A New Method for Improving the Fairness of Multi-Robot Task Allocation by Balancing the Distribution of Tasks

Youssef Msala [1,2*], Oussama Hamed [2], Mohamed Talea [2], Mohamed Aboulfatah [1]
[1] MISI Laboratory FST, University of Hassan I, Settat, Morocco
[2] TI Laboratory FSBM, University of Hassan II, Casablanca, Morocco
Email: [1] youssef.msala@gmail.com
*Corresponding Author

***Abstract*—This paper presents an innovative task allocation method for multi-robot systems that aims to optimize task distribution while taking into account various performance metrics such as efficiency, speed, and cost. Contrary to conventional approaches, the proposed method takes a comprehensive approach to initialization by integrating the K-means clustering algorithm, the Hungarian method for solving the assignment problem, and a genetic algorithm specifically adapted for Open Loop Travel Sales Man Problem (OLTSP). This synergistic combination allows for a more robust initialization, effectively grouping similar tasks and robots, and laying a strong foundation for the subsequent optimization process. The suggested method is flexible enough to handle a variety of situations, including Multi-Robot System (MRS) with robots that have unique capabilities and tasks of varying difficulty. The method provides a more adaptable and flexible solution than traditional algorithms, which might not be able to adequately address these variations because of the heterogeneity of the robots and the complexity of the tasks. Additionally, ensuring optimal task allocation is a key component of the suggested method. The method efficiently determines the best task assignments for robots through the use of a systematic optimization approach, thereby reducing the overall cost and time needed to complete all tasks. This contrasts with some existing methods that might not ensure optimality or might have limitations in their ability to handle a variety of scenarios. Extensive simulation experiments and numerical evaluations are carried out to validate the method's efficiency. The extensive validation process verifies the suggested approach's dependability and efficiency, giving confidence in its practical applicability.**

***Keywords*—balanced task allocation; heterogeneous agents, multi robot system; path planning; task decomposition.**

I. Introduction

Task allocation is an important aspect of multi-robot systems, where multiple robots are working in one team to achieve a shared goal [1]. The process of task allocation has to be optimal in order to achieve a high level of efficiency and effectiveness of the multi-robot system [2]–[4]. In other words, the tasks must be distributed among all the robots in the MRS in a way that maximizes the efficiency, minimizes the completion time, and optimizes the overall performance of the robots' system [5], [6]. Search and rescue [7]–[9] hunting operations [10]–[12] or transportation and logistics [13], and other fields could all be revolutionized by effective task allocation.

The multi-robot system can improve efficiency, reduce completion times, and achieve higher overall performance by allocating tasks in the best way possible. Take into account a search and rescue scenario in which various robot types work together to find survivors in a disaster-stricken area. Task distribution that is optimised can guarantee prompt rescue operations, significantly cut down on search time, and distribute tasks based on each robot's capabilities.

The presence of heterogeneous robots introduces a layer of complexity to the task allocation process within multi-robot systems. Each robot possesses distinct capabilities, spanning varying levels of performance, mobility, and sensory abilities. Effectively allocating tasks requires a nuanced understanding of these discrepancies, ensuring tasks are matched to robots equipped to handle them optimally. The challenge lies in creating an allocation strategy that not only maximizes overall system efficiency but also respects the unique strengths and limitations of individual robots. Balancing the distribution of tasks among heterogeneous robots demands a sophisticated approach that leverages their diversity to achieve collective excellence.

The architecture chosen for multi-robot systems significantly influences the success of task allocation strategies. An applicable architecture should provide a robust framework for seamless communication, coordination, and real-time information exchange among robots. This architecture should foster efficient decision-making by enabling the aggregation and analysis of data from various sources. It should accommodate the complexities introduced by heterogeneous robots, allowing for adaptive task allocation algorithms that consider both the characteristics of tasks and the capabilities of individual robots. Moreover, the architecture should facilitate the integration of dynamic factors, such as changing task priorities or environmental conditions, into the allocation process. A well-designed architecture enhances the adaptability and scalability of the system, enabling efficient and effective task allocation while promoting harmonious collaboration among the diverse robotic entities.

Creating a task allocation strategy that can handle the diversity of tasks and robots is an important challenge; which ensures a seamless distribution of capabilities and resources. Considerable advancements have been attained in multiple task allocation methodologies that derive from diverse approaches [14] such as centralized approaches [15], decentralized approaches [16], cloud approaches [17] and market-based approaches [18]. These methods do, however, have some drawbacks that highlight the demand for more development.

In methods based on centralized approach [19], [20], there is a central unity that is responsible to allocate tasks to the group of robots, where it assigns the tasks to the available robots with a defined criterion [21], [22]. Using this approach, the allocation is more efficient and optimum than the decentralized approaches, where the centralized computing unit has an overall overview about the robot properties and tasks [23], [24]. However, these methods depend mainly on the centralized agent where it can result in delays, inefficiencies when this agent is not available or not well-informed [25], [26]. Methods of task allocation based on decentralized approaches [27], [28] relies on making decisions at the level of each robot [29], [30], where these robots are responsible to choose tasks to perform based on their criteria [31], [32]. Therefore, these methods are more resilient because they are not based on a centralized agent for decision-making [33], [34]. However, the robots are prone to have conflicts or misunderstanding when they have different preferences or interpretations of the task allocation process [26], [35].

Market-based task allocation strategies in multi-robot systems are inspired by economic markets [36]–[38]. Robots bid on tasks as commodities, with allocation governed by market dynamics like supply, demand, price, and competition [39]–[41]. Robots autonomously select tasks based on their abilities and preferences, though these strategies are more complex than centralized or decentralized methods [42]. The success of such approaches hinges on the honesty of robot bids, as deceptive bids can lead to inefficiencies [43]–[45].

Despite the fact that the field of task allocation in MRS has made significant progress, there is still a significant research gap when it comes to addressing the issues presented by heterogeneous tasks and minimising travel expenses within the framework of a centralised architecture. Current methods have made progress but struggle to achieve optimal allocation in scenarios where factors diverge significantly.

Centralized approaches excel in coordination but may fail in dynamic scenarios or when real-time data is lacking. Decentralized approaches enhance resilience but can lead to conflicts due to divergent preferences. Market-based approaches offer autonomy but introduce complexities and susceptibility to dishonest bidding. Additionally, optimizing travel costs during task execution is a challenge, as they influence operational expenses, time efficiency, and resource consumption. It is still largely unknown how tasks should be distributed based on their varying levels of complexity and how far they must travel to be completed. An all-encompassing strategy is required, one that seeks to reduce cumulative travel distances in order to increase system efficiency while also taking into account the capabilities of individual robots and the characteristics of tasks. This can be accomplished by robots working together and planning strategically to use resources as efficiently as possible.

Currently, centralized approaches to task allocation for multi-robot systems have gained significant attention and been extensively researched [24], [46]–[58]. In order to resolve the cooperative task allocation problem of multi-robot systems, Wang et al proposed a unique multi-objective ant colony system (MOACS) strategy [59]. A unique solution construction approach and a novel pheromone updating rule are provided, both of which are based on the single objective ant colony system (ACS). Compared to the current Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Multi-Objective Particle Swarm Optimization, the proposed MOACS has shown superior performance and efficacy (MOPSO). In [60], the study examines how to carry out tasks that are distributed throughout the environment while reducing system costs as a whole. They provide a novel deployment-based framework that divides the problem into two subproblems: region partitioning and routing problem, in order to address the issue of multi-robot job allocation in very vast environments. A novel scheduling model is suggested that takes wait time into account [61]. The balanced heuristic mechanism (BHM) is used in this technique to select the best gathering station to reduce waiting time, and task rescheduling based on task correlation (TRBTC) is additionally utilized to mitigate travel costs.

This paper addresses the issue of task allocation in which the tasks and robots are heterogeneous and the overall cost of travel must be optimized. In other words, the tasks have various levels of difficulty, similar to robots, which have differing capacities. The tasks and robots are distributed within a confined workspace, and the robots are required to accomplish all feasible tasks. The cost is calculated based on the total distance travelled by each robot in the MRS, and the overall cost is determined by the highest distance travelled by any robot within the MRS. The proposed approach for minimizing the overall cost involves three stages. The first stage involves task assignment initialization, which directs the robots to their respective tasks based on distance and difficulty. In the second stage, pre-allocation is optimized by iteratively modifying the set of tasks assigned to each robot, resulting in reduced total costs. In the final stage, the set of tasks assigned to each robot is re-evaluated to identify any further opportunities for improvement.

This study focuses on creating a task allocation methodology that takes complexity, priority, and resource requirements into account for heterogeneous tasks in multi-robot systems. The objective is to maximise system performance while taking care of issues with various degrees of complexity and importance. By strategically allocating tasks based on spatial distribution, task requirements, and robot capabilities, the research aims to reduce travel expenses. Robot cumulative effort is reduced using this method, increasing productivity and resource use.

The following sections of this paper are structured as follows: section 2 introduces briefly background methods and presents the proposed method; section 3 gives the simulation test and verifies the feasibility of the proposed method. In section 4, conclusion and future works are discussed.

II. METHODS AND BACKGROUND

The initialization stage of the proposed method is constructed using three highly effective techniques that are widely adopted: K-means algorithm, Hungarian method, and Open-Loop TSP utilizing genetic algorithm. Each of these methods is briefly discussed.

A. K-means

The k-means algorithm [62]–[65] is one of the most used algorithms for clustering problems [66]–[68]. This algorithm takes a set of n data points and groups them into k clusters, where k is a pre-specified parameter. The k-means algorithm is used here to divide a set of n tasks into k robot clusters.

Given a set of points $(x_1, ..., x_n)$, the objective is to partition the n points into k sets:

$$S = S_1, S_2, ..., S_k, \quad (k \leqslant n) \tag{1}$$

by minimizing the distance between the points at inside each partition:

$$arg_S min \sum_{i=1}^{k} \sum_{x_j \in S_i} \parallel x_j - \mu_i \parallel^2 \tag{2}$$

where μ_i is the barycenter of the points in S_i

- Choose k points which represent the average position of the initial partitions $m_1^{(1)}, ..., m_k^{(1)}$
- Repeat until there is convergence :
  - Assign each observation to the nearest partition :

$$S_i^{(t)} = \left\{ x_j : \left\| x_j - m_i^{(t)} \right\| \leqslant \left\| x_j - m_{i^*}^{(t)} \right\| \forall i^* = 1, \ldots, k \right\} \tag{3}$$

  - Update the mean of each cluster :

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j \tag{4}$$

B. Hungarian method

The Hungarian Algorithm [69]–[71], also known as the Kuhn-Munkres Algorithm, is a combinatorial optimization algorithm that solves the assignment problem in polynomial [72]–[74]. The assignment problem is the problem of finding the optimal assignment of n tasks to n agents, given a cost matrix that indicates the cost of assigning each task to each agent [75]. The goal is to minimize the total cost of the assignment.

The Hungarian method is used to find the optimal assignment of robots to the set of tasks. The following steps explain the execution order of the Hungarian Algorithm.

Algorithm 1 Hungarian method

```
Input: Given a bipartite graph $G = (V = (X, Y), E)$, whose
partitions are $X$ and $Y$, with $E$ indicating the graph's edges
where $(|X| = |V| = n)$.
Output: A perfectly matched $M$

Description:
1: Perform initialization.
2: All numbers in each row are subtracted from the row's
   minimum number.
3: Based on the matrix obtained in the previous step, subtract
   all numbers in each column from the column's minimum
   number.
4: Use the fewest possible horizontal and vertical lines to
   cover all zeros in the resulting matrix. If n lines are needed,
   a best-fit assignment can be found from the zeros. The
   algorithm comes to a halt. Continue to Step 4 if less than
   n lines are necessary.
5: Create additional zeros:
     - Locate the smallest element (call it k) that is not
       covered by a line in Step 3.
     - Add k to all elements that are covered twice and
       subtract k from all uncovered
6: Return M
```

C. Open loop Travel salesman problem

The Traveling Salesman Problem (TSP) [76]–[78] is a classic problem in computer science, which seeks to find the shortest possible route that visits a given set of cities and returns to the starting point [79]. The open loop variant of TSP is a variant where the salesman does not need to return to the starting city. The problem can be formulated mathematically as follows:

Let G = (V, E) be a complete graph with $V = 1, 2, ..., n$ representing the set of cities and E representing the set of edges connecting the cities. The edge weight or distance between any two cities i and j is represented by $c_{(i,j)}$.

The objective is to find the shortest possible route that visits all cities exactly once and ends at a particular city. Let's assume that the salesman starts at city 1 and ends at city k. The problem can be formulated as a linear programming problem as follows:

$$Minimize\ Z = \sum_{(i,j)\in E} c_{(i,j)} x_{(i,j)} \tag{5}$$

subject to:

$$\sum_{(i,j)\in E} x_{(i,j)} = n-1 \tag{6}$$

(each city should be visited exactly once)

$$\sum_{(j\in V, j\neq i)} x_{(i,j)} = 1 \quad \forall i \in V,\ i \neq k \tag{7}$$

(exactly one outgoing edge from each city)

$$\sum_{j\in V,\ i\neq j} x_{(i,j)} = 1 \quad \forall i \in V,\ j \neq 1 \tag{8}$$

(exactly one incoming edge to each city)

$$\sum_{(i,j)\in S} x_{(i,j)} \leqslant |S|-1 \quad \forall S \subseteq V,\ 2 \in S, \notin\ kS \tag{9}$$

(subtour elimination constraint)

Here, $x_{(i,j)}$ is a binary decision variable that takes value 1 if the edge (i,j) is included in the tour, and 0 otherwise. This problem is NP-hard, which means that it is computationally infeasible to solve for large instances of the problem. Therefore, a variety of heuristic and approximation algorithms are often used to find near-optimal solutions.

The genetic algorithm is one of the popular metaheuristic approaches to solve the traveling salesman problem (TSP) inspired by the natural process of evolution. The algorithm works by evolving a population of candidate solutions over multiple generations and selecting the best solution as the output. The following steps explain the execution order:

- Initialization: Generate an initial population of candidate solutions (Each solution is a permutation of the cities).
- Evaluation: Evaluate the fitness of each solution in the population (The fitness of a solution is the length of the tour).
- Selection: Select the best solutions from the population.
- Crossover: Generate new solutions by combining the selected solutions using crossover operators.
- Mutation: Introduce random changes to the new solutions using mutation operators.
- Replacement: Replace the worst solutions in the population with new solutions.

Using the Open Loop Travel Salesman Problem with a genetic algorithm allows the robot to find the shortest path to execute the set of tasks

D. Proposed Method

The initialization stage of the proposed method is constructed using three highly effective techniques that are widely adopted: K-means algorithm [62], [66], Hungarian method [69], and Open-Loop TSP utilizing genetic algorithm [79]. Each of these methods is briefly discussed.

The aim is to distribute a set of tasks with different difficulty levels among a heterogeneous multirobot system operating within a defined space. We intend to allocate these tasks in a collaborative and balanced manner, which poses a search and optimization problem. To illustrate this problem, each robot and task is assigned a specific location within a two-dimensional space, with their Cartesian coordinates represented by real numbers x and y. Furthermore, robots and tasks are characterised by their capability and difficulty, denoted as c and d, respectively.

$$R^{C}_{i\ (x,y)} \quad \text{with } c \in [1, C] \quad \text{and } i \in [1, NR] \tag{10}$$

where C is the maximum capability existing in the MRS and NR is the total number of robots.

$$T^{d}_{j\ (x,y)} \quad \text{with } d \in [1, D] \quad \text{and } j \in [1, NT] \tag{11}$$

where D is the highest difficulty existing in the set of tasks and NT is the total number of tasks.

The tasks and robots are distributed into the workspace, and each robot must be assigned to tasks that match or fall within its capability level. The problem is how to achieve a well-balanced task allocation process, utilizing all the available robots, and minimizing the maximum distance travelled by any robot within the MRS.

The pre-initialization stage involves forming groups of tasks GT, where the count of tasks $\geq$ 2. These groups of tasks are created according to the difficulties of the tasks and the capabilities of the robots. This implies all tasks of difficulty d are temporarily assigned to the robots with a capability equal to d. If no robot possessing a capability c that matches d (where $c \neq d$) is available, then the group of tasks is assigned to the group of robots with the minimum c within the range [d,Cmax], where Cmax is the maximum capability of the robots in the MRS. In case a task requires a capability higher than Cmax, it becomes impossible to perform it. To illustrate how tasks are divided into groups, these examples are used: In problem one as shown in Fig. 1 the environment contains:

- Five robots:
  $R = R^5_1, R^4_2, R^3_3, R^2_4, R^1_5$
- Eleven tasks:
  $T = T^5_1, T^4_2, T^4_3, T^3_4, T^2_5, T^2_6, T^2_7, T^1_8, T^1_9, T^1_{10}, T^1_{11}$
- The groups of tasks generated are:
  $GT_1 = \{T^5_1\};\quad GT_2 = \{T^4_2, T^4_3\};\quad GT_3 = \{T^3_4\};$
  $GT_4 = \{T^2_5, T^2_6, T^2_7\};\quad GT_5 = \{T^1_8, T^1_9, T^1_{10}, T^1_{11}\}$

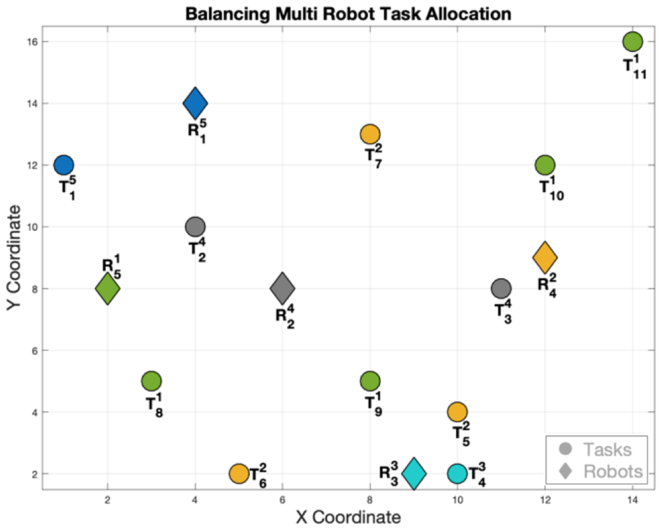

Fig. 1. Position, capability, and difficulty-based distribution of robots and task.

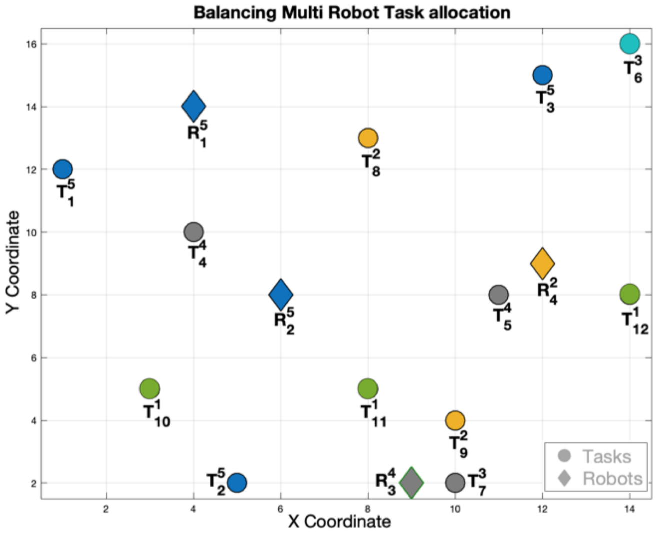

Fig. 2. Position, capability, and difficulty-based distribution of robots and task.

Once the tasks are divided into subgroups, each group is allocated to the robots that possess the required capability to accomplish them therefore:

- GR_1 is assigned to GT_1 where GR_1 = $\{R^5_1\}$ $<=>$ $R^5_1 => T^5_1$
- GR_2 is assigned to GT_2 where GR_2 = $\{R^4_2\}$ $<=>$ $R^4_2 => \{T^4_2, T^4_3\}$
- GR_3 is assigned to GT_3 where GR_3 = $\{R^3_3\}$ $<=>$ $R^3_3 => T^3_4$
- GR_4 is assigned to GT_4 where GR_4 = $\{R^2_4\}$ $<=>$ $R^2_4 => \{T^2_5, T^2_6, T^2_7\}$
- GR_5 is assigned to GT_5 where GR_5 = $\{R^1_5\}$ $<=>$ $R^1_5 => \{T^1_8, T^1_9, T^1_{10}, T^1_{11}\}$

In problem two as shown in Fig. 2 the environment contains:

- Four robots:
  $R = R^5_1, R^5_2, R^4_3, R^2_4$
- Twelve tasks:
  $T = T^5_1, T^5_2, T^5_3, T^4_4, T^4_5, T^3_6, T^3_7, T^2_8, T^2_9, T^1_{10}, T^1_{11}, T^1_{12}$
- The groups of tasks generated are:
  $GT_1 = \{T^5_1, T^5_2, T^5_3\}$; $GT_2 = \{T^4_4, T^4_5, T^3_6, T^3_7\}$;
  $GT_3 = \{T^2_8, T^2_9, T^1_{10}, T^1_{11}, T^1_{12}\}$

In this case:

- GR_1 is assigned to GT_1 where GR_1 = $\{R^5_1, R^5_2\}$
- GR_2 is assigned to GT_2 where GR_2 = $\{R^4_3\}$
- GR_3 is assigned to GT_3 where GR_3 = $\{R^2_4\}$

Once the pre-initialization stage is completed and the group of robots are assigned to their respective group of tasks, the initialization process proceeds in accordance with the criteria outlined in Table I.

TABLE I
ALGORITHM SELECTION IN INITIALIZATION PHASE BASED ON THE NUMBER OF TASKS AND ROBOTS IN EACH GROUP

GR_i	GT_i	K-means	Hungarian	OTSP
$= 1$	$= 1$			
$= 1$	≥ 2			x
≥ 2	$\leq GR_i$		x	x
≥ 2	$\geq GR_i$	x	x	x

K-means algorithm is used to divide each group of tasks GT into k clusters, where k is the number of the robots assigned to this group. Performing this algorithm on all task groups generates new task groups ϑ_i and their corresponding centroids z_i , with $\{i|1 \leq i \leq NR\}$. Then each group of tasks ϑ_i is allocated to the dedicated robot R_i using the Hungarian method, as shown in Table 2. For example, in problem two:

GR_1 is assigned to GT_1 $\equiv$ $\{R^5_1, R^5_2\} \Rightarrow \{T^5_1, T^5_2, T^5_3\}$
K-means$(\{T^5_1, T^5_2, T^5_3\}, k)$ $= (\{\vartheta_1, z_1\}, \{\vartheta_2, z_2\})$
with $\vartheta_1 = \{T^5_1, T^5_2\}$ and $\vartheta_2 = \{T^5_3\}$

TABLE II
TASK ASSIGNMENT USING HUNGARIAN METHOD BASED ON EUCLIDEAN DISTANCE BETWEEN ROBOT POSITIONS AND TASK GROUP CENTROIDS

	z_1	z_2
R^5_1	$\|\|R^5_1, z_1\|\|$ = 7.07	$\|\|R^5_1, z_2\|\|$ = 8.06
R^5_2	$\|\|R^5_1, z_1\|\|$ = 3.16	$\|\|R^5_2, z_2\|\|$ = 9.21

Using the Hungarian method shown in Table II, the best assignment that minimises the total travel cost for each robot is represented in the following manner:

Hungarian$(R_1^5, R_1^5, z_1, z_2) = (\{R_1^5 => z_2\}, \{R_2^5 => z_1\})$ thereby $R_1^5 => T_3^5$ and $R_2^5 => \{T_1^5, T_2^5\}$

After performing Hungarian method, each group tasks ϑ_i are allocated to dedicated robot R_i where $\{i | 1 \leq i \leq NR\}$ which means: $R_i \Rightarrow \vartheta_i = \{T_{i,j}\}$ where $\{j | 1 \leq j \leq NT\}$ is the index of task. To calculate the distance travelled by a robot, the norm is taken between the coordinates of its starting position and the assigned task:

$$Dist_i(R_i, \vartheta_i) = Dist_i(R_i, \{T_{i,j}\}) = \psi(R_i \cup \{T_{i,j}\}) \quad (12)$$

For this, three cases are possible:

- If the robot is not assigned to any task, then $\vartheta_i = \oslash$

$$\psi(R_i \cup \{T_{i,j}\}) = 0 \quad (13)$$

- If the robot is assigned to one task, then $\vartheta_i = T_{i,j}$

$$\psi(R_i \cup \{T_{i,j}\}) = \|R_i, T_{i,j}\| \quad (14)$$

- If the robot is assigned to many tasks

$$\psi(R_i \cup \{T_{i,j}\}) = \|R_i, T_{i,j}\| + \sum_{j \in \vartheta_i} \sum_{l=1}^{L-1} \|T_{i,j}^l, T_{i,j}^{l+1}\| \quad (15)$$

The constant L represents the tasks' aggregate in the group $\{T_{i,j}\}$. The priority ranking of tasks within this group is determined by the genetic algorithm for OLTSP function ξ, which generates the order index l, which indicates the optimal sequence for executing tasks to obtain the shortest path, where:

$$\xi(R_i, \vartheta_i) = \xi(\{R_i, T_{i,j}\}) = \{T_{i,j}^l\} \quad (16)$$

The cost function is the maximum distance travelled by a robot in the MRS :

$$\exists R_i \ Max(\psi \ (R_i \ \cup \{T_{i,j}\})) \quad (17)$$

The objective is to reduce the overall cost by optimising the cost function :

$$\Upsilon(Max(\psi(R_i \ \cup \ \{T_{i,j}\}))) \quad (18)$$

The following iterative optimization algorithm outlines the method for minimizing the overall cost function.

III. Results and discussion

To present the outcomes of our proposed approach, we used the MATLAB software installed in a computer equipped with an i7 / 8GB RAM configuration. This setup enabled us to conduct comprehensive evaluations of our approach and produce reliable results. To demonstrate the versatility and practicality of our algorithms in addressing various challenges in the field of BMRTA, we have selected several distinct examples that involve robots with varying capabilities and tasks with varying levels of complexity.

Algorithm 2 Iterative optimization stage

```
Input: A subgroup of tasks assigned to robots$\{R_i => \{GT_i\}\}$
Output: Optimal assignment of the subgroups of tasks to robot $\{R_i => \{GT_i^*\}\}$
Description:
repeat
    $\alpha$: Find the Robot $R_i^c$ with $Max(\psi(R_i^c \cup \{T_{i,j}\}))$ and label it $R_x^{c0}$
    from $GT_x$ extract the task $T_{x,j}^L$
    for each robot with capability $c0$ do
        Assign temporarily $T_{x,j}^L$ to $R_i^{c0}$ :
        $R_i^{c0} => GT_{i*} = GT_i + T_{x,j}^L$
        Reorder the tasks in $GT_{i*}$ :
        $\xi(GT_{i*}) = \xi(\{T_{i,j}\} \ + \ T_{x,j}^L)$
        if $\psi(R_i^{c0} \cup GT_{i*}) < \psi(R_x^{c0} \cup GT_x)$ then
            $GT_i = GT_{i*}$
            $GT_x = GT_x \setminus T_{x,j}^L$
            goto $\alpha$
        end if
    end for
    for $c$ in $c0 + 1$ to Cmax do
        for each robot with capability $c$ do
            Assign temporarily $T_{x,j}^L \ to \ R_i^c$ :
            $R_i^c \Rightarrow GT_{i*} = GT_i + T_{x,j}^L$
            Reorder the tasks in $GT_{i*}$:
            $\xi(GT_{i*}) = \xi(\{T_{i,j}\} + T_{x,j}^L)$
            if $\psi(R_i^c \cup GT_{i*}) < \psi(R_x^{c0} \cup GT_x)$ then
                $GT_i = GT_{i*}$
                $GT_x = GT_x \setminus T_{x,j}^L$
            end if
        end for
    end for
as long as $(\Upsilon(Max(\psi(R_i^c \ \cup \{T_{i,j}\})) \neq 0)$
```

TABLE III
Initial Robots Configuration: Position and Capability

Robots	R_1	R_2	R_3	R_4
Position	$(4,14)$	$(6,8)$	$(9,2)$	$(12,9)$
Capability	5	5	4	2

We present the initial configuration of the robots and tasks, including their positions, levels of capability, and difficulty, which were randomly generated, (Table III and Table IV).

This configuration is visually represented as a graph (Fig. 3), where robots are depicted as diamonds and tasks are depicted as circles. Following the completion of the initialization phase (Fig. 4), every robot is allocated to a group of tasks in consideration of both the capacity of the robots and the difficulty of the tasks, in the manner outlined below:

TABLE IV
INITIAL TASKS CONFIGURATION: POSITION AND DIFFICULTY

Tasks	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}
Position	$(1,12)$	$(5,2)$	$(12,15)$	$(4,10)$	$(11,8)$	$(14,16)$	$(10,2)$	$(8,13)$	$(10,4)$	$(3,5)$	$(8,5)$	$(14,8)$
Difficulty	5	5	5	4	4	3	3	2	2	1	1	1

- $R_1^5 => GT_1(\vartheta_1)$ where $GT_1(\vartheta_1) = (T_3^5)$ thereby $R_1^5 => T_3^5$
- $R_2^5 => GT_1(\vartheta_2)$ where $GT_1(\vartheta_2) = (T_2^5, T_1^5)$ thereby $R_2^5 => T_2^5, T_1^5$
- $R_3^4 => GT_2$ where $GT_2 = (T_7^3, T_5^4, T_4^4, T_6^3)$ thereby $R_3^4 => T_7^3, T_5^4, T_4^4, T_6^3$
- $R_2^4 => GT_3$ where $GT_3 = (T_{12}^1, T_9^2, T_{11}^1, T_{10}^1, T_8^2)$ thereby $R_2^4 => T_{12}^1, T_9^2, T_{11}^1, T_{10}^1, T_8^2$

The overall time required for the robots to complete their tasks is determined by the robot with the highest cost distance, which in this case is R3, with a cost of 26.02. The result achieved is not balanced or optimal (Table V).

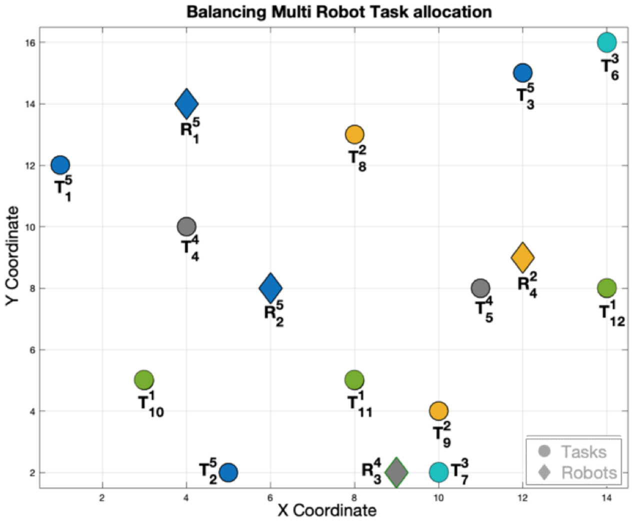

Fig. 3. 2D representation of initial configuration of robots and tasks

In the second phase of the approach, we aim to optimize the system in a balanced manner through collaboration among the robots:

- $R_1^5 => GT_1(\vartheta_1)$ where $GT_1(\vartheta_1) = (T_4^4, T_3^5, T_6^3)$ thereby $R_1^5 => T_4^4, T_3^5, T_6^3$
- $R_2^5 => GT_1(\vartheta_2)$ where $GT_1(\vartheta_2) = (T_2^5, T_1^5)$ thereby $R_2^5 => T_2^5, T_1^5$

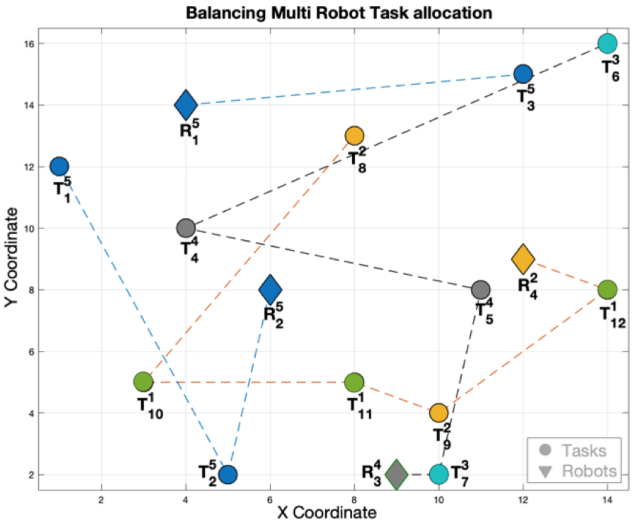

Fig. 4. 2D representation of initialization phase and robot-task assignment

TABLE V
INITIALIZATION PHASE: TRAVEL COST OF ROBOTS WITH ASSIGNED GROUP OF TASKS

R_i	R_1	R_2	R_3	R_4
GT_i	$GT_1(\vartheta_1)$	$GT_1(\vartheta_2)$	GT_2	GT_3
Cost	8, 06	16, 85	**26,02**	24, 56

- $R_3^4 => GT_2$ where $GT_2 = (T_7^3, T_5^4, T_8^2)$ thereby $R_3^4 => T_7^3, T_5^4, T_8^2$
- $R_2^4 => GT_3$ where $GT_3 = (T_{12}^1, T_9^2, T_{11}^1, T_{10}^1)$ thereby $R_2^4 => T_{12}^1, T_9^2, T_{11}^1, T_{10}^1$

The proposed algorithm's application during the optimization phase resulted in a balanced and optimal outcome, leading to a significant reduction in the total cost of distance travelled by a robot from 26.02 to 16.85 (Table VI and Fig. 5). This optimization was achieved within a run time of 0.38 seconds.

TABLE VI
OPTIMIZATION PHASE: TRAVEL COST OF ROBOTS WITH ASSIGNED GROUP OF TASKS

R_i	R_1	R_2	R_3	R_4
GT_i	$GT_1(\vartheta_1)$	$GT_1(\vartheta_2)$	GT_2	GT_3
Cost	15, 67	**16,85**	12, 91	15, 12

Below are additional simulations that showcase balanced optimization across different numbers of robots and tasks.

A. Experiment 1

This experiment involved generating three robots and nine tasks randomly, along with their positions (x, y), capacity, and

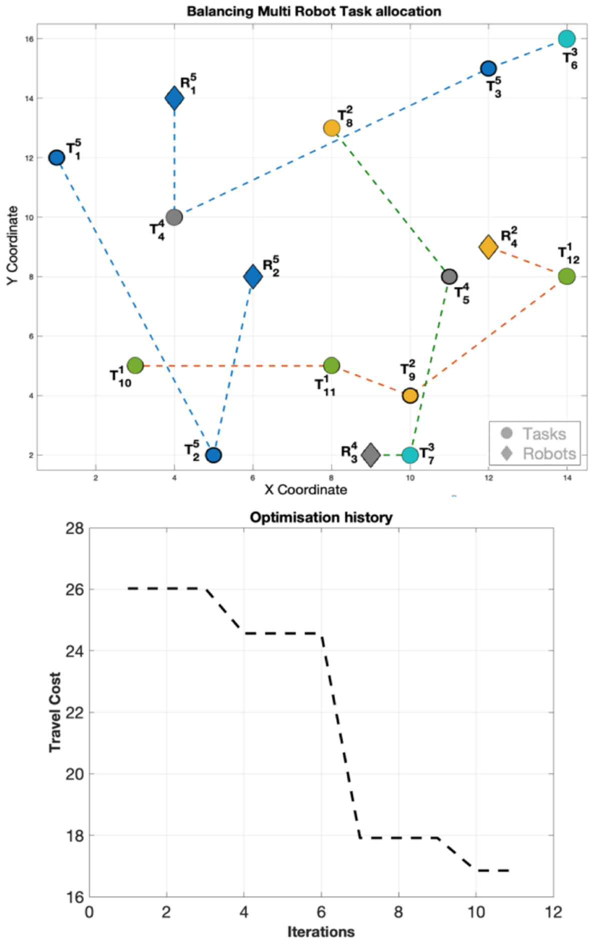

Fig. 5. 2D representation of final result for balanced collaborative multi-robot task allocation

difficulty levels. The initial maximum travel cost was found to be 211.18, which was quite high. However, after implementing the optimization phase, the travel cost was reduced by 90.75 (Fig. 6). This reduction was achieved within a short execution time of 0.34 seconds, demonstrating the effectiveness of the optimization algorithm used in this study.

TABLE VII
EXPERIMENT 1 RESULTS FOR TRAVEL COST MINIMIZATION OVER EXECUTION TIME

Robots	Tasks	Init. cost	Optim. cost	Exec. time(s)
3	9	211.18	**90.75**	0.34

B. Experimental Results

The experiments conducted in this study demonstrated the effectiveness of the optimization algorithm in reducing travel cost, even in scenarios where the number of tasks exceeded the available number of robots. This indicates that the algorithm can handle complex scenarios with ease and still achieve a balanced and optimal solution.

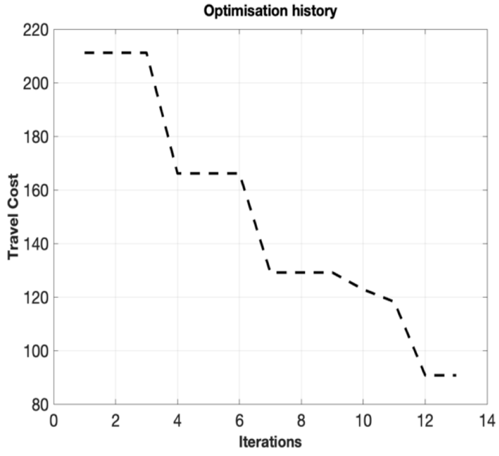

Fig. 6. Experiment 1 optimization history of travel cost by iteration

TABLE VIII
TABLE SUMMARIZING THE EXPERIMENTAL RESULTS FOR MINIMIZING TRAVEL COSTS OVER TIME OF EXECUTION

Experiment	2	3	4	5
Robots	8	10	30	30
Tasks	20	30	100	200
Initialization cost	192.36	211.84	224.88	165.90
Optimization cost	**93.68**	**104.05**	**105.66**	**101.26**
Execution time(s)	20	30	100	200

Interestingly, despite the increase in problem complexity, the execution time required to achieve the optimal solution remained minimal. This suggests that the algorithm is efficient and can be used in real-world applications where time is of the essence.

Overall, these findings highlight the potential of the proposed optimization algorithm in improving the efficiency and effectiveness of robotic systems, particularly in scenarios where resources are limited or where there is a high degree of uncertainty.

IV. CONCLUSION

In this paper, a novel method for assigning heterogeneous tasks to heterogeneous robots is presented. This approach aims to reduce the overall cost of task execution, which is the MRS task with the highest cost incurred by a robot. A variety of tasks with different levels of difficulty are present in the environment, along with a group of robots that are capable of handling tasks that are either equal to or below their capabilities.

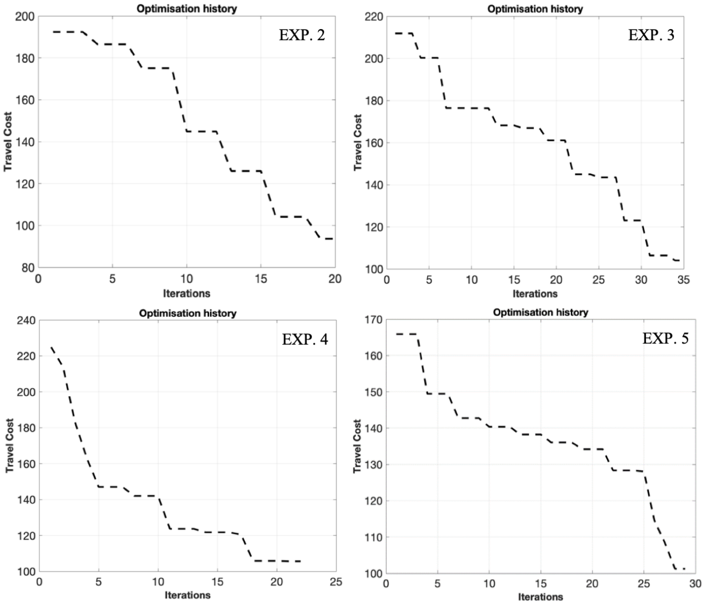

Fig. 7. Experiments: optimization history of travel cost by iteration

The suggested process consists of three steps. The K-means algorithm is used to determine the centroid of each task group and the Hungarian method is used to assign robots in a balanced manner for a set of tasks in the initialization of the allocation based on the difficulties, capabilities, and distances. In order to determine whether the current allocation is optimal or if further optimization is necessary, a validation and re-optimization process is lastly carried out. This centralized approach produces a balanced distribution among all the robots in the MRS and is expandable for any environment, regardless of the number and nature of robots and tasks.

The proposed method is effective and appropriate for MRS task allocation applications, as demonstrated by the simulation experiments and numerical results. However, the effectiveness of this proposed method is constrained because it disregards environments with obstacles. Furthermore, given that each task takes the same amount of time to complete, the task cost only considers the distance between the robot and the task. Therefore, we will consider the environments that are filled with obstacles and the amount of time needed to complete each task in our future work.

References

[1] Y. Chang, Y. Tian, J. P. How, and L. Carlone, “Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 210–11 218.

[2] C. Wei, Z. Ji, and B. Cai, “Particle Swarm Optimization for Cooperative Multi-Robot Task Allocation: A Multi-Objective Approach,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2530–2537, Apr. 2020.

[3] X. Chen, S. He, Y. Zhang, L. C. Tong, P. Shang, and X. Zhou, “Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework,” *Transportation Research Part C: Emerging Technologies*, vol. 114, pp. 241–271, May 2020.

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

[5] S. Choudhury, J. K. Gupta, M. J. Kochenderfer, D. Sadigh, and J. Bohg, “Dynamic multi-robot task allocation under uncertainty and temporal constraints,” *Autonomous Robots*, vol. 46, no. 1, pp. 231–247, Jan. 2022.

[6] X. Chen, P. Zhang, G. Du, and F. Li, “A distributed method for dynamic multi-robot task allocation problems with critical time constraints,” *Robotics and Autonomous Systems*, vol. 118, Aug. 2019.

[7] C. Cai, J. Chen, Q. Yan, and F. Liu, “A multi-robot coverage path planning method for maritime search and rescue using multiple auvs,” *Remote Sensing*, vol. 15, no. 1, p. 93, 2022.

[8] A. Borboni, I. Elamvazuthi, and N. Cusano, “Eeg-based empathic safe cobot,” *Machines*, vol. 10, no. 8, p. 603, 2022.

[9] B. Sahu, P. K. Das, and R. Kumar, “A modified cuckoo search algorithm implemented with sca and pso for multi-robot cooperation and path planning,” *Cognitive Systems Research*, 2023.

[10] O. Hamed and M. Hamlich, “Improvised multi-robot cooperation strategy for hunting a dynamic target,” in *2020 International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, 2020, pp. 1–4.

[11] O. Hamed and M. Hamlich, “Hybrid formation control for multi-robot hunters based on multi-agent deep deterministic policy gradient,” in *MENDEL*, vol. 27, no. 2, 2021, pp. 23–29.

[12] B. A. Baba, S. Garba, and N. Faruk, “Analysis of multi-robot cooperative hunting using differential game technique,” *SLU Journal of Science and Technology*, vol. 6, no. 1&2, pp. 1–7, 2023.

[13] A. Farinelli, E. Zanotto, E. Pagello *et al.*, “Advanced approaches for multi-robot coordination in logistic scenarios,” *Robotics and Autonomous Systems*, vol. 90, pp. 34–44, 2017.

[14] H. Aziz, A. Pal, A. Pourmiri, F. Ramezani, and B. Sims, “Task allocation using a team of robots,” *Current Robotics Reports*, pp. 1–12, 2022.

[15] H. Kang, H. Kim, and Y. M. Kwon, “Recen: resilient manet based centralized multi robot system using mobile agent system,” in *2019 IEEE symposium series on computational intelligence (SSCI)*, 2019, pp. 1952–1958.

[16] A. Elfakharany and Z. H. Ismail, “End-to-end deep reinforcement learning for decentralized task allocation and navigation for a multi-robot system,” *Applied Sciences*, vol. 11, no. 7, p. 2895, 2021.

[17] Y. Msala, M. Hamlich, and A. Mouchtachi, “A new robust heterogeneous multi-robot approach based on cloud for task allocation,” in *2019 5th International Conference on Optimization and Applications (ICOA)*, 2019, pp. 1–4.

[18] F. Quinton, C. Grand, and C. Lesire, “Market approaches to the multi-robot task allocation problem: a survey,” *Journal of Intelligent & Robotic Systems*, vol. 107, no. 2, p. 29, 2023.

[19] W. Yu, Y. Huang, and A. Garcia-Ortiz, “Optimal Task Allocation Algorithms for Energy Constrained Multihop Wireless Networks,” *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7744–7754, Sep. 2019.

[20] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu, “Privacy-Preserving Online Task Allocation in Edge-Computing-Enabled Massive Crowdsensing,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7773–7787, Oct. 2019.

[21] H. Wu, A. Ghadami, A. E. Bayrak, J. M. Smereka, and B. I. Epureanu, “Task allocation with load management in multi-agent teams,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8823–8830.

[22] J. Hu, H. Coffin, J. Whitman, M. Travers, and H. Choset, “Large-Scale Heterogeneous Multi-robot Coverage via Domain Decomposition and Generative Allocation,” in *Algorithmic Foundations of Robotics XV*, ser. Springer Proceedings in Advanced Robotics, S. M. LaValle, J. M. O’Kane, M. Otte, D. Sadigh, and P. Tokekar, Eds., Cham, 2023, pp. 52–67.

[23] R. J. Alitappeh and K. Jeddisaravi, “Multi-robot exploration in task allocation problem,” *Applied Intelligence*, vol. 52, no. 2, pp. 2189–2211, Jan. 2022.

[24] J. G. Martin, F. J. Muros, J. M. Maestre, and E. F. Camacho, "Multi-robot task allocation clustering based on game theory," *Robotics and Autonomous Systems*, vol. 161, p. 104314, 2023.

[25] X. Kong, Y. Gao, T. Wang, J. Liu, and W. Xu, "Multi-robot Task Allocation Strategy based on Particle Swarm Optimization and Greedy Algorithm," May 2019, pp. 1643–1646.

[26] F. Zitouni, R. Maamri, and S. Harous, "FA–QABC–MRTA: a solution for solving the multi-robot task allocation problem," *Intelligent Service Robotics*, vol. 12, no. 4, pp. 407–418, Oct. 2019.

[27] T. Li, H.-S. Shin, and A. Tsourdos, "Threshold Greedy Based Task Allocation for Multiple Robot Operations," Sep. 2019.

[28] G. M. Skaltsis, H.-S. Shin, and A. Tsourdos, "A survey of task allocation techniques in MAS," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, Jun. 2021, pp. 488–497.

[29] Y. Chen, U. Rosolia, and A. D. Ames, "Decentralized Task and Path Planning for Multi-Robot Systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4337–4344, Jul. 2021.

[30] L. Zhou, V. D. Sharma, Q. Li, A. Prorok, A. Ribeiro, P. Tokekar, and V. Kumar, "Graph Neural Networks for Decentralized Multi-Robot Submodular Action Selection," Sep. 2022.

[31] P. Ghassemi and S. Chowdhury, "Multi-robot task allocation in disaster response: Addressing dynamic tasks with deadlines and robots with range and payload constraints," *Robotics and Autonomous Systems*, vol. 147, p. 103905, 2022.

[32] P. Mahato, S. Saha, C. Sarkar, and M. Shaghil, "Consensus-based fast and energy-efficient multi-robot task allocation," *Robotics and Autonomous Systems*, vol. 159, p. 104270, 2023.

[33] S. Guo, S. Cao, and J. Guo, "Study on Decentralization of Spherical Amphibious Multi-robot Control System Based on Smart Contract and Blockchain," *Journal of Bionic Engineering*, vol. 18, no. 6, pp. 1317–1330, Nov. 2021.

[34] G. Notomista, S. Mayya, S. Hutchinson, and M. Egerstedt, "An Optimal Task Allocation Strategy for Heterogeneous Multi-Robot Systems," in *2019 18th European Control Conference (ECC)*, Jun. 2019, pp. 2071–2076.

[35] D. Albani, W. Hönig, D. Nardi, N. Ayanian, and V. Trianni, "Hierarchical Task Assignment and Path Finding with Limited Communication for Robot Swarms," *Applied Sciences*, vol. 11, no. 7, p. 3115, Jan. 2021.

[36] X. Huang, W. Zhang, J. Yang, L. Yang, and C. K. Yeo, "Market-based dynamic resource allocation in Mobile Edge Computing systems with multi-server and multi-user," *Computer Communications*, vol. 165, pp. 43–52, Jan. 2021.

[37] S. N., K. C. R.M., R. M.M., and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Industrial Robot: the international journal of robotics research and application*, vol. 47, no. 6, pp. 929–942, Jan. 2020.

[38] P. Gautier, J. Laurent, and J.-P. Diguet, "Dqn as an alternative to market-based approaches for multi-robot processing task allocation (mrpta)," *International Journal of Robotic Computing*, vol. 3, no. 1, 2021.

[39] Y. Emam, S. Mayya, G. Notomista, A. Bohannon, and M. Egerstedt, "Adaptive Task Allocation for Heterogeneous Multi-Robot Teams with Evolving and Unknown Robot Capabilities," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, May 2020, pp. 7719–7725.

[40] W. Yao, N. Qi, N. Wan, and Y. Liu, "An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles," *Aerospace Science and Technology*, vol. 86, pp. 455–464, Mar. 2019.

[41] D. T. Nguyen, L. B. Le, and V. K. Bhargava, "A Market-Based Framework for Multi-Resource Allocation in Fog Computing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 3, pp. 1151–1164, Jun. 2019.

[42] S. A. Nelke, S. Okamoto, and R. Zivan, "Market Clearing–based Dynamic Multi-agent Task Allocation," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 1, pp. 1–25, Jan. 2020.

[43] N. Geng, Z. Chen, Q. A. Nguyen, and D. Gong, "Particle swarm optimization algorithm for the optimization of rescue task allocation with uncertain time constraints," *Complex & Intelligent Systems*, vol. 7, no. 2, pp. 873–890, Apr. 2021.

[44] H. Wu and H. Shang, "Potential game for dynamic task allocation in multi-agent system," *ISA Transactions*, vol. 102, pp. 208–220, Jul. 2020.

[45] R. H. A. Zubo and G. Mokryani, "Active Distribution Network Operation: A Market-Based Approach," *IEEE Systems Journal*, vol. 14, no. 1, pp. 1405–1416, Mar. 2020.

[46] J. Ma, H. Lu, J. Xiao, Z. Zeng, and Z. Zheng, "Multi-robot Target Encirclement Control with Collision Avoidance via Deep Reinforcement Learning," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 2, pp. 371–386, Aug. 2020.

[47] C. Shyalika, T. Silva, and A. Karunananda, "Reinforcement Learning in Dynamic Task Scheduling: A Review," *SN Computer Science*, vol. 1, no. 6, p. 306, Sep. 2020.

[48] J. G. Martin, J. R. D. Frejo, R. A. García, and E. F. Camacho, "Multi-robot task allocation problem with multiple nonlinear criteria using branch and bound and genetic algorithms," *Intelligent Service Robotics*, vol. 14, no. 5, pp. 707–727, 2021.

[49] O. M. Gul, "Energy harvesting and task-aware multi-robot task allocation in robotic wireless sensor networks," *Sensors*, vol. 23, no. 6, p. 3284, 2023.

[50] J. Li, J. Wu, J. Li, A. K. Bashir, M. J. Piran, and A. Anjum, "Blockchain-Based Trust Edge Knowledge Inference of Multi-Robot Systems for Collaborative Tasks," *IEEE Communications Magazine*, vol. 59, no. 7, pp. 94–100, Jul. 2021.

[51] W. Liang, Z. Ning, S. Xie, Y. Hu, S. Lu, and D. Zhang, "Secure fusion approach for the Internet of Things in smart autonomous multi-robot systems," *Information Sciences*, vol. 579, pp. 468–482, Nov. 2021.

[52] H.-Y. Lee, P. Zhou, B. Zhang, L. Qiu, B. Fan, A. Duan, J. Tang, T. L. Lam, and D. Navarro-Alarcon, "A distributed dynamic framework to allocate collaborative tasks based on capability matching in heterogeneous multi-robot systems," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[53] J. Hu, P. Bhowmick, I. Jang, F. Arvin, and A. Lanzon, "A Decentralized Cluster Formation Containment Framework for Multirobot Systems," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1936–1955, Dec. 2021.

[54] H. Mitiche, D. Boughaci, and M. Gini, "Iterated Local Search for Time-extended Multi-robot Task Allocation with Spatio-temporal and Capacity Constraints," *Journal of Intelligent Systems*, vol. 28, no. 2, pp. 347–360, Apr. 2019.

[55] A. Rauniyar and P. K. Muhuri, "Multi-robot coalition formation and task allocation using immigrant based adaptive genetic algorithms," *Computational Intelligence in Emerging Technologies for Engineering Applications*, pp. 205–225, 2020.

[56] H. H. E. KAYIR, "An effective method to use centralized q-learning in multi-robot task allocation," *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, vol. 27, no. 5, pp. 579–588, 2021.

[57] M. De Ryck, D. Pissoort, T. Holvoet, and E. Demeester, "Decentral task allocation for industrial AGV-systems with routing constraints," *Journal of Manufacturing Systems*, vol. 62, pp. 135–144, Jan. 2022.

[58] W. Mao, Z. Liu, H. Liu, F. Yang, and M. Wang, "Research Progress on Synergistic Technologies of Agricultural Multi-Robots," *Applied Sciences*, vol. 11, no. 4, p. 1448, Jan. 2021.

[59] S. Wang, Y. Liu, Y. Qiu, Q. Zhang, F. Huo, Y. Huangfu, C. Yang, and J. Zhou, "Cooperative task allocation for multi-robot systems based on multi-objective ant colony system," *IEEE Access*, vol. 10, pp. 56 375–56 387, 2022.

[60] R. J. Alitappeh and K. Jeddisaravi, "Multi-robot exploration in task allocation problem," *Applied Intelligence*, vol. 52, no. 2, pp. 2189–2211, 2022.

[61] X. Jing and Z. Cui, "An optimization task scheduling model for multi-robot systems in intelligent warehouses," in *Bio-Inspired Computing: Theories and Applications: 16th International Conference*, 2022, pp. 3–17.

[62] J. MacQueen, "Classification and analysis of multivariate observations," in *5th Berkeley Symp. Math. Statist. Probability*, 1967, pp. 281–297.

[63] M.-S. Yang and I. Hussain, "Unsupervised Multi-View K-Means Clustering Algorithm," *IEEE Access*, vol. 11, pp. 13 574–13 593, 2023.

[64] L. YiFei, H.-L. Minh, S. Khatir, T. Sang-To, T. Cuong-Le, C. MaoSen, and M. Abdel Wahab, "Structure damage identification in dams using sparse polynomial chaos expansion combined with hybrid K-means clustering optimizer and genetic algorithm," *Engineering Structures*, vol. 283, p. 115891, May 2023.

[65] J. Wang, Z. Cao, G. Xu, J. Yang, and A. Kareem, "An adaptive Kriging method based on K-means clustering and sampling in n-ball for structural reliability analysis," *Engineering Computations*, vol. 40, no. 2, pp. 378–410, Jan. 2023.

[66] Z. Huang, "Extensions to the k-means algorithm for clustering large data sets with categorical values," *Data mining and knowledge discovery*, vol. 2, no. 3, pp. 283–304, 1998.

[67] J. Cai, J. Hao, H. Yang, X. Zhao, and Y. Yang, "A review on semi-supervised clustering," *Information Sciences*, vol. 632, pp. 164–200, Jun. 2023.

[68] Q. Huang, R. Gao, and H. Akhavan, "An ensemble hierarchical clustering algorithm based on merits at cluster and partition levels," *Pattern Recognition*, vol. 136, p. 109255, Apr. 2023.

[69] H. W. Kuhn, "Kuhn hw. the hungarian method for the assignment problem. naval research logistic quaterly 1955; 2: 83-97," *Naval Research Logistic Quaterly*, vol. 2, pp. 83–97, 1955.

[70] S. Dhouib, "Novel optimization method for unbalanced assignment problems with multiple jobs: The Dhouib-Matrix-AP2," *Intelligent Systems with Applications*, vol. 17, p. 200179, Feb. 2023.

[71] Z. K. Hashim and M. A. K. Shiker, "A new technique to solve the assignment problems," *AIP Conference Proceedings*, vol. 2414, no. 1, p. 040018, Feb. 2023.

[72] S. Zhang, Y. Xue, H. Zhang, X. Zhou, K. Li, and R. Liu, "Improved Hungarian algorithm–based task scheduling optimization strategy for remote sensing big data processing," *Geo-spatial Information Science*, pp. 1–14, Apr. 2023.

[73] R. Aggarwal and N. Singh, "An Approach to Learn Structural Similarity Between Decision Trees Using Hungarian Algorithm," in *Proceedings of 3rd International Conference on Recent Trends in Machine Learning, IoT, Smart Cities and Applications*, ser. Lecture Notes in Networks and Systems, V. K. Gunjan and J. M. Zurada, Eds., 2023, pp. 185–199.

[74] P. Li and L. Zhou, "Assignment Algorithms for Multi-Robot Multi-Target Tracking with Sufficient and Limited Sensing Capability," *arXiv preprint arXiv:2303.04894*, Mar. 2023.

[75] B. Liu, S. Wang, Q. Li, X. Zhao, Y. Pan, and C. Wang, "Task Assignment of UAV Swarms Based on Deep Reinforcement Learning," *Drones*, vol. 7, no. 5, p. 297, May 2023.

[76] C.-H. Tsai, Y.-D. Lin, C.-H. Yang, C.-K. Wang, L.-C. Chiang, and P.-J. Chiang, "A Biogeography-Based Optimization with a Greedy Randomized Adaptive Search Procedure and the 2-Opt Algorithm for the Traveling Salesman Problem," *Sustainability*, vol. 15, no. 6, p. 5111, Jan. 2023.

[77] Y. Rocha and A. Subramanian, "Hybrid genetic search for the traveling salesman problem with hybrid electric vehicle and time windows," *Computers & Operations Research*, vol. 155, p. 106223, Jul. 2023.

[78] J. Zheng, K. He, J. Zhou, Y. Jin, and C.-M. Li, "Reinforced Lin–Kernighan–Helsgaun algorithms for the traveling salesman problems," *Knowledge-Based Systems*, vol. 260, p. 110144, Jan. 2023.

[79] P. Akshatha, V. Vashisht, and T. Choudhury, "Open loop travelling salesman problem using genetic algorithm," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, no. 1, pp. 112–116, 2013.