A Gossip-Based Approach for Measurement Task Allocation and Routing in Multi-Robot Systems with Heterogeneous Sensing

Hamza Chakraa, Diego Deplano, Carla Seatzu, Dimitri Lefebvre and Mauro Franceschelli

***Abstract*— This paper presents a decentralized task allocation strategy for heterogeneous multi-robot systems to minimize makespan during mission execution. The approach leverages a Gossip-based consensus mechanism, where robots communicate and exchange task information to optimize task distribution. The problem is modelled as a Multi-Robot Task Allocation (MRTA) challenge with the objective of minimizing task completion time (makespan). The proposed heuristic algorithm operates by iteratively improving task sequences via local exchanges between robots. Simulations demonstrate the algorithm's effectiveness in assigning tasks while considering various robot capabilities and environmental constraints, resulting in improved mission performance and reduced overall task completion time.**

***Index Terms*— Multi-Robot System (MRS), Task Allocation, Gossip Algorithm, Decentralized Optimization.**

I. Introduction

This paper explores inspection planning in industrial settings using a Multi-Robot System (MRS), with particular attention to high-risk areas where hazardous materials are produced, handled, or stored. Monitoring these environments plays a key role in preventing accidents and ensuring the safety and security of operations. Multiple autonomous robots are favored over manned systems due to their cost-effectiveness. While autonomous inspection technology is still advancing, strategic planning is needed to maximize the efficient use of multiple vehicles and keep inspection costs low. Coordinating Unmanned Ground Vehicles (UGVs), Unmanned Aerial Vehicles (UAVs), and even Unmanned Surface Vehicles (USVs) presents an effective solution for collecting inspection data through their integrated sensors.

In practical operations, missions are often planned manually by experts. To streamline this process, it is essential to equip robots with decision-making algorithms that efficiently distribute tasks. These algorithms must also determine the sequence of tasks for each robot, the type of information to be gathered, and the order in which tasks should be completed. Some areas in the environment may require multiple tasks and thus multiple robots, while others, which are less critical, may require fewer. The solution involves assigning specific target locations to each robot and arranging their tasks into a sequence analogous to the Traveling Salesman Problem (TSP). Each robot performs a subset of measurement tasks within these locations, ensuring that the team completes the entire mission across all locations at the lowest possible cost. Moreover, the coordination can be either centralized or decentralized.

Multi-robot patrolling problems have been extensively explored in the literature [1], [2], which has led to the investigation of Multi-Robot Task Allocation (MRTA) problems with scheduling [3], [4]. MRTA focuses on coordinating robots to perform multiple tasks with the goal of optimizing a specific objective function. Various researchers have introduced strategies and taxonomies to assist in addressing MRTA. Gerkey and Matarić [5] introduced a taxonomy to classify these problems based on robot capabilities, task demands, and time constraints. The term *Single-Task robots (ST)* refers to scenarios in which robots are limited to executing one task at a time, while *Multi-Task robots (MT)* denotes robots capable of handling multiple tasks simultaneously. Tasks can be either *Single-Robot tasks (SR)* or *Multi-Robot tasks (MR)*, depending on the number of robots required for completion. In *Instantaneous Assignment (IA)*, robots are assigned tasks one at a time, with no future planning involved, while *Time-extended Assignment (TA)* involves allocating a sequence of tasks to each robot over a defined planning period.

This paper focuses on missions involving multiple mobile robots, each outfitted with sensors to take measurements at various locations throughout an industrial setting. It is important to highlight that a robot can collect multiple measurements at a single location if it has the necessary sensors. Designing missions for teams of robots that leverage their combined capabilities introduces new limitations and complexities not considered in previous models. In previous studies, the authors focused on centralized methods [6], [7] based on Genetic Algorithms (GA) to establish a foundational ground for the system under a single decision maker. Centralized approaches are often easier to implement, allowing for efficient coordination and management. However, for scalability, decentralized mechanisms are more suitable. In this work, we shift our focus to decentralized mechanisms to address these limitations, providing greater flexibility.

The rest of the paper is organized as follows. A summary of the related work on decentralized MRTA planning is presented in the next section. Then, the studied problem is formalized in Sections III and IV. Details about the

Hamza Chakraa is with Université Le Havre Normandie, Normandie Univ, GREAH UR 3220, F-76600 Le Havre, France, and also with the DIEE, University of Cagliari, 09124 Cagliari, Italy. Email: hamza.chakraa@gmail.com

Diego Deplano, Carla Seatzu and Mauro Franceschelli are with the DIEE, University of Cagliari, 09123 Cagliari, Italy. Emails: {diego.deplano, mauro.franceschelli, carla.seatzu}@unica.it

Dimitri Lefebvre is with Université Le Havre Normandie, Normandie Univ, GREAH UR 3220, F-76600 Le Havre, France. Email: dimitri.lefebvre@univ-lehavre.fr

Open-source code is available online at: https://github.com/chakraah/gossip-planner

Gossip algorithm are provided in Section V. The results are presented and discussed in Section VI through a scalability analysis. Finally, the conclusion and perspectives are presented in Section VII.

II. Literature review

The MRTA problem can be solved using one of three main strategies: market-based, optimization-based, or behavior-based approaches. It has been shown that optimization-based approaches are the most effective [4]. For optimization-based approaches, there are two main categories: exact approaches and approximate approaches. Approximate methods have been adopted in the MRTA literature because of their reasonable computational complexity for large-scale problems.

Decentralized methods have become increasingly popular in this field due to their scalability benefits compared to centralized approaches. Studies indicate a growing preference for decentralized approaches in multi-robot coordination [4]. These methods operate based on predefined rules, shared knowledge, or simulated interactions embedded in the algorithm. For instance, many decentralized algorithms employ consensus algorithms, in which each robot makes decisions based on local information or predictive models of the actions of other robots.

Shorinwa et al. [8] presented a distributed algorithm that uses consensus to solve a MRTA problem without a central unit, enabling optimization through local communication with neighbors. Their approach includes three algorithms that balance computational efficiency and communication requirements, achieving faster and optimal solutions for linear and convex problems, as demonstrated in surveillance tasks. Additionally, Kalempa et al. [9] introduced a consensus-driven fault-resilient scheduling mechanism for a MRS in smart factories. Their method employs a hierarchical decision-making model with fuzzy controllers to ensure task reallocation and system robustness in the event of robot failures. The approach was tested in virtual warehouse environments. O'Brien et al. [10] presented a MRTA system for coordinated exploration in underground environments. Their approach uses a consensus-based auction protocol for dynamic task allocation, enabling robots to autonomously explore and adapt to changing conditions while reducing communication breakdowns. Furthermore, Mahato et al. [11] introduced a strategy that relies on synchronous transmission interactions to address issues arising from unstable or unavailable network infrastructures. This strategy aims to achieve efficient consensus while limiting the exchange of information during the distributed task allocation process.

In addressing the problem presented in this paper, authors in [12] proposed a consensus-based approach that effectively integrates both ascending and descending consensus strategies for task assignment. Moreover, they employed a 2-opt local search method to determine the routing.

Based on this literature review, we propose adopting a Gossip-based algorithm (a decentralized consensus method) [13], [14], [15], [16] to improve scalability and compared to [12]. While [15], [16] address other application domains, our variant explicitly incorporates heterogeneity in both robot capabilities and task types. In contrast to [12], which depends on structured consensus steps, we leverage pairwise gossip exchanges to achieve flexible and efficient task redistribution, combined with 2-opt routing for local path optimization.

III. Problem statement

The problem addressed in this paper involves heterogeneous robots equipped with different types of sensors that must perform various tasks subject to different requirements and constraints. The MRTA problem considered here is to assign a set of measurement tasks to a set of robots in a way that yields good performance with respect to the objective function (minimizing the cumulative energy consumption or time). Note that a task is defined as a pair consisting of a specific location and a given measurement, so a single robot is responsible for completing a measurement operation. Additionally, a robot can carry out one or more activities simultaneously. So, this study is classified as MT-SR-TA [5].

The mission environment is represented by a grid-based map forming a two-dimensional mesh of size $(N_x \times N_y)$, over which R mobile robots $\mathcal{R} = \{r_1, \dots, r_k, \dots, r_R\}$ navigate (a 2D representation is employed, assuming that UAVs maintain a constant flight level). The system consists of V cells, identified by the set $\mathcal{V} = \{v_1, \dots, v_j, \dots, v_V\}$ with central coordinates (x_j, y_j). These cells have specific dimensions based on the problem's needs. Thus, the environment is represented as a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E}$ is the set of edges connecting the cells. The weight function $W : \mathcal{E} \times \mathcal{R} \to \mathbb{R}^+$ assigns a positive weight to each edge for a given robot. Every edge $e \in \mathcal{E}$ connects two vertices $v_j, v_{j'} \in \mathcal{V}$ for a given robot r_k and has an associated elementary cost $W((j, j'), k) = \tilde{c}(j, j', k) > 0$. It is essential to note that moving directly from v_j to $v_{j'}$ is not possible if the cells are not adjacent, in which case $\tilde{c}(j, j', k) = \infty$. This representation provides a practical approach to describing the state space, enabling the calculation of travel costs between positions while accounting for obstacles.

Multiple measurements must be performed at specific locations, referred to as "sites," within the environment. The set of sites to be visited is represented as $\mathcal{A} = \{a_1, \dots, a_i, \dots, a_A\}$, where $\mathcal{A} \subseteq \mathcal{V}$ and A denotes the number of sites. The robots begin and end their tour at the common site a_1 (the depot), where no specific assignment is required. The set $\mathcal{M} = \{m_1, \dots, m_q, \dots, m_M\}$ denotes the range of measurement types, which likely corresponds to the diverse data or information the robots are required to collect during their mission. The set of tasks that the robots must perform to complete the mission is defined as $\mathcal{J} = \{j_1, \dots, j_t, \dots, j_J\}$, where J represents the total number of tasks. In this context, a task $j_t = (i, q)$ is defined as a measurement of type $m_q \in \mathcal{M}$ to be performed at a specified site $a_i \in \mathcal{A}$. Additionally, we define the function $\mathcal{T} : \mathcal{A} \times \mathcal{M} \to \{0, 1\}$ as follows:

$$t_{i,q} = \begin{cases} 1, & \text{if } (i, q) \in \mathcal{J}, \\ 0, & \text{otherwise.} \end{cases}$$

As mentioned earlier, the mission considers a group of R mobile robots equipped with sensors for task execution. Each sensor is designed for a specific measurement, thereby forming the set $\mathcal{M}$ of all sensor types. Each robot $r_k \in \mathcal{R}$ is equipped with a set of sensors $\mathcal{M}_k \subseteq \mathcal{M}$. We define the function $\mathcal{P} : \mathcal{R} \times \mathcal{M} \rightarrow \{0, 1\}$ as follows:

$$p_{k,q} = \begin{cases} 1, & \text{if } m_q \in \mathcal{M}_k, \\ 0, & \text{otherwise.} \end{cases}$$

The movement cost between two given sites a_i and a_j for a robot r_k is defined as an elementary cost $c(i, j, k)$, which can be interpreted as the time required by r_k to move from a_i to a_j. This cost is determined using Dijkstra's algorithm, which computes the shortest path between every pair of sites based on the previously defined elementary costs between the cells in the graph. The objective of this optimization is to minimize the makespan function by organising the sequences of tasks, represented as $\mathcal{S}(k)$ for each robot r_k.

Definition 1. A MinMax (makespan) function, which optimizes the highest individual cost among the team of robots, $\mathcal{C} : \{\mathcal{S}(k), r_k \in \mathcal{R}\} \rightarrow \mathbb{R}^+$, is defined as:

$$\mathcal{C}_{max} = \max_{r_k \in \mathcal{R}} \mathcal{C}(\mathcal{S}(k)) \tag{1}$$

The problem can be formalized using Mixed Integer Linear Programming (MILP) as detailed in Equations (2) to (7). The boolean decision variable $x^k_{i,j}$ is defined as:

$$x^k_{i,j} = \begin{cases} 1, & \text{if } r_k \in \mathcal{R} \text{ travels from } a_i \in \mathcal{A} \text{ to } a_j \in \mathcal{A}, \\ 0, & \text{otherwise.} \end{cases}$$

Furthermore, the integer decision variable $u^k_i \in \mathbb{N}^+$ determines the order in which robot r_k visits site a_i within its assigned mission.

$$\min \quad \max_{r_k \in \mathcal{R}} \sum_{a_i, a_j \in \mathcal{A}} x^k_{i,j} \times c(i, j, k) \tag{2}$$

$$\text{s.t.} \quad \sum_{a_i \in \mathcal{A}} x^k_{1,i} = 1 \quad \forall r_k \in \mathcal{R} \tag{3}$$

$$\sum_{a_j \in \mathcal{A}} x^k_{i,j} = \sum_{a_j \in \mathcal{A}} x^k_{j,i} \quad \forall a_i \in \mathcal{A}, \forall r_k \in \mathcal{R} \tag{4}$$

$$u^k_i + x^k_{i,j} \leq u^k_j + (A - 1) \times (1 - x^k_{i,j}), \quad \forall a_i \in \mathcal{A}, \forall a_j \in \mathcal{A} \setminus \{a_1\}, \forall r_k \in \mathcal{R} \tag{5}$$

$$\sum_{a_i, a_j \in \mathcal{A}} x^k_{i,j} \times c(i, j, k) \leq \mathcal{B}_k \quad \forall r_k \in \mathcal{R} \tag{6}$$

$$\sum_{r_k \in \mathcal{R}} \sum_{a_i \in \mathcal{A}} p_{k,q} \times x^k_{i,j} \geq t_{j,q}, \quad \forall m_q \in \mathcal{M}, \forall a_j \in \mathcal{A} \tag{7}$$

Equation (2) defines the cost function, which aims to minimize the highest cost among the robots in the MinMax optimization. Equation (3) specifies that each robot must start the mission from the depot a_1. Equation (4) mandates that every entry into a site must correspond to an exit, ensuring a return to the depot. Equation (5) eliminates any subtours, thereby ensuring the formation of directed cycles within each robot's sequence [17]. Additionally, to address the autonomy limitations of the robots, Constraint (6) requires that the total mission cost for each robot r_k does not exceed $\mathcal{B}_k$. Lastly, Constraint (7) guarantees that every task is completed by a robot equipped with the appropriate sensor.

IV. Bounding the Optimal Solution

In this section, we establish both lower and upper bounds for the optimal solution of the problem, denoted as C^*. These bounds help in evaluating the efficiency of the heuristic considered later in this paper by providing a constant-factor approximation to the optimal solution.

A. Lower Bound: LP Relaxation

Definition 2. The optimal solution C^* is lower bounded by solving the LP relaxation of the MILP formulation. This relaxation provides an efficient means of estimating the lower bound on C^* [17].

The MILP formulation presented in the previous section includes binary and integer decision variables $x^k_{i,j}$ and u^k_i, which define the sequencing of robots in the scheduling framework. To obtain a relaxed version of the problem, we allow these variables to take continuous values within their respective domains, leading to an LP relaxation:

$$0 \leq x^k_{i,j} \leq 1, \quad \forall r_k \in \mathcal{R}, \forall a_i, a_j \in \mathcal{A} \tag{8}$$

$$u^k_i \in \mathbb{R}^+, \quad \forall r_k \in \mathcal{R}, \forall a_i \in \mathcal{A} \tag{9}$$

By relaxing these variables, the feasible solution space is expanded, allowing fractional values for decision variables. Consequently, the LP relaxation provides an objective value, denoted as C_{LB}, which serves as a lower bound on the optimal makespan C^* of the original MILP problem. Although the LP relaxation ignores integer constraints and the solution obtained may not be feasible for the original problem, it still offers a valuable benchmark for evaluating heuristics.

B. Upper Bound Computation via Monte Carlo Simulations

Definition 3. The optimal solution C^* can be upper bounded by employing heuristic-based simulations such as Monte Carlo methods. These approaches enable the estimation of C_{UB} by generating feasible solutions through randomized task assignments and travel sequences.

Monte Carlo simulations offer a practical approach for constructing upper bounds by iteratively producing potential solutions and evaluating their makespan values. The general procedure for estimating an upper bound using Monte Carlo simulations is as follows:

1) Generate multiple feasible solutions by randomly assigning tasks to robots across several independent simulation runs.
2) Evaluate the makespan for each generated solution, ensuring that all constraints are met.
3) Identify the minimum makespan value observed across all simulated scenarios, which serves as the estimated upper bound C_{UB}.

The effectiveness of this approach depends on the number of iterations performed. By increasing the number of Monte

Carlo trials, the probability of obtaining a reliable upper bound improves, thereby reducing the gap between C_{UB} and the true optimal value C^*. Comparing C_{UB} with the LP-based lower bound C_{LB} provides a measure of solution quality and indicates the need for further refinements in problem-solving approaches.

V. Gossip-based heuristic approach

A. Algorithm strategy and workflow

This section introduces the decentralized Gossip Heuristic algorithm for solving the task allocation problem outlined in the previous section. The Gossip algorithm is a decentralized communication protocol commonly used in distributed systems. In this protocol, nodes (or processes) periodically exchange information with randomly selected peers, enabling the spread of data such as updates, state information, or failure reports across the network. In the context of our robotic problem, the nodes represent the robots, and the data consists of the tasks to be assigned. Each time two robots engage in Gossip, they exchange information about tasks, potentially redistributing them for better allocation.

Algorithm 1: Decentralized Gossip Heuristic

```
Compute an initial feasible sequence of tasks for
  each robot.
Compute a list of possible robot pairs for task
  exchange, where each pair has at least one common
  sensor: possible_pairs.
F = 1
while F = 1 do
    F = 0
    Shuffle the order of possible_pairs randomly.
    foreach pair in possible_pairs do
        (a) Apply Algorithm 2 on the current pair;
        (b) If the solution is improved, set F = 1
```

In the decentralized Gossip Heuristic (Algorithm 1), robots update their assignment based on the task exchange mechanism presented in Algorithm 2. It is designed to optimize task allocation for the problem at hand in a decentralized manner. Initially, each robot computes a feasible sequence of tasks it can perform. A list of possible robot pairs, where each pair shares at least one common sensor, is generated for potential task exchanges. The algorithm operates in a loop where each iteration involves randomly shuffling the order of possible pairs and sequentially applying the task exchange mechanism (Algorithm 2) to each pair. Unlike randomized gossip protocols that typically rely on local, probabilistic peer interactions, our algorithm systematically evaluates all possible pairs of compatible robots in each iteration. This strategy ensures broader task exchange opportunities across the team, increasing the likelihood of improving task sequences. By combining deterministic coverage with randomized order, the method enables an efficient optimization of assignments.

Algorithm 2: Task exchange mechanism between robots r_k and r_q for MinMax minimization

Input: Task sequences $\mathcal{S}(k)$, $\mathcal{S}(q)$ for robots r_k, r_q

Assumption: $\mathcal{C}(\mathcal{S}(q)) < \mathcal{C}(\mathcal{S}(k))$

Let $\mathcal{J}_{ex}$ be the set of tasks assigned to robot r_k that robot r_q is capable of performing.

while $\mathcal{J}_{ex} \neq \emptyset$ **do**

- (a) Randomly select a task $t_j \in \mathcal{J}_{ex}$;
- (b) Update the set of exchangeable tasks: $\mathcal{J}_{ex} = \mathcal{J}_{ex} \setminus \{t_j\}$;
- (c) Add t_j to the sequence of robot r_q: set $\mathcal{S}_{new}(q) = 2\text{-}opt\big(\mathcal{S}(q) \cup \{t_j\}\big)$;
- (d) **if** $\mathcal{C}(\mathcal{S}_{new}(q)) < \mathcal{C}(\mathcal{S}(k))$ **then**
    - Update the sequences: set $\mathcal{S}(q) = \mathcal{S}_{new}(q)$ and $\mathcal{S}(k) = \mathcal{S}(k) \setminus \{t_j\}$.

Algorithm 2 describes the protocol for exchanging tasks between two robots r_k and r_q. Initially, the algorithm takes as input the sequences of tasks currently assigned to each robot in the pair. Then, a set of exchangeable tasks $\mathcal{J}_{ex}$ is identified. This set represents the tasks that both robots can perform and are assigned to the robot with the higher cost sequence (assumed to be r_k). Afterwards, the algorithm proceeds as long as there are tasks available in the exchangeable tasks set. Within each step, a task is randomly selected from this set and added to the sequence of the robot that has the lower cost. This addition requires determining the best position at which to insert t_j in the sequence; hence, the 2-opt algorithm is applied at this stage (2-opt is an optimization technique typically employed to improve routes by removing and replacing segments of a path that intersect) [7]. After updating the task sequence for the robot with the lower cost, the algorithm checks whether the cost of this newly updated sequence is still lower than the cost of the original task sequence of the other robot. If this condition is satisfied, it indicates that the task exchange has resulted in a more efficient (or, in the worst case, equivalent) allocation of tasks. Consequently, the task sequences are updated: the new task sequence for robot r_q becomes the updated sequence, while robot r_k removes the exchanged task from its own sequence. This iterative process continues until there are no more tasks left in the set $\mathcal{J}_{ex}$.

The decentralized Gossip Heuristic algorithm is a fast and highly efficient approach for task allocation in robotic networks. By systematically exploring all possible task exchanges among robot pairs, it rapidly converges to high-quality solutions while significantly reducing the likelihood of being trapped in local optima. This ensures enhanced overall performance. Moreover, its architecture enables good scalability, making it ideal for large-scale systems.

B. Characterization of the Gossip-based heuristic

Proposition 1. The decentralized Gossip Heuristic algorithm terminates in a finite number of iterations.

Proof. Let $\mathcal{C}_{max}(t)$ denote the makespan at iteration t. The algorithm improves the solution if and only if:

$$\mathcal{C}_{max}(t+1) < \mathcal{C}_{max}(t)$$

or retains the current solution if:

$$\mathcal{C}_{max}(t+1) \geq \mathcal{C}_{max}(t)$$

Since each task exchange either reduces or maintains the makespan, the sequence is non-increasing:

$$\mathcal{C}_{max}(1) \geq \mathcal{C}_{max}(2) \geq \mathcal{C}_{max}(3) \geq \dots$$

The number of possible configurations is finite due to the finite number of tasks and robots. Since the algorithm strictly improves or retains the makespan without revisiting worse configurations, it must eventually reach a state where no better assignment is possible. Thus, the algorithm terminates in a finite number of iterations. $\square$

Proposition 2. The time complexity of one iteration of Algorithm 1 is given by:

$$\mathcal{O}\left(J \cdot R^2\right)$$

Proof. At each iteration, the algorithm evaluates a maximum of $\frac{R(R-1)}{2}$ pairs of robots, which amount to:

$$\frac{R(R-1)}{2} = \mathcal{O}(R^2)$$

For each pair, the algorithm considers up to J task exchanges. The task exchanges involve updating task sequences and recalculating costs, which have a constant complexity per operation. Hence, the complexity per iteration is[1]:

$$\mathcal{O}(J \cdot R^2). \ \square$$

The next section presents simulation-based validation, demonstrating the algorithm's ability to maintain scalability and performance in extensive robot-task networks.

VI. Numerical simulations and analysis

In this section, we present numerical results to evaluate the performance of the proposed algorithm. The analysis is conducted across multiple scenarios involving varying numbers of robots and tasks to assess the algorithm's effectiveness in solving the optimization problem. The evaluation focuses on two key aspects:

- **Solution quality**: The obtained results are compared with both the upper and lower bounds of the optimal value to understand how closely the algorithm approximates the optimal solution.
- **Convergence behaviour**: The algorithm's convergence is analyzed in terms of the number of iterations required and the execution time across different problem instances.

[1]2-opt is excluded from the complexity analysis for simplicity and due to its non-deterministic number of applications per iteration.

A. Comparison with Upper and Lower Bounds

Figure 1 presents the comparison between three key cost values: C_{GO}, which corresponds to the solution obtained by executing Algorithm 1; C_{LB}, the lower bound derived by relaxing the MILP formulation (Section IV); and C_{UB}, the upper bound estimated through Monte Carlo simulations (Section IV).

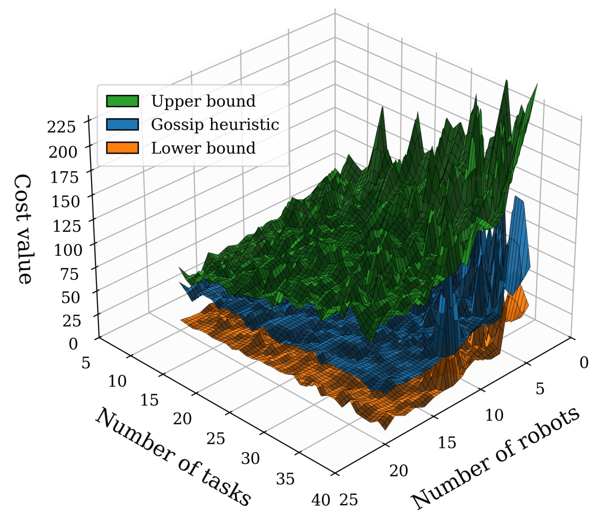

Fig. 1. Comparison of the cost value C_{GO} with the upper bound C_{UB} and lower bound C_{LB} across different scenarios

For each scenario, the values of C_{GO} and C_{UB} are computed as the mean over 10 independent experimental runs to account for stochastic variations and ensure statistical robustness. The results consistently indicate that the cost value obtained through the Gossip-based heuristic (C_{GO}) always lies between the upper and lower bounds of the optimal solution. This confirms that the proposed algorithm provides a good quality solution.

B. Convergence analysis

To further evaluate the algorithm's efficiency, we examine its convergence properties by analyzing two factors: execution time and number of iterations. We investigate how these factors vary with the different scenarios considered.

Figure 2 shows the execution time required to reach a solution by the Gossip-based heuristic for each of the scenarios considered in Figure 1. We can see that the algorithm is very fast, and the execution time does not exceed 0.4 s for all instances (the method scales well with problem size). However, the execution time approaches its maximum value in instances with a few robots and a large number of tasks. This is primarily due to large-size routing processes with the 2-opt method.

Figure 3 illustrates the number of iterations required for the Gossip-based heuristic to reach equilibrium across different scenarios. The observed trends indicate that the number of iterations depends on the balance between the number of robots and tasks. Specifically, scenarios with a larger number of tasks relative to robots tend to require more iterations before convergence.

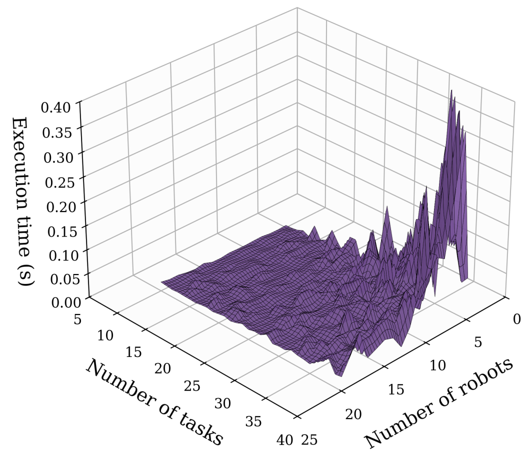

Fig. 2. Gossip heuristic execution time across several scenarios

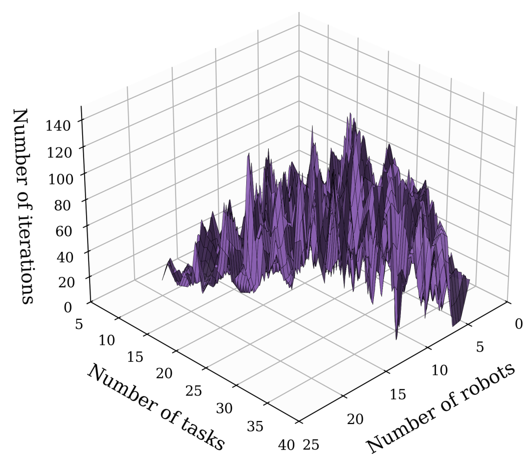

Fig. 3. Number of iterations required for the Gossip-based heuristic to reach equilibrium across several scenarios

Overall, these findings underscore the effectiveness of the proposed approach in addressing large-scale MRTA problems while overcoming scalability challenges often encountered in centralized methods [6], [7].

VII. Conclusion

In this paper, we introduced a decentralized task allocation approach for heterogeneous MRS based on a Gossip-based consensus mechanism. The proposed algorithm optimizes the makespan by iteratively improving task distribution through local exchanges between robots. The simulations demonstrated that our heuristic effectively balances task execution while maintaining a close approximation to optimal solutions with fast computational time.

Looking ahead, the computation of an upper bound on the number of iterations of Algorithm 1 should be investigated. Moreover, we are interested in specific applications like precision farming [13], [14], particularly within open networks where robots can freely join or leave (whether by choice or due to physical constraints) and can execute distributed algorithms to infer global information about the robot fleet [18], [19], [20].

References

[1] David Portugal and Rui Rocha. A survey on multi-robot patrolling algorithms. In Luis M. Camarinha-Matos, editor, *Technological Innovation for Sustainability*, volume 349, pages 139–146, 2011.

[2] Diego Deplano, Mauro Franceschelli, Simon Ware, Su Rong, and Alessandro Giua. A discrete event formulation for multi-robot collision avoidance on pre-planned trajectories. *IEEE Access*, 8:92637–92646, 2020.

[3] Alaa Khamis, Ahmed Hussein, and Ahmed Elmogy. Multi-robot Task Allocation: A Review of the State-of-the-Art. In *Cooperative Robots and Sensor Networks 2015*, volume 604, pages 31–51. 2015.

[4] Hamza Chakraa, François Guérin, Edouard Leclercq, and Dimitri Lefebvre. Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art. *Robotics and Autonomous Systems*, 168:104492, 2023.

[5] Brian P. Gerkey and Maja J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *The International Journal of Robotics Research*, 23(9):939–954, 2004.

[6] Hamza Chakraa, Edouard Leclercq, François Guérin, and Dimitri Lefebvre. A centralized task allocation algorithm for a multi-robot inspection mission with sensing specifications. *IEEE Access*, 11:99935–99949, 2023.

[7] Hamza Chakraa, Edouard Leclercq, François Guérin, and Dimitri Lefebvre. A multi-robot mission planner by means of beam search approach and 2-opt local search. In *2023 9th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2023.

[8] Ola Shorinwa, Ravi N. Haksar, Patrick Washington, and Mac Schwager. Distributed multirobot task assignment via consensus admm. *IEEE Transactions on Robotics*, 39(3):1781–1800, 2023.

[9] Vivian Cremer Kalempa, Luis Piardi, Marcelo Limeira, and Andre Schneider de Oliveira. Multi-robot task scheduling for consensus-based fault-resilient intelligent behavior in smart factories. *Machines*, 11(4):431, 2023.

[10] Matthew O'Brien, Jason Williams, Shengkang Chen, Alex Pitt, Ronald Arkin, and Navinda Kottege. Dynamic task allocation approaches for coordinated exploration of subterranean environments. *Autonomous Robots*, 47(8):1559–1577, 2023.

[11] Prabhat Mahato, Sudipta Saha, Chayan Sarkar, and Md. Shaghil. Consensus-based fast and energy-efficient multi-robot task allocation. *Robotics and Autonomous Systems*, 159:104270, 2023.

[12] Dimitri Lefebvre, Isabel Demongodin, Rabah Ammour, Sara Hsaini, and Moulay El Hassan Charaf. Consensus approach based on negotiation and 2-opt optimization for mrta problems in a decentralized setting. In *2024 10th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 32, pages 964–969, 2024.

[13] Mauro Franceschelli, Alessandro Giua, and Carla Seatzu. Fast discrete consensus based on gossip for makespan minimization in networked systems. *Automatica*, 56:60 – 69, 2015.

[14] Mauro Franceschelli, Alessandro Giua, and Carla Seatzu. A gossip-based algorithm for discrete consensus over heterogeneous networks. *IEEE Transactions on Automatic Control*, 55(5):1244 – 1249, 2010.

[15] Mauro Franceschelli, Daniele Rosa, Carla Seatzu, and Francesco Bullo. Gossip algorithms for heterogeneous multi-vehicle routing problems. *Nonlinear Analysis: Hybrid Systems*, 10:156–174, 2013.

[16] Diego Deplano, Carla Seatzu, and Mauro Franceschelli. A distributed online heuristic for a large-scale workforce task assignment and multi-vehicle routing problem. In *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*, pages 3174–3180, 2024.

[17] Tolga Bektas. The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3):209–219, 2006.

[18] Diego Deplano, Nicola Bastianello, Mauro Franceschelli, and Karl H. Johansson. A unified approach to solve the dynamic consensus on the average, maximum, and median values with linear convergence. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 6442–6448, 2023.

[19] Nicola Bastianello, Diego Deplano, Mauro Franceschelli, and Karl H. Johansson. Robust online learning over networks. *IEEE Transactions on Automatic Control*, 70(2):933–946, 2025.

[20] Diego Deplano, Nicola Bastianello, Mauro Franceschelli, and Karl H Johansson. Optimization and learning in open multi-agent systems. *arXiv preprint arXiv:2501.16847*, 2025.