    $T, r \leftarrow MaximumMakespanAgent(\mathcal{P})$
end while

B. ACHIEVING LOAD-BALANCING THROUGH TASK REDISTRIBUTION

While the previous procedure considers the work done by other agents due to traversals to promote cooperation and reduce the overall makespan, the resulting plans could still need to be balanced regarding the workload. As a result, some agents may end up with considerably lower makespans than the maximum makespan, thus underutilized.

The third stage of CoLoSSI, named *Makespan balancing and refinement* (MBR), is used to re-balance the overall workload among the agents to reduce the current mission makespan.

For example, note that Agent 1 in Fig. 2 has the maximal makespan of 10 time units, while Agent 2 has a makespan of 6 time units. Thus MBR increases Agent 2's load while reducing Agent 1's to achieve a lower maximal makespan. This example is shown in Fig. 3.

The load-balancing stage takes the mission plan obtained in the previous step as input. It revises the agents' plans as follows. The agent with the maximum makespan distributes the workload to other agents to reduce the mission makespan. This distribution is done by checking if tasks in the maximal agent's schedule could be inserted into another agent's schedule without increasing the maximum makespan. The plan of any other agent can be expanded up to a makespan value strictly less than the mission makespan. New tasks can be added, or the current time commitment to tasks already in their plan can be increased. The process repeats as long as the value of the mission makespan can be decreased.

V. DISTRIBUTED PLANNING IN COMMUNICATION-CONSTRAINED ENVIRONMENTS

Potential application domains of the STASP-HMR include SAR missions that require coordinating multiple agents' efforts (e.g., transporting items, mapping, and detecting dangerous substances) over large areas. The nature of the tasks introduces inherent uncertainties regarding their execution. Furthermore, the dimensions of the area and the potential lack of networking infrastructure imply that network connectivity cannot be taken for granted. Due to these complicating factors, mission planning cannot usually be done in an open-loop modality, that is, computing a mission plan once before the start of the mission and then letting the agents execute their assigned tasks.

To deal with these issues, we now describe how CoLoSSI can be fully decentralized. In this implementation, mission planning is performed iteratively, in a closed-loop modality, to account for new evidence, leverage the intermittent network connections, and deal with unexpected issues.

A. SHARED KNOWLEDGE REPRESENTATION

We assume that all agents know all the tasks composing the mission in advance. We also assume that agents can share their task efficiency models with other agents. During the mission, each agent tracks the amount of time it spends doing each task. This information is described in *incremental updates*, which allows reconstructing a local estimation of the completion map C_m once added up. An incremental update is a tuple $< k, i, t_{start}, t_{end} >$ that states that agent k has provided service to task i from time t_{start} to t_{end}.

These updates are generated by the agent that performs the service and can be shared with other agents. During the mission, all agents keep a list of the updates it has generated so far and the updates it has received from other agents. At any time, an agent can estimate the current completion of a task i, by estimating the amount of work that task has received based on the list of incremental updates and the task efficiency models and subtracting this value from the initial workload. As a result, each agent has a local estimation of the completion map.

B. INFORMATION EXCHANGE

In real-world deployments, data communication is usually supported by wireless network devices. However, this communication can be sparse when networked teams operate over large-scale environments, and the communication range of the network restricts data exchange.

Based on the shared representation, the final goal is to exploit the sparse communication interactions among the agents at most. To this end, a gossip-based mechanism [52] for information exchange allows that information spreads among the team. The mechanism is implemented as follows. During the mission, agents broadcast periodic network discovery messages that announce their presence to other agents. Using these messages, each agent maintains a list of agents from which it has received a message in the recent time. From this list, and with a certain frequency, the agent chooses another agent to initiate a synchronization procedure. This procedure lets each agent merge and synchronize their list of updates, possibly improving their local perception of the mission.

C. ADAPTIVE ITERATIVE REPLANNING USING COLOSSI

The adaptive iterative replanning approach is based on continually acquiring new information about the environment and the assessment of the current status of the agents. It is implemented as a *re-optimization procedure* [53] that solves a sequence of static problem instances over time.

The iterative replanning allows us to overcome several challenges. In particular, when the information available concerning real-world environments are usually incomplete or imperfect (e.g., due to communication constraints), and the environment itself is often dynamic. While executing the tasks, the agents continuously gather new information about the state of the environment, and the current status of the agents and the tasks they are executing using the information exchange strategy described above. The updated information is aggregated and considered in the next replanning iteration in order to make new plans according to changing situations.

More specifically, the iterative replanning works as follows. At the start of the mission, all agents initiate a replanning procedure with their neighbors. Given the communication constraints, several subgroups may be formed. Each group executes CoLoSSI and computes an initial mission plan. Each agent decides on a replanning interval and starts executing its plan. Generally, each agent may decide on a different replanning interval (i.e., asynchronous replanning). After the interval of one agent expires, that agent triggers replanning with its neighbors. At this point, the corresponding agent sends a message to its neighbors indicating that a new replanning round should be performed. Note that a potentially different group of agents initiates the execution of CoLoSSI, and new plans – and new replanning intervals – for all the agents in that group are decided.

1) ADAPTIVE REPLANNING INTERVAL

In this work, we consider two ways the replanning intervals are defined: periodic and adaptive. With periodic intervals, replanning occurs at regular, fixed times. Instead, adaptive intervals are decided based on the mission's current (estimated) completion. Adaptive intervals allow the change in the frequency of replanning as the mission is being completed. In particular, in preliminary results, we noticed that high-frequency replanning is most useful at the initial stage of the mission but not as much in the latter stages.

Our implementation of adaptive replanning uses intervals defined as

$$F(I, \mathcal{T}') = I + 2\left\lfloor \frac{|\mathcal{T}'|}{|\mathcal{T}|} \right\rfloor \tag{2}$$

where I represents the initial replanning interval and $\mathcal{T}'$ represent the tasks already completed, that is $\mathcal{T}' = \{C_m(i) < \epsilon : i \in \mathcal{T}\}$.

Using Eq. 2, agents start the mission and replan at small intervals I. As the mission progresses, they gradually increase their replanning interval based on the number of tasks that have been completed.

VI. CO-COLOSSI: CONNECTIVITY-AWARE VARIANT FOR COMMUNICATION-RESTRICTED ENVIRONMENTS

We note that the base approach presented in CoLoSSI does not have any fixed mechanism for robots to maintain the groups they have formed nor recruit other robots they come across while doing their tasks. This means the robots need not be in the same group after completing one round of replanning. Each robot regularly looks around to find the existence of other robots and forms a group with them if they exist. Suppose a robot comes across another robot that does not belong to its group. In that case, they exchange their task efficacy model and move on. This primitive mission planning approach mainly aims to complete missions with no mechanism to prompt the robots to form and maintain bigger groups and complete the missions more efficiently.

One of the significant shortcomings of distributed planning in communication-restricted scenarios is that the robots may not be able to maintain the groups they formed initially or at some replanning cycles. This shortcoming also affects CoLoSSI when some of the following scenarios occur at replanning time:

- An agent is within communication range of a group of other agents that may include members not present in the agent's original group
- An agent ends up in a position where it is isolated from other agents and thus proceeds to complete all the remaining tasks by itself.

A. TEAM FORMATION USING MEETUP POINTS

1) SINGLE MEETUP STRATEGY

The first extension we propose to avoid the aforementioned scenarios is a well-defined mechanism that enables a robot to maintain its initial group after every replanning cycle. The

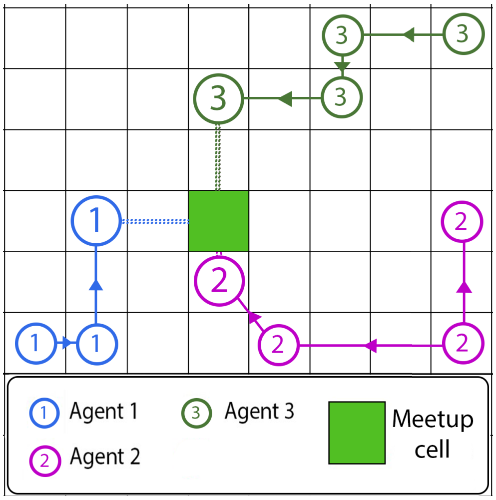

FIGURE 4. Illustrates robots deciding the meetup cell (green cell) based on their last position after completing their scheduled tasks.

objective is that once a robot is part of a group, it does not lose any of its members and, over time, keeps forming larger groups until its group encapsulates all the robots completing the mission.

This approach allows the robots to decide on a common task (location) to meet up at after executing all other tasks on their schedules. Therefore, at every replanning cycle, all the group's robots will be at the exact location (e.g., a sector in the SAR example) and look for other robots while maintaining their initial group. The calculation of which task to meet up on happens when the robots generate their mission plans for the next time interval using CoLoSSI in their respective groups. After the robots generate their mission plans, each one of them shares its last task. This is because the last task determines their final position after a particular time interval is completed. The robots then calculate a task relatively near each of the robots' last task/position and use this task as a meetup point, as shown in Fig. 4

We use the centroid of the locations of the last tasks in the plans of all robots in the group to determine the meetup task (and consequently the meetup point).

The robots will then add a path to the meetup task in their respective schedules.

To account for the time the robots will spend to travel to their meetup point, they partition the replanning interval into two sections. Suppose we consider T the total time robots have for an individual replanning cycle. Most of the time is allocated for completing the tasks and is represented by T_{work}. The remaining time is given for travel and is described as T_{travel}. The T_{travel} is chosen so all robots can reach the meetup point irrespective of their final position. At any replanning cycle, $T = T_{work} + T_{travel}$. The robots now generate their respective mission plans that will be completed within designated T_{work} time and then travel to the meetup point in the remaining time T_{travel} and replan at the meetup point when all the robots arrive.

Although using meetup points accomplishes maintaining and increasing group sizes, it has some shortcomings. Firstly, each robot of a particular group has decided on a calculated meetup task. Thus, they cannot recruit other robots they meet while completing their respective tasks, as this would be conflicting. Secondly, as each robot meetups at the same point, their area of communication with other robots is relatively small compared to what could have been possible given the number of robots. This is mainly because all robots are assigned to meet at a single point rather than spread up to cover more areas.

2) EXTENDED RADIUS STRATEGY

To address limitations inherent in the Single Meetup Strategy, we introduce the Extended Radius Strategy. This adaptation aims to significantly enhance the communication range of robot groups by optimizing their spatial distribution.

We extend the concept of meetup points to dramatically increase any group's communication area by spreading apart the robots over a place so that all are within communication range rather than meeting at a single point. This is done by stationing one of the robots at the meetup point called *hub robot* and the others in the periphery of the communication range of this *hub robot*. All other robots will now travel to any point on the edge of the hub robot. This will allow each group of robots to have a wider area of communication and enable them to detect other robots faster and more efficiently.

Fig. 5 shows an example of the proposed approach. This example showcases an environment where agents have a communication restriction with a distance of 1 cell. The solid lines and circles indicate tasks that have already been completed/traversed and the dashed lines indicate the scheduled tasks. As the figure shows, Agent 1 (in yellow) has been designated as the hub robot. Agents 2, 3, and 4 are scheduled to move to tasks that are spread out 1 to 2 cells away from Agent 1, thereby increasing the overall communication range of the group.

Although this approach allows a wider area of communication, it requires that the group executing this has at least three robots. This is a crucial prerequisite as without enough robots to act as a hub and link all robots, the robots can get separated if positioned far away. Therefore, we need at least three robots where one will take the role of *hub robot*, and others can station themselves at the periphery of the hub. The algorithm proceeds with the classical single meetup point strategy until the number of robots has reached the required number to implement the extended radius strategy. Once this condition is satisfied, the robots implement star topology.

B. RECRUITMENT TO FORM LARGER GROUPS

Finally, we also allow robots to recruit other robots of different groups if they encounter one another while doing tasks. Note that each robot has the efficacy model of the tasks and is aware of the work done by the entire group. Moreover, each robot here is now aware of the meetup point it has to return to, which implies that direct recruitment may conflict

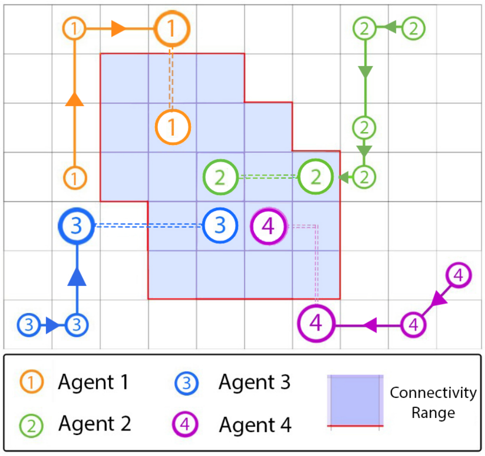

FIGURE 5. Shows the arrangement of the robots after a replanning cycle is completed in a 2-cell communication setting. Here one robot is at the hub (yellow cell), whereas the other robots are at the periphery of the hub (red cells).

as the robots have different meetup cells calculated by their respective groups. Therefore, they sync their efficacy model and attempt recruitment when the robots meet.

The initiated recruitment process is based on the following criteria:

- Cumulative work done by the respective groups
- Number of robots in the respective groups

The comparison is first made based on the cumulative work done by the respective groups which is found by summing the work done by each of the individual robots in that group. Suppose the cumulative work done by each group is the same. In that case, the respective group size of robots is compared, and the one with the bigger group recruits the robot. The rationale behind these criteria is to allow bigger groups that have done more work to grow in size to increase overall efficiency. If the robots have the same work done and the same group size, the robot is recruited into any of the groups. Moreover, each robot can recruit at most one robot during a specific replanning cycle. This prevents further conflicts when a robot shifts from one group to another. The recruiter moves the new robot to the periphery of the *hub robot* or directly to the meetup point if the robot's group size is less than required for the meetup points with an extended radius. This is done by performing another round of replanning of the recruiter and the recruited robot. At an instant, let T denote the time allocated per replanning cycle. If the recruitment process were initiated at some time T', the replanning would only happen for the remaining time, i.e., $T - T'$ time units. Performing tasks in specific recruitment scenarios might be impossible due to time constraints. In that case, the agents directly move toward the meetup point.

In this approach, the time interval allocated to the robot needs to be partitioned in a calculated manner so that robots can arrive at their stationed position irrespective of their final location. For this, the time interval is partitioned similarly to the proposed meetup strategy.

VII. EMPIRICAL SETUP

In this section, we outline the empirical setup used to evaluate the performance of the proposed algorithms in simulated search and rescue (SAR) missions. Our focus is on the structure of the search areas, composition of agent teams, computational approach for solution comparison, agent dynamics, and communication model of the environment. Table 2 summarizes the main parameters used in the evaluation.

A. SEARCH AREA STRUCTURE

The search areas are pre-determined and represented in a cellular grid format. Each cell in the grid signifies a potential search location for the agents. We analyze the algorithm's efficacy across three distinct grid sizes: (i) a $500 \times 500\ m^2$ area, (ii) a $1000 \times 1000\ m^2$ area, and (iii) a $2000 \times 2000\ m^2$ area. Each cell in these grids spans an area of $100 \times 100\ m^2$, providing a consistent basis for comparing different grid sizes. These specific sample sizes were selected to represent small, medium, and large search areas commonly encountered in SAR missions. This selection ensures that our evaluation comprehensively covers different operational scales, thereby enhancing the robustness and validity of our results.

B. AGENT TEAM COMPOSITION

Our approach utilizes heterogeneous agent teams commonly deployed in SAR operations. These teams comprise four distinct types of agents: (i) human rescuers, (ii) scent-tracking dogs, (iii) aerial drones (specifically quadrotors), and (iv) fixed-wing drones. The number of agents in each problem instance varies from 4 to 20 while maintaining an equal distribution of each agent type. Furthermore, each agent is characterized by one of four efficiency levels for task completion, ranging from highly inefficient (sixteen time steps per task) to highly efficient (two time steps per task). This range of agent counts and efficiency levels allows for a thorough assessment of our algorithms under various operational conditions, ensuring that our findings are robust and applicable to real-world SAR scenarios.

C. COMPUTATIONAL COMPARISON APPROACH

To benchmark our algorithm, we employ CPLEX®, a mathematical solver, as a comparative tool. The solver operates under constraints, including a one-hour time limit, a 4 GB memory cap, and a maximum of 2 parallel threads. In scenarios where computational limits are reached, CPLEX provides the best solution found within the timeframe, accompanied by a relative optimality gap or MIP gap. Due to the stringent computational constraints, CPLEX often yields sub-optimal solutions derived from its branch-and-bound search process. It should be noted that the CPLEX solver is a centralized solution algorithm, and we use this to benchmark our decentralized algorithms. This comparison provides a rigorous benchmark, ensuring that

TABLE 2. Summary of parameters used in the evaluation.

Parameter	Values
Grid size	5x5, 10x10, 20x20
Number of agents	4, 6, 8, 10, 12, 14, 16, 18, 20
Number of Agent Types	1, 2, 4
Agent Efficacy φ	0.0625, 0.125, 0.25, 0.5
Solver runs	50
Instances per grid size	[200, 500]

our decentralized approaches are evaluated against a high-standard baseline, thereby reinforcing the validity of our results.

D. AGENT DYNAMICS AND INFORMATION EXCHANGE

In these environments, agents are initially grouped and positioned at various locations. Each agent holds an updated copy of the completion map, which reflects the overall mission progress. This map is dynamically updated as agents complete tasks and encounter other agents. For instance, if Agent One finishes five tasks and Agent Two completes four different tasks, upon meeting, they will synchronize their maps, collectively reflecting the completion of nine tasks. This synchronization is a key feature in efficiently managing and tracking mission progress.

E. COMMUNICATION MODEL AND GROUP FORMATION

Communication among agents is modeled based on a Euclidean open-space disk, with a defined maximum range for signal transmission and reception. This limited communication range necessitates dynamic detection of other agents, facilitating group formation. Agents use an auction-based algorithm for task distribution within these groups. Agents exchange information and synchronize their completion maps during group formation and task completion. Additionally, if an agent is isolated during the replanning phase, it assumes responsibility for all remaining tasks, underlining the importance of agent connectivity for efficient mission execution.

VIII. RESULTS AND DISCUSSION

This section covers the evaluation of the proposed mission planning approaches in various environments. We present the evaluation of our algorithm in the context of SAR scenarios in the wilderness (WiSAR), which is adapted from previous works [48].

A. PERFORMANCE ANALYSIS OF COLOSSI VS. CPLEX®

We compare the performance of the CoLoSSI algorithm against the CPLEX® mathematical solver in terms of solution optimality and computational efficiency to establish the suitability of CoLoSSI for makespan estimation. Specifically, we analyze the CoLoSSI algorithm's effectiveness in managing different sizes of agent teams under specific spatial and temporal restrictions. This involved acquiring 20 separate instances for each team size and running CoLoSSI 10 times per instance to calculate the mean makespan. In parallel, we solved each instance with the CPLEX solver, using its output as a standard for solution quality. Figure 6 presents the optimality gap, which measures the performance difference between CoLoSSI's solutions and those derived from the CPLEX solver.

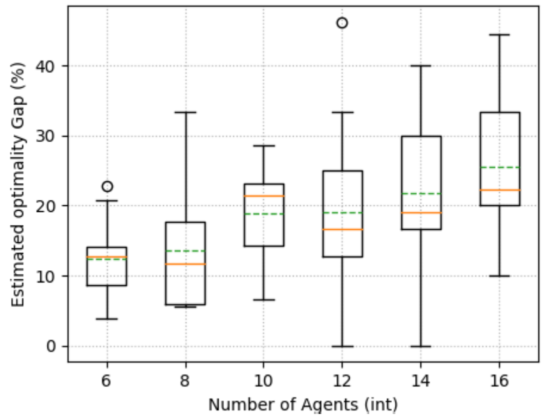

FIGURE 6. Empirical estimation of the optimality gap of solutions provided by CoLoSSI in 5 × 5 mission layouts.

Our analysis indicates that the solutions from CoLoSSI are within a 34% optimality gap of the CPLEX solutions. For smaller teams of 6 and 8 agents, CoLoSSI maintains a median optimality gap of approximately 10% and 12%, respectively, showcasing superior performance in scenarios with fewer operational agents. As the agent count increases to 10, the median gap rises to 15%, possibly due to the coordination challenges and increased complexity of the auctioning process that are less prevalent with smaller teams. When the number of agents increases to 12, the median gap reaches 18%. This indicates that the algorithm is starting to leverage the increased agent count more effectively.

However, for larger teams of 14 and 16 agents, the median optimality gaps further increase to approximately 22% and 25%. This is due to the compounding complexity of task coordination among a greater number of agents. The increase in the median gap is coupled with a widening of the IQR, particularly noticeable in the 14-agent scenarios, suggesting greater variability in solution quality for larger teams. The centralized coordination role of the auctioneer in managing the bids and distributing tasks becomes more challenging as the number of agents grows. The auction process, though designed to handle concurrent bids, may not always result in the most efficient agent-task pairings, leading to suboptimal task assignments and increased overall mission time. The variability in the performance of CoLoSSI with medium-sized teams of 10 and 12 agents highlights the algorithm's sensitivity to the number of agents and the complexity of their coordination. This illustrates the need for refined strategies within the CoLoSSI framework to ensure scalability and maintain solution quality as agent numbers increase.

TABLE 3. Comparison of makespan estimations for missions executed by Sequential Single Item (SSI) Auctioning algorithm and CoLoSSI Algorithm on 10 × 10 grid and 20 × 20 grid sizes.

No. of Agents	10x10 Grid		20x20 Grid	
	SSI	CoLoSSI	SSI	CoLoSSI
4	295	143	1397	687
6	256	127	1294	648
8	207	98	1195	593
10	154	76	1092	548
12	123	61	996	497
14	103	52	897	442
16	84	39	803	398
18	71	34	702	347

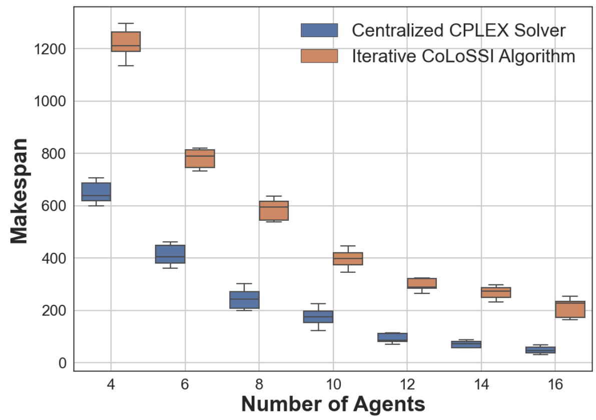

FIGURE 7. Comparison of makespan between the centralized CPLEX solver and the iterative CoLoSSI algorithm with a 25-time unit replanning interval.

B. BENCHMARKING COLOSSI AGAINST STATE-OF-THE-ART SEQUENTIAL SINGLE ITEM AUCTIONING

In assessing the effectiveness of CoLoSSI's enhancements for task scheduling, we observe a detailed comparative analysis against the state-of-the-art Sequential Single Item (SSI) Auctioning algorithm. This comparison spans 500 and 210 unique scenarios within 10 × 10 and 20 × 20 grid environments, respectively, each subjected to 50 independent algorithm runs to guarantee statistical reliability.

Our findings, as showcased in Table 3, highlight CoLoSSI's significant advances in reducing makespan across a wide array of test cases. A notable observation is the decreasing trend in makespan with an increasing count of agents, consistent across both grid setups. Specifically, within the 10 × 10 grid scenarios, the SSI algorithm's median makespan of 295 time units is notably reduced to 143 time units when applying CoLoSSI. Similarly, for the more challenging 20 × 20 grid configurations, the significant decrease in makespan from 1397 to 687 time units for the same number of agents underscores CoLoSSI's scalability and its strategic adaptability to more complex and demanding environments. This level of performance enhancement, consistently achieving over 50% improvement across various configurations, not only reinforces CoLoSSI's superiority over the SSI approach but also emphasizes its potential to redefine operational benchmarks in multi-agent systems.

An equally important observation from the results is the consistency in performance enhancement post-CoLoSSI refinement, as evidenced by the convergence in makespan values between the SSI and CoLoSSI algorithms. This consistency is not a trivial outcome but a testament to the sophistication of CoLoSSI's negotiation and cooperation mechanisms, which ensure that the collective effort of the agents is aligned towards the most efficient completion of tasks.

Such remarkable improvements stem from two pivotal technical advancements inherent to CoLoSSI. Firstly, the introduction of a bespoke bidding function adept at navigating the diverse capabilities within the agent team, paving the way for a more meticulously strategic allocation of tasks. The second advancement involves a post-bid negotiation process, fostering agent collaboration and equitable task distribution. These enhancements collectively contribute to a significant reduction in overall mission completion time, affirming CoLoSSI's optimized efficiency over the SSI algorithm, the recognized state-of-the-art in task distribution and scheduling.

C. COMMUNICATION-RESTRICTED SCENARIOS

1) COLOSSI'S ITERATIVE REPLANNING PERFORMANCE

We analyze the cost of the iteratively replanning CoLoSSI algorithm's capabilities, particularly in comparison to the centralized solution approach represented by the CPLEX solver in terms of makespan. To ensure a robust comparison, we executed CoLoSSI across different scenarios with varying numbers of agents, each subjected to 50 iterations to ensure statistical stability especially when dealing with environments with network constraints and limited connectivity similar to real-world environments. The orange series in figure 7 represents the makespan outcomes of CoLoSSI's adaptive iterative replanning method, whereas the blue series corresponds to the centralized approach's results.

Looking at the interquartile range (IQR) of the data in figure 7, we notice that for the CPLEX solver, the IQR decreases steadily when the number of agents increases, ranging from about 700-time units with 4 agents to about 100-time units with 16 agents. This trend can be explained by the centralized solver's capability to optimally allocate tasks to a growing number of resources, thereby reducing the variability in task completion times. In contrast, the IQR for the iterative CoLoSSI algorithm does not display a consistent reduction. Notably, the box for 16 agents is larger than for other higher agent counts, which is counterintuitive as one would expect lower variability with more agents. This is because at a certain point, adding more agents leads to a complexity threshold beyond which the algorithm's coordination mechanism–relying on frequent communication and synchronization–begins to saturate. This saturation could result in certain agents being underutilized due to communication bottlenecks, leading to a wider spread in makespan outcomes. It should be noted that for both the CPLEX solver and iterative CoLoSSI when the number of agents is small (4 to 8), the IQR has a considerable breadth to it. However, when the number of agents increases (10 to 16), the IQR is narrow except for 16 agents for iterative CoLoSSI.

2) ANALYSIS OF REPLANNING INTERVAL EFFICACY

In assessing the impact of replanning intervals on mission efficiency, our simulations incorporated various team sizes and applied different interval strategies, executing 50 runs per configuration. Our approach utilized a diverse agent team composition with varying levels of task efficiency, showing realistic heterogeneity in agent capabilities. Figure 8 indicates a direct correlation between the frequency of replanning and task completion speed. Shorter intervals consistently led to quicker mission completions across different mission scales. This efficiency can be attributed to the agents' increased opportunity to integrate new information and collaborate effectively at an earlier stage. The frequency with which teams engage in replanning provides insight into their operational efficiency. Figure 9 details the number of replans executed before mission completion across various team sizes and grid layouts. We observe that a higher frequency of replanning directly correlates with reduced mission makespan, showcasing the benefits of frequent information sharing and coordination among agents.

Examining the adaptive replanning strategy based on Eq. 2, we found that its performance closely mirrors that of the most frequent fixed interval (5-time units), yet with fewer replanning events. This finding supports the hypothesis that frequent initial replanning is crucial for gathering information and forming effective agent sub-teams. As the mission progresses and the agent's knowledge of the environment aligns more closely, the added value of replanning frequently starts to decrease. These observations indicate that while regular replanning can lead to quicker task completions at the start of the mission, its effectiveness decreases over time. The adaptive method provides a measured approach, adjusting the rate of replanning to suit the current stage of the mission without causing unnecessary complications through too much coordination. This careful management is crucial in real-world situations where computational power and the ability to respond quickly are limited and valuable.

The results from our simulations support using the adaptive replanning strategy in CoLoSSI, showing that varying the replanning intervals in response to how much of the mission has been completed can improve overall mission performance. By employing this strategy, agents use the latest information to coordinate their actions effectively while avoiding the less productive aspects of replanning too often as the mission nears its end. This adaptive method is, therefore, a sensible choice for managing tasks among distributed agents in changing conditions.

3) COMPARATIVE EVALUATION OF CO-COLOSSI'S MEETUP STRATEGIES

The performance of the Co-CoLoSSI algorithm is influenced significantly by choosing the correct replanning interval strategy and meetup strategy. Our evaluation compares the extended radius approach with the single-cell meetup strategy, examining their efficacy in completing missions across a range of replanning intervals. Figure 10 indicates a clear advantage for the extended radius strategy over the single-cell meetup approach, particularly evident in teams with fewer agents. The extended radius strategy's superior performance can be attributed to its extended range of communication and coordination, allowing agents to form larger and more effective groups. This is crucial in scenarios with fewer agents, where the ability to rapidly form larger teams can significantly reduce the makespan by minimizing redundant task execution. However, as the size of agent groups increases, the disparity in performance between the two strategies diminishes. For groups with more than 12 agents, the makespan results of both strategies converge, suggesting that the benefits of extended communication range are less pronounced when the agent team is already large. In such cases, the inherent communication capabilities within a large team may suffice to maintain efficiency, rendering the extended radius strategy's advantage less impactful. Consistently, the adaptive replanning interval strategy has shown to be the most effective, dynamically aligning the frequency of replanning with the mission's progression. This adaptability is advantageous, offering a balance between the need for regular updates in the mission's early stages and reduced frequency as the mission nears completion, thus optimizing resource utilization throughout the operation. This analysis reinforces the superiority of the adaptive replanning interval, especially when considering the operational flexibility it provides. By adjusting to the real-time demands of the mission, this strategy ensures that agents are working with the most current information in a way tailored to the evolving mission dynamics. This approach not only streamlines mission execution but also maximizes the effective use of available resources, a critical consideration in time-sensitive scenarios.

In summary, we notice that the extended radius strategy shows significant efficiency improvements in smaller teams especially when employing the adaptive replanning interval as its replanning strategy. This combination of strategies emerges as the most consistently effective approach across various team sizes and mission stages.

4) EVALUATING RECRUITMENT'S ROLE IN MISSION EFFICIENCY

The extended-radius Co-CoLoSSI algorithm introduces a recruitment feature, allowing robots to incorporate others into their group during mission execution. We modified the Co-CoLoSSI with an extended radius to restrict robots from recruiting peers and compared this with the standard

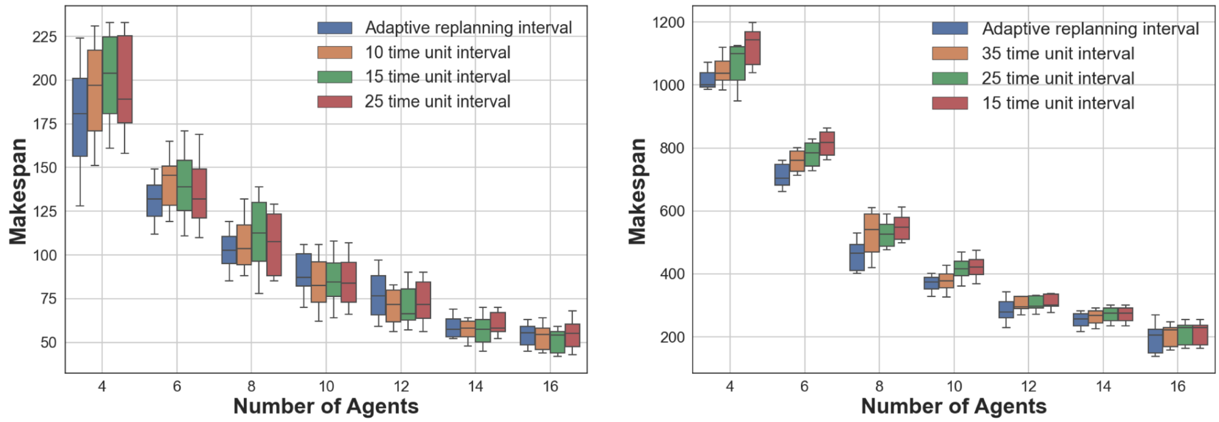

FIGURE 8. Performance comparison of the iterative CoLoSSI in different mission layouts: a 10 × 10 grid layout (*left*) and a 20 × 20 grid layout (*right*). The figure illustrates the system's efficiency under communication constraints in each scenario.

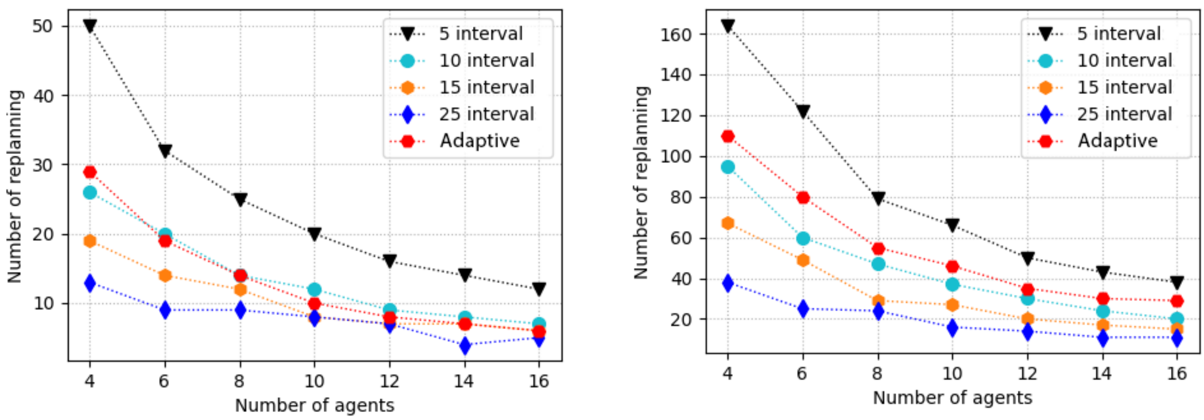

FIGURE 9. Frequency of replanning for heterogeneous teams of varying sizes in mission layouts: a 10 × 10 grid (left) and a 20 × 20 grid (right), as generated by CoLoSSI. This highlights how team size impacts the replanning frequency in different spatial configurations.

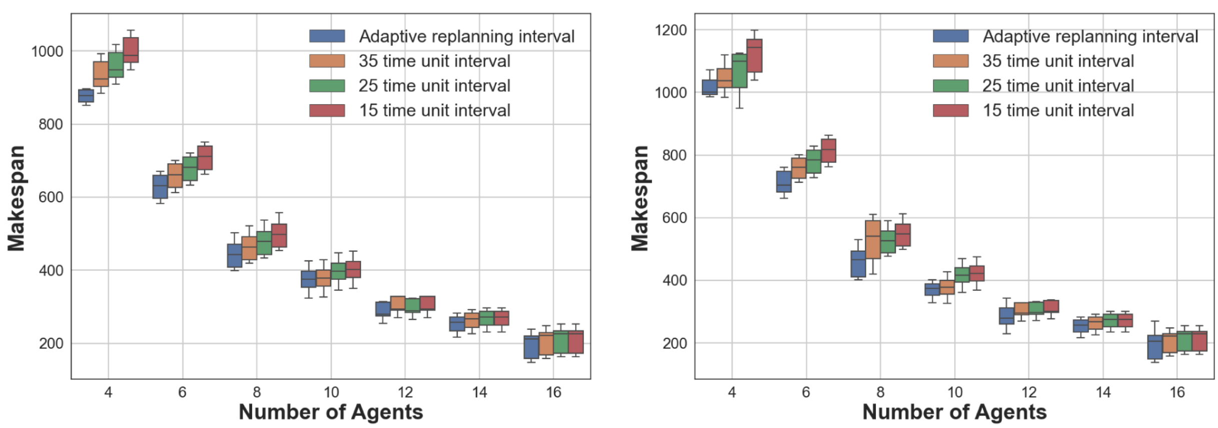

FIGURE 10. Evaluating the Co-CoLoSSI algorithm's makespan across various meetup strategies and replanning intervals in a communication-restricted environment.

version that allows recruitment. Figure 11 contrasts the makespan results from the version without recruitment against those where recruitment is active. Both versions of the Co-CoLoSSI algorithm were executed with the *adaptive replanning interval* for consistency and fair comparison. Moreover, as proved in the aforementioned results, the *adaptive replanning interval* yields the lowest mission makespan. In this manner, we can see the effect of recruitment while keeping the replanning interval at its optimal setting. The data clearly show that enabling robots to recruit others consistently reduces the time taken to complete the mission.

The benefit of recruiting is particularly evident with fewer robots in the field. With limited numbers, each robot's ability to join larger groups is crucial, as it helps to minimize task overlap and inefficient duplication of effort. In the case of

FIGURE 11. Comparison of the mission completion times (makespans) for Co-CoLoSSI with extended radius, shown both with recruitment and without recruitment strategies. This illustrates the impact of recruitment on operational efficiency.

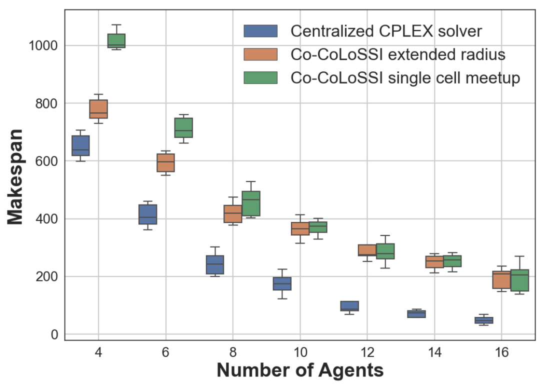

FIGURE 12. Comparative analysis of Decentralized Co-CoLoSSI Algorithm with Adaptive Replanning and different meetup strategies and Active Recruitment Versus Centralized CPLEX Solver.

smaller group sizes, the isolation of robots into smaller, independent groups can lead to a fragmented approach to task management. This fragmentation often results in redundant task execution, an inefficient allocation of resources, and, consequently, an inflated makespan. Conversely, enabling recruitment allows for a more cohesive and unified task strategy, effectively reducing unnecessary task redundancy. It should be noted that the effect of the recruitment strategy noticeably becomes less significant when the group sizes of the robots are large. This trend can be observed by looking at the groups with more than 10 agents in Fig. 11. This is because, during the start of the mission, the groups are already large enough to distribute the tasks very effectively, and any additional recruitment does not significantly increase the task distribution efficacy.

The results affirm the recruitment strategy's effectiveness, especially in small-scale agent teams, where the advantage of pooling resources through recruitment substantially enhances mission efficiency.

5) COMPARATIVE ANALYSIS OF DECENTRALIZED CO-COLOSSI AND CENTRALIZED CPLEX SOLVER

we compare the performance of the Co-CoLoSSI algorithm, which incorporates adaptive replanning, Extended radius meetup, and active recruitment, against the centralized CPLEX solver within communication-restricted environments. Analyzing Fig. 12, we note that the Co-CoLoSSI algorithm with the extended radius strategy significantly improves performance compared to the single-cell meetup strategy, particularly for smaller teams of agents. The extended radius strategy shows a median makespan that is up to 25% lower than that of the single-cell strategy when fewer than 10 agents are used, indicating its effectiveness in improving agent coordination and task completion, even with limited communication. As agent numbers increase beyond 10, the performance of both Co-CoLoSSI strategies begins to converge, suggesting that the larger the group, the less impactful additional recruitment becomes for task distribution efficiency. Figure 12 indicates a significant trend that the Co-CoLoSSI algorithm utilizing an adaptive replanning interval combined with the extended radius meetup strategy and active recruitment aligns closely with the performance of the centralized CPLEX solver. This trend is particularly noteworthy because, although the centralized solver offers an optimal solution in a centralized manner, the Co-CoLoSSI's algorithm achieves comparable efficiency under the realistic constraints of limited communication while executing in a decentralized manner.

This evidence reinforces the Co-CoLoSSI as a viable alternative to centralized methods, particularly when centralized control is infeasible. The decentralized algorithm showcases the agility required in rapidly evolving scenarios, ensuring that agents operate at peak coordination efficiency despite intermittent communication. The ability of Co-CoLoSSI to adapt to the number of agents and the progression of the mission while dynamically adjusting its replanning intervals and group strategies–translates into a flexible, scalable, and efficient approach to decentralized task management.

In the context of missions where communication channels are often compromised, the configuration comprising the Co-CoLoSSI algorithm with adaptive replanning, extended radius meetup, and active recruitment stands out as the most robust decentralized approach. It demonstrates operational competence, closely approaching the centralized solution's makespan while executing in a fraction of the time taken by the centralized approach. This analysis confirms the Co-CoLoSSI algorithm's advanced capabilities, presenting it as a near-optimal decentralized solution for complex, communication-limited operational environments.

IX. CONCLUSION

We present CoLoSSI, an auction-based solution approach for cooperative, load-balancing task allocation and scheduling using sequential single-item auctions. The work was motivated by the computation and communication constraints of real-world scenarios, mainly due to the lack of network infrastructure and the need for rapid re-computation of plans to adapt to the mission dynamics. The main features of the algorithm are a bidding function that optimizes completion time and explicitly takes into account team heterogeneity and non-atomic task completion and two post-processing schemes that further enhance team cooperation and load-balancing. We also introduce a variant, named Co-CoLoSSI, which improves the performance of CoLoSSI in communication-limited environments. The main features of Co-CoLoSSI include effective ways of providing communications to the robots based on the occurrence of spatial relations among the robots. These spatial relations are arranged around meetup points that are enforced inside the auction-based algorithm, and aimed at favoring communication during the execution of the algorithm. We validate the scalability of our methods and show that we could balance the trade-off between computation vs. coordination with the distributed system, in general, performing within 50% with respect to centralized implementation. Furthermore, results show that the connectivity-aware variant Co-CoLoSSI improves the team's performance by promoting network connectivity.

REFERENCES

[1] Y. Rizk, M. Awad, and E. W. Tunstel, "Cooperative heterogeneous multi-robot systems: A survey," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–31, Apr. 2019.

[2] E. F. Flushing, L. M. Gambardella, and G. A. Di Caro, "On decentralized coordination for spatial task allocation and scheduling in heterogeneous teams," in *Proc. 15th Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 988–996.

[3] M. Otte, M. J. Kuhlman, and D. Sofge, "Auctions for multi-robot task allocation in communication limited environments," *Auto. Robots*, vol. 44, nos. 3–4, pp. 547–584, Mar. 2020.

[4] S. Koenig, C. Tovey, M. Lagoudakis, V. Markakis, D. Kempe, P. Keskinocak, A. Kleywegt, A. Meyerson, and S. Jain, "The power of sequential single-item auctions for agent coordination," in *Proc. Natl. Conf. Artif. Intell.*, vol. 2, 2006, pp. 1625–1629.

[5] N. Sullivan, S. Grainger, and B. Cazzolato, "Sequential single-item auction improvements for heterogeneous multi-robot routing," *Robot. Auto. Syst.*, vol. 115, pp. 130–142, May 2019.

[6] S. S. Ponda, L. B. Johnson, A. N. Kopeikin, H.-L. Choi, and J. P. How, "Distributed planning strategies to ensure network connectivity for dynamic heterogeneous teams," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 5, pp. 861–869, Jun. 2012.

[7] Y. Cao, T. Long, J. Sun, Z. Wang, and G. Xu, "Comparison of distributed task allocation algorithms considering non-ideal communication factors for multi-UAV collaborative visit missions," *IEEE Robot. Autom. Lett.*, vol. 1, no. 1, pp. 1–8, Aug. 2024.

[8] S. Nayak, S. Yeotikar, E. Carrillo, E. Rudnick-Cohen, M. K. M. Jaffar, R. Patel, S. Azarm, J. W. Herrmann, H. Xu, and M. Otte, "Experimental comparison of decentralized task allocation algorithms under imperfect communication," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 572–579, Apr. 2020.

[9] I. Ansari, A. Mohamed, E. F. Flushing, and S. Razak, "Cooperative and load-balancing auctions for heterogeneous multi-robot teams dealing with spatial and non-atomic tasks," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Nov. 2020, pp. 213–220.

[10] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *Int. J. Robot. Res.*, vol. 23, no. 9, pp. 939–954, Sep. 2004.

[11] 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, Oct. 2013.

[12] E. Nunes, M. Manner, H. Mitiche, and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints," *Robot. Auto. Syst.*, vol. 90, pp. 55–70, Apr. 2017.

[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. Nunes and M. Gini, "Multi-robot auctions for allocation of tasks with temporal constraints," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 1–24.

[15] Q. Xu, Z. Su, D. Fang, and Y. Wu, "BASIC: Distributed task assignment with auction incentive in UAV-enabled crowdsensing system," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. 2416–2430, Feb. 2024.

[16] C. Street, B. Lacerda, M. Muhlig, and N. Hawes, "Right place, right time: Proactive multi-robot task allocation under spatiotemporal uncertainty," *J. Artif. Intell. Res.*, vol. 79, pp. 137–171, Jan. 2024.

[17] G. Picard, "Multi-agent consensus-based bundle allocation for multi-mode composite tasks," in *Proc. Int. Conf. Auto. Agents Multiagent Syst. (AAMAS)*, 2023, pp. 504–512.

[18] M. Goarin and G. Loianno, "Graph neural network for decentralized multi-robot goal assignment," *IEEE Robot. Autom. Lett.*, vol. 9, no. 5, pp. 4051–4058, May 2024.

[19] J. G. Martin, F. J. Muros, J. M. Maestre, and E. F. Camacho, "Multi-robot task allocation clustering based on game theory," *Robot. Auto. Syst.*, vol. 161, Mar. 2023, Art. no. 104314.

[20] S. Sariel-Talay, T. R. Balch, and N. Erdogan, "A generic framework for distributed multirobot cooperation," *J. Intell. Robotic Syst.*, vol. 63, no. 2, pp. 323–358, Aug. 2011.

[21] S. Raja, G. Habibi, and J. P. How, "Communication-aware consensus-based decentralized task allocation in communication constrained environments," *IEEE Access*, vol. 10, pp. 19753–19767, 2022.

[22] O. Shorinwa, R. N. Haksar, P. Washington, and M. Schwager, "Distributed multirobot task assignment via consensus ADMM," *IEEE Trans. Robot.*, vol. 39, no. 4, pp. 1–20, Jun. 2022.

[23] M. B. Dias, R. Zlot, N. Kalra, and A. Stentz, "Market-based multirobot coordination: A survey and analysis," *Proc. IEEE*, vol. 94, no. 7, pp. 1257–1270, Jul. 2006.

[24] F. Quinton, C. Grand, and C. Lesire, "Market approaches to the multi-robot task allocation problem: A survey," *J. Intell. Robotic Syst.*, vol. 107, no. 2, p. 29, Feb. 2023.

[25] M. D'Emidio and I. Khan, "Collision-free allocation of temporally constrained tasks in multi-robot systems," *Robot. Auto. Syst.*, vol. 119, pp. 151–172, Sep. 2019.

[26] M. Y. Ansari, I. A. Changaai Mangalote, P. K. Meher, O. Aboumarzouk, A. Al-Ansari, O. Halabi, and S. P. Dakua, "Advancements in deep learning for B-mode ultrasound segmentation: A comprehensive review," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 8, no. 3, pp. 2126–2149, Jun. 2024.

[27] P. Rai, M. Y. Ansari, M. Warfa, H. Al-Hamar, J. Abinahed, A. Barah, S. P. Dakua, and S. Balakrishnan, "Efficacy of fusion imaging for immediate post-ablation assessment of malignant liver neoplasms: A systematic review," *Cancer Med.*, vol. 12, no. 13, pp. 14225–14251, Jul. 2023.

[28] M. Y. Ansari, I. A. C. Mangalote, D. Masri, and S. P. Dakua, "Neural network-based fast liver ultrasound image segmentation," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jun. 2023, pp. 1–8.

[29] J. Ma, Y. He, F. Li, L. Han, C. You, and B. Wang, "Segment anything in medical images," *Nature Commun.*, vol. 15, no. 1, p. 654, Jan. 2024.

[30] M. Y. Ansari, M. Qaraqe, R. Righetti, E. Serpedin, and K. Qaraqe, "Unveiling the future of breast cancer assessment: A critical review on generative adversarial networks in elastography ultrasound," *Frontiers Oncol.*, vol. 13, Dec. 2023, Art. no. 1282536.

[31] H. Altaheri, G. Muhammad, M. Alsulaiman, S. U. Amin, G. A. Altuwaijri, W. Abdul, M. A. Bencherif, and M. Faisal, "Deep learning techniques for classification of electroencephalogram (EEG) motor imagery (MI) signals: A review," *Neural Comput. Appl.*, vol. 35, no. 20, pp. 14681–14722, Jul. 2023.

[32] Y. Ansari, O. Mourad, K. Qaraqe, and E. Serpedin, "Deep learning for ECG arrhythmia detection and classification: An overview of progress for period 2017–2023," *Frontiers Physiol.*, vol. 14, Sep. 2023, Art. no. 1246746.

[33] Y. Huang, H. Li, and X. Yu, "A novel time representation input based on deep learning for ECG classification," *Biomed. Signal Process. Control*, vol. 83, May 2023, Art. no. 104628.

[34] M. Y. Ansari, M. Qaraqe, F. Charafeddine, E. Serpedin, R. Righetti, and K. Qaraqe, "Estimating age and gender from electrocardiogram signals: A comprehensive review of the past decade," *Artif. Intell. Med.*, vol. 146, Dec. 2023, Art. no. 102690.

[35] V. Chandrasekar, M. Y. Ansari, A. V. Singh, S. Uddin, K. S. Prabhu, S. Dash, S. A. Khodor, A. Terranegra, M. Avella, and S. P. Dakua, "Investigating the use of machine learning models to understand the drugs permeability across placenta," *IEEE Access*, vol. 11, pp. 52726–52739, 2023.

[36] M. Y. Ansari, V. Chandrasekar, A. V. Singh, and S. P. Dakua, "Re-routing drugs to blood brain barrier: A comprehensive analysis of machine learning approaches with fingerprint amalgamation and data balancing," *IEEE Access*, vol. 11, pp. 9890–9906, 2023.

[37] H. Askr, E. Elgeldawi, H. A. Ella, Y. A. M. M. Elshaier, M. M. Gomaa, and A. E. Hassanien, "Deep learning in drug discovery: An integrative review and future challenges," *Artif. Intell. Rev.*, vol. 56, no. 7, pp. 5975–6037, Jul. 2023.

[38] M. Y. Ansari and M. Qaraqe, "MEFood: A large-scale representative benchmark of quotidian foods for the middle east," *IEEE Access*, vol. 11, pp. 4589–4601, 2023.

[39] E. Feo-Flushing, L. M. Gambardella, and G. A. D. Caro, "Spatially-distributed missions with heterogeneous multi–robot teams," *IEEE Access*, vol. 9, pp. 67327–67348, 2021.

[40] F. Zitouni, S. Harous, and R. Maamri, "A distributed approach to the multi-robot task allocation problem using the consensus-based bundle algorithm and ant colony system," *IEEE Access*, vol. 8, pp. 27479–27494, 2020.

[41] S. Chen, T. X. Lin, S. Al-Abri, R. C. Arkin, and F. Zhang, "Hybrid SUSD-based task allocation for heterogeneous multi-robot teams," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 1400–1406.

[42] N. Dahlquist, B. Lindqvist, A. Saradagi, and G. Nikolakopoulos, "Reactive multi-agent coordination using auction-based task allocation and behavior trees," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2023, pp. 829–834.

[43] M. Lippi, P. Di Lillo, and A. Marino, "A task allocation framework for human multi-robot collaborative settings," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 7614–7620.

[44] Y. Kantaros and M. M. Zavlanos, "Distributed communication-aware coverage control by mobile sensor networks," *Automatica*, vol. 63, pp. 209–220, Jan. 2016.

[45] J. Scherer and B. Rinner, "Multi-robot persistent surveillance with connectivity constraints," *IEEE Access*, vol. 8, pp. 15093–15109, 2020.

[46] J. Stephan, J. Fink, V. Kumar, and A. Ribeiro, "Concurrent control of mobility and communication in multirobot systems," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1248–1254, Oct. 2017.

[47] Y. Marchukov and L. Montano, "Communication-aware planning for robot teams deployment," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6875–6881, Jul. 2017.

[48] E. F. Flushing, M. Kudelski, L. M. Gambardella, and G. A. Di Caro, "Connectivity-aware planning of search and rescue missions," in *Proc. SSRR*, Oct. 2013, pp. 1–8.

[49] H. Shen, L. Pan, and J. Qian, "Research on large-scale additive manufacturing based on multi-robot collaboration technology," *Additive Manuf.*, vol. 30, Dec. 2019, Art. no. 100906.

[50] Z. Y. Lim, S. G. Ponnambalam, and K. Izui, "Multi-objective hybrid algorithms for layout optimization in multi-robot cellular manufacturing systems," *Knowl.-Based Syst.*, vol. 120, pp. 87–98, Mar. 2017.

[51] C. Zhang and N. Noguchi, "Development of a multi-robot tractor system for agriculture field work," *Comput. Electron. Agricult.*, vol. 142, pp. 79–90, Nov. 2017.

[52] J. W. Durham, R. Carli, P. Frasca, and F. Bullo, "Discrete partitioning and coverage control for gossiping robots," *IEEE Trans. Robot.*, vol. 28, no. 2, pp. 364–378, Apr. 2012.

[53] L. Chen and E. Miller-Hooks, "Optimal team deployment in urban search and rescue," *Transp. Res. Part B, Methodol.*, vol. 46, no. 8, pp. 984–999, Sep. 2012.

ISHAQ ANSARI is currently pursuing the bachelor's degree in electrical and computer engineering (major) with Texas A&M University at Qatar. Before joining Texas A&M, he studied with the Computer Science Department, Carnegie Mellon University in Qatar, for two years. His academic interests include machine learning, deep learning, algorithm development, computer vision, and robotics. His educational background provides a strong foundation in both computer science and engineering principles, which are essential for his research interests.

ABUBAKR MOHAMMED received the B.Sc. degree in computer science from Carnegie Mellon University in Qatar, in 2023. From 2020 to 2023, he was a Course Assistant with Carnegie Mellon University. He was also a Computer Engineer with Rimads, in Summer 2022. He is currently with Starlink, Qatar, as an Associate Software Engineer. He has also pursued research in program verification, automated theorem provers, and the use of interactive provers in educational contexts. His research interests in robotics include multi-robot task allocation and mission planning.

YAQOOB ANSARI is currently pursuing the bachelor's degree in electrical and computer engineering with Texas A&M University in Qatar. Prior to his studies with Texas A&M, he was enrolled with the Computer Science Program, Carnegie Mellon University in Qatar, for two years. His academic pursuits are characterized by a strong emphasis on machine learning, deep learning, and computer vision. His research primarily explores innovative applications of these technologies within the field of engineering. The unique combination of his studies in both computer science and engineering provides a robust foundation for his research endeavors, positioning him well to contribute to advancements in interdisciplinary technology solutions.

MOHAMMED YUSUF ANSARI received the B.Sc. degree in computer science from Carnegie Mellon University and the M.Sc. degree in data science from Hamad Bin Khalifa University. He is currently pursuing the Ph.D. degree in computer engineering with Texas A&M University.

SAQUIB RAZAK received the bachelor's and master's degrees in electrical engineering from The University of Texas at Austin, and the Ph.D. degree in computer science from the State University of New York, Binghamton, in 2009. He has been a Teaching Faculty in computer science and engineering with the University of Michigan, Ann Arbor, since January 2023. Before joined to Michigan, he was an Associate Teaching Professor of computer science with Carnegie Mellon University in Qatar and the Director of HBJ Center for Computer Science Education in K-12. He was an Embedded Software Engineer with Motorola Inc., for eight years before moving to academia. His research interests include embedded systems, the Internet of Things, wireless networks, and computer science education.

EDUARDO FEO FLUSHING received the first M.Sc. degree in informatics from the University of Trento, Italy, in 2010, the second M.Sc. degree in software system engineering from RWTH Aachen University, Germany, in 2010, and the Ph.D. degree from the University of Lugano, Switzerland, in 2017. He is an Assistant Teaching Professor with Carnegie Mellon University in Qatar. From 2018 to 2021, he was a Postdoctoral Associate with the Department of Computer Science, Carnegie Mellon University in Qatar. From 2011 to 2017, he was with the Dalle Molle Institute for Artificial Intelligence (IDSIA), Switzerland. He is an Erasmus Mundus Master's Alumni. His research interests include integrating cyber-physical systems, artificial intelligence, wireless networking, and multi-robot systems to develop affordable solutions for societal challenges.