A Centralized Task Allocation Algorithm for a Multi-Robot Inspection Mission With Sensing Specifications

Hamza Chakraa, Edouard Leclercq, François Guérin, Dimitri Lefebvre

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 August 2023, accepted 9 September 2023, date of publication 13 September 2023,
date of current version 19 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3315130

RESEARCH ARTICLE

A Centralized Task Allocation Algorithm for a Multi-Robot Inspection Mission With Sensing Specifications

HAMZA CHAKRAA, (Student Member, IEEE), EDOUARD LECLERCQ, FRANÇOIS GUÉRIN, AND DIMITRI LEFEBVRE, (Senior Member, IEEE)
Université Le Havre Normandie, GREAH, 75 Rue Bellot, 76600 Le Havre, France

Corresponding author: Hamza Chakraa (hamza.chakraa@univ-lehavre.fr)

This work was supported by the National Research Agency, France, Region Normandie, and Region Hauts de France, through the Project "Formal Approach and Artificial Intelligence for the Monitoring and Intervention Optimization of Mobile Agents in Industrial Sites" under Grant ANR-21-SIOM-0009-APPRENTIS 2021–2023.

ABSTRACT Recently, considerable attention has focused on enhancing the security and safety of industries with high-risk level activities in order to protect the equipment and environment. In particular, chemical processes and nuclear power generation may have a deep impact on their surroundings. In the case of major events, such as chemical spills, oil rig explosions, or nuclear accidents, collecting accurate and rapidly evolving data becomes a challenging task. So, coordinating a fleet of autonomous mobile robots is a very promising way to deal with unpredicted events and also prevent malicious actions. This paper addresses the problem of assigning optimally a set of tasks to a set of mobile robots equipped with different sensors to minimize a global objective function. The robots perform sensing tasks in order to monitor the area and to facilitate firefighters and inspectors work if a disaster occurs by providing the necessary measures. For this purpose, a centralized Genetic Algorithm (GA) is proposed to determine the task each robot will perform and the order of execution. The proposed approach is tested through a simulation scenario of a grid map environment that represents an industrial area of the city of Le Havre, France. Moreover, a comparative study is conducted with the Hybrid Filtered Beam Search (HFBS) approach and the Mixed-Integer Linear Programming (MILP) solver Cplex. The results demonstrate that the GA approach offers a favorable balance between optimality and execution time.

INDEX TERMS Multi-robot system (MRS), task allocation, genetic algorithm (GA), combinatorial optimization, path planning, industrial area.

I. INTRODUCTION

Nowadays, the increase of critical systems and dangerous sites (where the activity is linked to handling, manufacturing, or storing dangerous substances) [1] imposes high-level security measures to prevent failures and disasters. In case a critical environment is not protected, there could be significant losses in terms of both people and resources. So, employing a team of mobile robots equipped with sensors and load ranges is a promising way to deal with safety issues. Robots are capable of delivering goods, taking measurements, distributing survival kits to possible victims, clearing debris to reach them, and many other tasks. Based on their working environment and the tools they employ, mobile robots can be divided into three categories. Firstly, Unmanned Aerial Vehicles (UAVs) can fly such as drones and helicopters. Then, Unmanned Ground Vehicles (UGVs) and Automated guided Vehicles (AGVs) are robots on the ground that can move on wheels, tracks, or legs. And finally, Autonomous Underwater Vehicles (AUVs) or Unmanned Surface Vehicles (USVs) are robots that travel and operate underwater.

Multi-Robot Systems (MRS) have been used in several applications and industries. Agriculture [2], monitoring [3],

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

search and rescue [4], target detection [5], product pick-up and delivery in distribution centers [6], and health care [7] are some pioneering MRS applications. Further, Multi-Robot Task Allocation (MRTA) is one of the most interesting topics of MRS. It is a problem that aims to coordinate several mobile robots to complete a large set of tasks with specific constraints to optimize an objective function. The MRTA problem has been put out in many different forms. It can be considered as the Linear Assignment Problem (LAP) from Operations Research [8] where each robot in a team can undertake one task at a time and each task has to be completed by a single robot. The LAP can be solved using the Hungarian algorithm with $\mathcal{O}(n^3)$ complexity [9]. Furthermore, other types of MRTA problems are NP-hard and some of them can be modeled as multiple Travelling Salesman Problems [10] or Vehicle Routing Problems [11].

To categorize the MRTA topic, many researchers have proposed different taxonomies. Gerkey and Matarić [12] proposed a taxonomy based on robot capabilities, task requirements, and time to classify the problem. For robots, Single-Task robots (ST) refers to problems where robots can execute only one task at a time. Besides, Multi-Task robots (MT) means that robots are capable of executing multiple tasks at a time. For tasks, Single-Robot tasks (SR) means that the task requires exactly one robot to be done and Multi-Robot tasks (MR) means that the task may require multiple robots. Regarding the assignment time, Instantaneous Assignment (IA) deals with one planning i.e. each robot performs one task, there is no future planning. Besides, a Time-extended Assignment (TA) assigns a sequence of tasks to each robot over a planning horizon. A given MRTA problem can be categorized by combining the three axes. For example, ST-MR-IA is a problem where robots can execute one task at a time, tasks may require the coordination of multiple robots, and the allocation is instantaneous. Korash et al. [13] extend Gerkey and Matarić's taxonomy by including a new element that expresses the tasks as atomic or compound bundles and the dependencies between them. Furthermore, Nunes et al. [14] divided TA into two subsections: temporal and ordering constraints.

MRS algorithms come in different forms and can be classified into two main categories: centralized and decentralized approaches [15]. On the one hand, centralized methods are based on a server that is responsible for communicating all the information to the robots (robots do not communicate with each other). On the other hand, decentralized methods rely on a communication network since robots exchange information directly with each other without the need for a supervisor. Furthermore, Gerkey and Matarić described three types of assignments: Offline, Iterated, and Online assignment. In Offline allocation, the tasks are provided at once and there is no dynamic assignment. Besides, Iterated and Online assignments deal with the reassignment in dynamic environments [16] where several constraints may appear during task execution (failure of robots, dynamical tasks, moving obstacles, environment changes ...). The iterated assignment is an iterated version of the offline assignment where a new solution is computed once a new task is added to the system. For Online assignments, robots obtain new tasks after completing their previously allocated ones, they do not cancel their prior allocation.

In this study, we consider a multiple mobile robots mission where each robot is equipped with specific sensors to perform measurements that are spatially distributed in different positions of an industrial environment. Developing Multi-Robot missions that integrate the capabilities of robots adds new restrictions and additional constraints that weren't addressed in the past formulations. In our context, the tasks to be performed are measurements that are collected by the sensors carried by the robots. A sequence of tasks is assigned to each robot such that the team aims to take all tasks at the minimum cost. Following the task allocation, a path planning phase will be performed, enabling the robots to efficiently navigate and accomplish their designated missions. Note that a robot may perform several measurements in the same position simultaneously if equipped with the appropriate sensors. Therefore, the problem is classified according to Gerkey and Matarić's taxonomy as belonging to the MT-SR-TA configuration. The resolution of problem configurations involving MT robots. has been notably limited in the existing research efforts. In particular, an additional effort in the field of MT-SR-TA configuration is needed [17], [18]. Our MRTA problem presents a multidimensional complexity due to the combined challenges of handling MT robots, incorporating spatially distributed measurements, and optimizing task allocation. The problem is combinatorial by nature and results in exponential growth of the set of possible solution candidates as the number of robots and tasks increases.

The contributions of the paper are to provide a comprehensive model of the problem and to propose an offline centralized Genetic Algorithm (GA) approach to solve tasks assignment and planning simultaneously. The main innovations are as follows: to develop a decision-making tool for rapidly organizing information collection during industrial incidents and providing potential solutions based on GA. This tool has the potential to significantly improve the efficiency and effectiveness of monitoring and incident response in the case of industrial events. Moreover, the proposed setting includes specific constraints that were not considered in the existing studies. Finally, the encoding of the problem specifications on the genome is fully detailed, such that the method becomes repeatable for comparison purposes. To the best of the authors knowledge, this paper is the first contribution that proposes to solve the considered MRTA problem with GA. In contrast to the most related works [18], [19], our contribution tackles a more intricate facet of the MRTA problem. The authors in [18] primarily focus on optimizing parallel task allocation without accounting for robots capabilities and tasks requirements within a MT-SR configuration. The authors in [19] introduce a

Genetic Mission Planner (GMP) for heterogeneous multi-robot systems, considering robots capabilities and tasks requirements excluding explicit treatment of parallel tasks execution and working within a ST-SR context with multiple depots. In comparison, our work addresses a notably distinct MRTA scenario. We propose an innovative model featuring a single depot that encompasses the integration of diverse robots capabilities depending on tasks requirements and the execution of parallel tasks within a MT-SR configuration. This comprehensive approach reflects the real-world complexities of multi-robot missions and presents a novel solution paradigm that surpasses the limitations of the existing literature.

Further, the cost is interpreted as the robot's travel time from one location to another one. Moreover, the battery range of each robot is considered in the cost function as the maximum cumulative travel time. So, real-world constraints are taken into account during the optimization process. Finally, we validate the proposed method by including a comparative study that compares the given approach with MILP and another existing approach. This comparison provides insights into the strengths and weaknesses of the method and demonstrates its effectiveness relative to other approaches. In summary, The main contributions of our paper are as follows:

1) Novel problem consideration: this paper addresses the MRTA problem in the context of a fleet of sensing robots that aim to perform multiple tasks based on their capabilities. This is an important and relatively unexplored area of research within MRTA [17], [19], [20].
2) MILP formulation: this paper proposes a Mixed-Integer Linear Programming (MILP) formulation of the MRTA problem in the given context. This formulation offers a mathematical model that incorporates the key aspects of the problem and can be used to optimize the assignment and planning of the sensing robots using a dedicated solver such as Cplex.
3) Evolutionary centralized algorithm: an evolutionary centralized algorithm that solves the addressed MRTA problem is introduced. This approach provides a solution for the simultaneous assignment and routing of the robots taking into account the unique capabilities of each robot and the specific requirements of each task.

The rest of this paper is organized as follows: in the second section, we summarize the MRTA state of the art. Section III is dedicated to describe the problem studied and presents a mathematical model that characterizes it. Further, the techniques used in this paper are introduced and detailed in Section IV, and the results are discussed through a simulation scenario in Section V. Finally, Section VI concludes with some recommendations and mentions for future studies.

II. STATE OF THE ART

In this study, we address the resolution of a specific MRTA problem using GA. The reader can refer to [12], [13], [14], [15], [16], [17], and [21] for detailed reviews of GA and other methods applied in the MRTA context.

A. MULTI-ROBOT TASK ALLOCATION (MRTA)

MRTA problem can be solved using one of the three main strategies: market-based, optimization-based, or behaviour-based approaches. Market-based techniques, which draw their inspiration from market trading, offer practical answers to the MRTA problem. There have been various resource allocation and optimization problems resulting from the similarity between distributed computer systems and economic systems [22]. They are based on a bidding procedure where robots compete for tasks to satisfy a given set of specific requirements. Market-based methods can be either centralized where a central auctioneer (server or robot) is responsible for collecting bids and assigning tasks to robots, or distributed where robots solve conflicts on their own.

The auction algorithm is a classic market-based technique that has been adopted in MRTA [23]. In [24], a distributed auction technique has been used to compute the optimal solution to a MRTA problem with task deadline limitations. Further, while respecting the maximum number of tasks each robot is capable of performing, the cost function is to maximize the overall payoff of the assignment. Liu et al. developed a distributed market-based algorithm to compute the optimal solution for ST-SR-IA problems. This algorithm has a polynomial time complexity and outperforms the conventional auction method. Afterward, the popular Consensus-Based Bundle Algorithm (CBBA), which links auction characteristics and a consensus process to create a conflict-free allocation was presented in [25]. Researchers in [26] were inspired by this work to suggest alternative algorithms that could outperform CBBA in terms of optimality and communication. In general, market-based method extensions were made by several studies in order to address complex coordination constraints [27].

For optimization-based approaches, there are two main categories: exact approaches and approximate approaches [28]. The ST-SR-IA is the only configuration where a solution can be computed in polynomial time. Thus, it can be solved using an exact approach such as the Hungarian Algorithm (HA) [29] or Integer Linear Programming (ILP) [30]. Besides, approximated methods such as heuristics and meta-heuristics have been widely adopted in the MRTA literature because of their reasonable numerical complexity for large-scale problems. Bio-inspired algorithms such as Ant Colony Optimization were used to solve MRTA problems [31]. It was also solved using other heuristics and meta-heuristics such as Beam Search [32], Tabu Search [33], Particle Swarm Optimization [34], and Simulated Annealing [35]. Even though meta-heuristics solve problems in a reasonable time, it is still limited in larger-size problems because of the highest difficulty of the problem. Consequently, researchers have proposed to divide the process into two stages: the first part is about tasks assignment, and the second part is about

tasks planning and scheduling. In [36], a large-scale problem is solved by partitioning the state space using a k-means clustering technique. Following the grouping of the tasks, the HA efficiently distributes robots throughout the clusters and then GA is applied in each cluster to deal with the planning.

The last possible strategies to solve MRTA problems are behavior-based approaches [37] where tasks are divided into behaviors. It focuses on robots behaviour that will be empowered to make decisions while the mission is in progress instead of relying on a supervisor. In [38], a behavior-based algorithm called ALLIANCE was developed to perform tasks as soon as possible. Further, the BLE (Broadcast of Local Eligibility) algorithm [39] was developed to deal with dynamic task allocation (new tasks appearing during the mission). Additionally, some specific categories of MRTA with tasks that include dependencies and temporal window constraints can be solved using Markov decision processes [40].

B. TASK ALLOCATION WITH GA

GA is a metaheuristic technique that is inspired by Darwin's process of natural selection [41]. It belongs to the class of Evolutionary Algorithms that solve optimization problems from a population of solutions (called individuals). It has been used to solve a variety of NP-hard problems for specific applications and areas such as Facility Layout Problems [42], Job Shop Scheduling (JSP) [43], Inventory Control [44], Image Processing [45], Medical Imaging [46], and Wireless Networking [47]. In the MRTA context, GA is widely used. Authors of [48] presented a problem where 3 robots must inspect fixed locations in an industrial area. Allocation of tasks was performed using GA and then trajectories were computed by the A* algorithm in a centralized manner. Besides, a decentralized task assignment based on GA and communication across several UAVs has been reported in [49]. It aims to calculate a set of tasks for each UAV that minimizes their cumulative flight time. Further, in situations where a task requires the combined effort of more than one robot, GA was also employed in [50] and [51]. A group of homogeneous robots must carry out several tasks according to their priorities. So, the cost function is calculated as the total sum of the robots total travel time, the time spent on each task, the time spent waiting for other robots to perform a task that requires cooperation, and the waiting time due to precedence constraints. The study of [52] has considered a MRTA problem with humanoid robots for search and rescue in dangerous areas. In the first phase, tasks were clustered using k-medoid clustering, and then GA assigned robots to clusters and solved the routing in each group. Likewise, Alitappeh et al. [53] used the Voronoi diagram for partitioning the state space, and then they applied GA and Q-learning techniques in order to solve tasks planning.

The use of meta-heuristic methods, including GA, has been found to be effective in providing good solutions within a reasonable time frame, even for large-scale MRTA problems. GA is indeed a commonly used approach in this field due to its efficiency in exploring large solution spaces and its adaptability to different problem formulations, objective functions, and constraints. Furthermore, it is worth noting that the majority of studies in the literature insufficiently addressed additional constraints that can be crucial in real-world MRTA scenarios. For instance, they do not take into account different robots capabilities and tasks requirements, where multiple tasks might be required at the same position. In this paper, the utilization of GA is extensively employed for addressing this formulation. The robot capabilities and tasks requirements are both encoded into the GA representation and fitness functions, which take into account both the task allocation and the compatibility between robots and tasks. By incorporating these constraints into the evolutionary process, the GA generates solutions that not only adhere to the problem's requirements but also optimize the given objectives.

III. PROBLEM STATEMENT

A. DESCRIPTION AND ASSUMPTIONS

The environment of navigation is a spatial grid map that is composed of cells and considers the coordination between UGVs and UAVs. For UAVs, a three-dimensional mesh of size ($N_x \times N_y \times N_z$) is taken into consideration, while for UGVs, a two-dimensional mesh of size ($N_x \times N_y$) is considered. The state space is composed of V cells characterized by their addresses $\mathcal{V} = \{v_1, \ldots, v_j, \ldots, v_V\}$ and the spatial coordinates of their center of gravity (x_j, y_j, z_j). Note that the size of each one is defined according to problem specifications. Moreover, one or more robots may simultaneously visit a cell and there could be various kinds of obstacles or one-way paths in the environment.

Several measurements must be performed in some particular cells of the environment, called "sites" in the next of the paper. The set of sites to be visited is defined as a subset of $\mathcal{V}$ and we refer to this subset as $\mathcal{A} = \{a_1, \ldots, a_i, \ldots, a_A\}$. Each site a_i corresponds to a cell v_j of the environment. The robots start and return to the same site a_1 called "depot" where no task is required. $\mathcal{M} = \{m_1, \ldots, m_q, \ldots, m_M\}$ is the set of measurement types. These measurements likely represent the different data or information that the robots need to collect during the mission. The set of tasks, that must be performed by the robots in order to accomplish the mission, is defined by $\mathcal{J} = \{j_1, \ldots, j_t, \ldots, j_T\}$ where T represents the total number of tasks. We formally define a task $j_t = (i, q)$ as a measurement of type $m_q \in \mathcal{M}$ to be performed in a given site $a_i \in \mathcal{A}$. Further, we define the function $\mathcal{T}$: $\mathcal{M} \times \mathcal{A} \rightarrow \{0, 1\}$ such that $\mathcal{T} = [t_{i,q}]$, with $t_{i,q} = 1$ if a measurement of type m_q is required in the site a_i, otherwise $t_{i,q} = 0$.

The mission is carried out by a team of R heterogeneous mobile robots $\mathcal{R} = \{r_1, \ldots, r_k, \ldots, r_R\}$ equipped with specific sensors that execute the tasks. Depending on their sensors, robots are of different types and two robots may belong to the same type. Each sensor is suitable for a certain

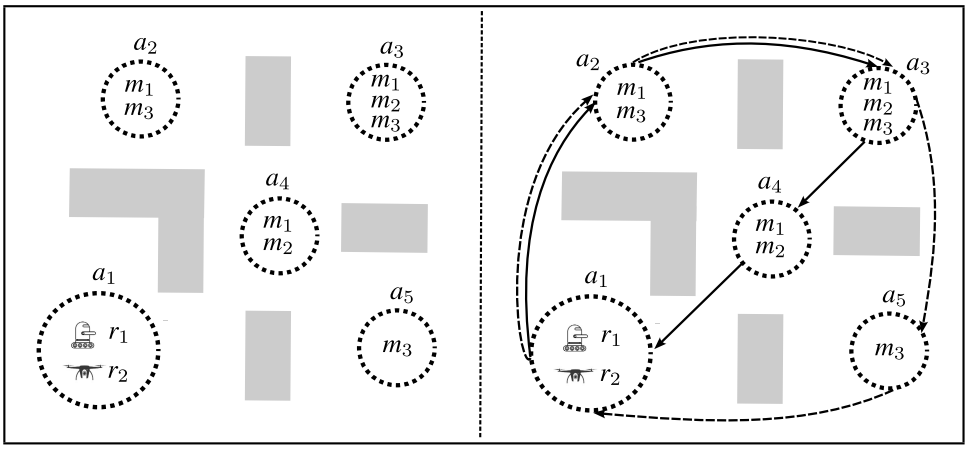


FIGURE 1. Mission example: An illustrative scenario (left), A possible solution (right).

measurement, and $\mathcal{M}$ represents also the set of sensors. The function $\mathcal{P}$: $\mathcal{M} \times \mathcal{R} \rightarrow \{0, 1\}$ is defined such that $\mathcal{P} = [p_{k,q}]$, with $p_{k,q} = 1$ if robot r_k is equipped with a sensor of type m_q, otherwise $p_{k,q} = 0$. Moreover, each robot is subject to limitations in terms of time of service or energy consumption. $\mathcal{B}_k$ represents a limited maximal travel cost for robot r_k where the cost may refer to time or energy.

Once the environment, tasks, and robots are defined, the minimum travel costs between sites must be calculated. First, for each robot, each displacement between two adjacent (neighboring) cells v_j and $v_{j'}$ is defined by an elementary cost $\tilde{c}(j, j', k) > 0$ that is assumed to be known. Note that moving directly from v_j to $v_{j'}$ is not possible if the cells aren't adjacent (in this case $\tilde{c}(j, j', k) = \infty$). Afterward, the environment of each robot r_k is represented as a weighted graph that depends at first on the 2 or 3-dimensional mesh used for r_k. Each cell is represented by a node and the transitions between the cells correspond to the potential displacements associated with the elementary costs.

At this level, the Dijkstra algorithm [54] is used to calculate the smallest cost $c(i, i', k)$ between each pair of sites $a_i, a_{i'} \in \mathcal{A}$ for each robot r_k (due to the heterogeneity of the fleet, each robot has its own graph and costs table). Consequently, for any sites $a_i, a_{i'}, a_{i''} \in \mathcal{A}$, $a_i, a_{i'}, a_{i''}$ being different or identical, we have $c(i, i', k) \leq c(i, i'', k) + c(i'', i', k)$. Observe that the same symbols, $\tilde{c}(j, j', k)$ and $c(i, i', k)$, can be used to denote the amount of energy or time depending on the situation. In the rest of this paper, we refer to energy units as EU and time units as TU.

The problem considered in this paper is to assign suitable robots to the tasks and to plan the sequence of tasks $S(k)$ for each robot r_k in order to minimize a cost function. The following assumptions are considered:

- A robot can execute one or several tasks at the same location at a time (MT);
- Each task is completed by a single robot (SR);
- Each robot's allocation is considered over a planning horizon (TA);
- All the robots start and return to the depot a_1;
- Each task is carried out by a robot equipped with the required sensor;

According to Gerkey and Matarić's taxonomy, our problem falls in the MT-SR-TA configuration. The MinSum and MinMax global cost functions are commonly used in combinatorial optimization [55]. The MinSum objective function $C_1 : \{\mathcal{S}(k), r_k \in \mathcal{R}\} \rightarrow \mathbb{R}$ is used to evaluate the overall cost incurred by all robots. In the next, we will use this function with energy units. In such a case, it is interpreted as the amount of energy used by the entire fleet of robots to accomplish all tasks:

$$C_1 = \sum_{r_k \in \mathcal{R}} C(\mathcal{S}(k)) \tag{1}$$

where $C(\mathcal{S}(k))$ stands for the cost of the sequence of tasks $\mathcal{S}(k)$ to be performed by the robot r_k. Further, the MinMax objective function $C_2 : \{\mathcal{S}(k), r_k \in \mathcal{R}\} \rightarrow \mathbb{R}$ is used to evaluate the highest individual cost of the robots. In a search and rescue mission where the objective is to reach a victim as soon as possible, it might be preferable to use the MinMax cost function with time units:

$$C_2 = \max_{r_k \in \mathcal{R}} C(\mathcal{S}(k)) \tag{2}$$

The approach proposed in this paper is suitable for the two objective functions mentioned previously. As a problem context, we consider a multi-robot mission of gathering measurements in an industrial area for surveillance in the first phase and for intervention in case a disaster occurs. For surveillance missions, we use the MinSum function to minimize the cumulative energy consumption by the entire fleet. Further, if an industrial accident turns out, the MinMax function will be employed in order to gather the information in a short time.

Example: Figure 1 left presents an example of a possible scenario. Here are the key details:

- Sites: the tasks are distributed over four specific sites, denoted as $\{a_2, a_3, a_4, a_5\}$ and a_1 is the base location

TABLE 1. Variables used in this paper.

Variable	Description
$\mathcal{V}$	The set of positions
$\mathcal{A}$	The set of sites
$\mathcal{R}$	The set of robots
$\mathcal{M}$	The set of measurements types
$\mathcal{J}$	The set of tasks
$\mathcal{P}$	The function that defines the robots capabilities
$\mathcal{T}$	The function that defines the tasks requirements
$c(i,i',k)$	The cost between each two sites a_i and $a_{i'}$
$\tilde{c}(j,j',k)$	The elementary cost between each two adjacent positions v_j and $v_{j'}$
$\mathcal{S}$	The computed sequence of tasks for each robot
τ_p	Population size
τ_g	Number of generations
τ_c	Crossover probability
τ_m	Mutation probability

TABLE 2. Abbreviations used in this paper.

Abbreviation	Description
MRS	Multi-Robot-System
MRTA	Multi-Robot Task Allocation
MT	Multi-Task robots
SR	Single-Robot tasks
GA	Genetic Algorithm
HFBS	Hybrid Filtered Beam Search
MILP	Mixed-Integer Linear Program
ILP	Integer Linear Programming
TU	Time Unit
EU	Energy Unit

TABLE 3. Tasks required in each site.

Site \ Measurement	m_1	m_2	m_3
a_1	✗	✗	✗
a_2	✓	✗	✓
a_3	✓	✓	✓
a_4	✓	✓	✗
a_5	✗	✗	✓

for the robots from where they start and return after completing their assigned tasks.

- Measurements: there are three types of measurements considered in this scenario, namely $\mathcal{M} = \{m_1, m_2, m_3\}$.
- Tasks: there are a total of eight tasks that need to be assigned to the robots. Table 3 defines tasks in each site, i.e., defines the function $\mathcal{T}$.
- Robots: there is a team of two robots, denoted as $\mathcal{R} = \{r_1, r_2\}$ equipped with different sensors. The robot r_1 is equipped with two sensors that can perform m_1 and m_2. On the other hand, robot r_2 is equipped with two sensors capable of performing m_2 and m_3. The capabilities of each robot and its corresponding sensors are summarized in Table 4, which defines the function $\mathcal{P}$.

TABLE 4. Capabilities of each robot.

Robot \ Sensor	m_1	m_2	m_3
r_1	✓	✓	✗
r_2	✗	✓	✓

In this scenario, a MinSum optimization is considered to determine the sequences of the two robots, r_1, and r_2. For simplicity, the reserve of autonomy of the robots is assumed to be large enough, meaning that they have the necessary energy resources to perform their tasks efficiently. In Figure 1 right, the trajectory of robot r_1 is represented by the solid black lines, while the trajectory of robot r_2 is represented by the dotted black lines. The robot r_1 visits successively a_2, a_3 and a_4 to perform respectively m_1 in a_2, m_1 again in a_3, and $\{m_1, m_2\}$ in a_4. Besides, r_2 visits a_2, a_3, and a_5 to perform m_3 in a_2 $\{m_2, m_3\}$ in a_3, and m_3 in a_5.

B. MILP FORMULATION

The problem may be formalized with Mixed Integer Linear Programming (MILP) as detailed in Equations (3) to (9). The boolean decision variable $x_{i,j}^k$ is 1 if robot r_k moves from state a_i to a_j to take one or more measurements, and it is 0 otherwise. In addition, the integer decision variable u_i^k gives the order of visit of the site a_i by robot r_k in the case where a_i belongs to the mission of r_k, otherwise the value of u_i^k do not matter.

Equations (3) and (4) indicate the objective functions, that minimize the sum of the cost of each robot in the case of the MinSum optimization (3) and minimize the max of the cost of the robots in the case of MinMax optimization (4). Equations (5) to (7) impose that each robot circulates along a circuit including the site a_1 (depot) [56]. Constraint (8) guarantees that each task is accomplished by at least one robot. However, it inherently tends to converge towards the optimal solution of having one robot per task due to the associated minimal cost. Constraint (9) deals with the autonomy limitations of the robots, i.e. the overall cost of the mission of robot r_k must not exceed $\mathcal{B}_k$.

$$\min \sum_{r_k \in \mathcal{R}} \sum_{a_i, a_j \in \mathcal{A}} x_{i,j}^k \times c(i,j,k) \tag{3}$$

$$\text{or min} \max_{r_k \in \mathcal{R}} \sum_{a_i, a_j \in \mathcal{A}} x_{i,j}^k \times c(i,j,k) \tag{4}$$

$$\text{s.t.} \sum_{a_i \in \mathcal{A}} x_{1,i}^k = 1 \qquad \forall r_k \in \mathcal{R} \tag{5}$$

$$\sum_{a_j \in \mathcal{A}} x_{i,j}^k = \sum_{a_j \in \mathcal{A}} x_{j,i}^k \quad \forall a_i \in \mathcal{A}, \ \forall r_k \in \mathcal{R} \tag{6}$$

$$u_i^k + x_{i,j}^k \leq u_j^k + (A-1) \times (1 - x_{i,j}^k)$$
$$\forall a_i \in \mathcal{A}, \ \forall a_j \in \mathcal{A}/\{a_1\}, \ \forall r_k \in \mathcal{R} \tag{7}$$

$$\sum_{r_k \in \mathcal{R}} \sum_{a_i \in \mathcal{A}} p_{k,q} \times x_{i,j}^k \geq t_{j,q} \quad \forall m_q \in \mathcal{M}, \ \forall a_j \in \mathcal{A} \tag{8}$$

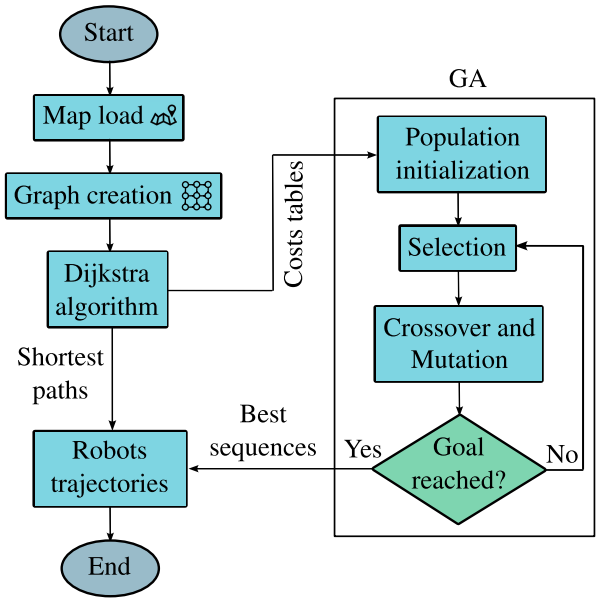


FIGURE 2. Architecture of the proposed system.

$$\sum_{a_i,a_j \in \mathcal{A}} x_{i,j}^{k} \times c(i,j,k) \leq \mathcal{B}_k \quad \forall r_k \in \mathcal{R} \tag{9}$$

IV. PROPOSED GA-BASED SOLVER

Figure 2 illustrates the architecture of the proposed system. First of all, we represent the environment as a grid map that will be converted into a directed graph. Further, the Dijkstra algorithm will compute the shortest paths between each pair of sites (a_i, a_j) and the corresponding traveling costs. At this stage, the generated costs table will be used as inputs for the GA that will perform task allocation (task allocation means the calculation of the complete sequence for each robot and thus determining assignment and planning). Finally, the trajectory of each robot will be computed according to the computed sequences by the allocation solver and the shortest paths generated previously by the Dijkstra algorithm.

In GA, an initial population of individuals serves as a starting generation. Further, to create the following generation, different operations are applied to this population. The fitness of each member of the population is calculated in every generation (the cost value of each individual) and from the existing population, the fittest individuals are stochastically chosen, and certain individual's genome is changed to create a new generation. When the solution has reached a suitable fitness level or the algorithm has produced a maximum number of generations, the algorithm ends. The primary goal of the GA is to raise the population of solutions fitness values using crossover and mutation operators before identifying the best solution. The rest of this section presents the details of our genetic process. Let us introduce some basic terms that will be used throughout this section:

- Population: a subset of all potential (encoded) solutions to the problem in the current generation.
- Chromosome: a potential solution to the given problem.
- Gene: element of a chromosome.

Position	2	2	3	3	3	4	4	5
Measure	1	3	1	2	3	1	2	3
Assigned robot	1	2	1	2	2	1	1	2

FIGURE 3. Chromosome encoding.

- Fitness function: a function that takes a solution as an input and outputs if it is suitable or not. In this study, the fitness function is the objective function.

A. CHROMOSOME ENCODING

In the context of the problem being studied, the chromosome or individual encoding is a fundamental component of a GA. It serves as a representation of a solution and is inspired by the structure of real DNA chromosomes. It is typically represented as a string or a bit-string format. Each position within the string corresponds to a gene, which represents a specific attribute or characteristic of the solution. In this particular work, the chromosome is represented using a table consisting of three lines. The structure of the chromosome table is as follows:

- The first line of the table represents the positions. These positions refer to the specific sites where the tasks need to be performed by the robots.
- The second line of the table represents the associated measurements. Each position in this line corresponds to a measurement type that needs to be performed at that position.
- The third line of the table designates the assigned robots. Similar to the previous lines, each position in this line corresponds to the robot that is assigned to perform the measurement at the respective site.

Therefore, each column in the chromosome table, which is composed of a position, an associated measurement, and the assigned robot represents a gene in the chromosome. The information encoded in the chromosome describes the assignment of tasks to robots, considering the positions and the measurements. Afterward, to provide a visual representation of the solution encoding, Figure 3 illustrates the chromosome table for the example presented in Figure 1. This visual representation helps to understand how the information is organized and encoded within the chromosome.

Since the proposed method aims to calculate a sequence of tasks for each robot, the chromosome will be converted to a sequence of tasks for each robot considering departure and return to the depot. In detail, the sequences of the chromosome in Figure 3 are $\mathcal{S}(1) = (a_1, \varnothing)(a_2, m_1)(a_3, m_1)(a_4, m_1)(a_4, m_2)(a_1, \varnothing)$ and $\mathcal{S}(2) = (a_1, \varnothing)(a_2, m_3)(a_3, m_2)(a_3, m_3)(a_5, m_3)(a_1, \varnothing)$.

B. POPULATION INITIALIZATION

The GA process begins with population initialization. In this step, the first generation of individuals is created. The total

Position	2	2	3	3	3	4	4	5
Measure	1	3	1	2	3	1	2	3
Suitable robots	1	2	1	[1 2]	2	1	[1 2]	2

FIGURE 4. Robots that can perform each task.

number of generations in the algorithm is denoted as τ_g. The population size for each generation is denoted as τ_p.

There are two common approaches for population initialization: random initialization and heuristic initialization. In random initialization, the initial population is generated by randomly creating individual solutions or chromosomes. On the other hand, Heuristic initialization involves filling the initial population using a heuristic method. It can provide a few good solutions to seed the population, which may help to speed up the convergence toward optimal solutions. However, it has been noted that employing a heuristic to initialize the entire population can lead to a population with highly identical solutions and minimal variety. The random solutions are the ones that push the population toward optimality according to experimental observations. Therefore, instead of providing the entire population with heuristic-based solutions, heuristic initialization can be used to seed the population with a few good solutions and fill in the gaps with random ones. Additionally, it has been noted that, in some instances, heuristic initialization only affects the population's initial fitness; yet, in the end, it is the variety of solutions that produce optimality.

In this study, the initial population is created randomly. Nevertheless, in order to avoid unsuitable solutions, only appropriate robots are considered for a given task. Figure 4 presents the table from which the first population is generated. A random draw is done in the third line which allows for choosing one of the suitable robots for each task that involves the measurement m_2.

C. SELECTION AND FITNESS EVALUATION

As previously mentioned, the fitness function is a crucial component of GA that evaluates how ''fit'' or ''acceptable'' a candidate's solution is in relation to the problem under discussion. The calculation of fitness value must be quick enough because it is performed repeatedly in a GA. The purpose of this optimization is to minimize the objective function. After each iteration, a selection process takes place according to the cost function and the chosen chromosomes will construct the new population.

GA evaluates the quality of solutions represented by the chromosomes in a population using fitness (cost) as a designator. The purpose of a GA's selection component is to employ fitness to lead the evolution of individuals. There should be a selective push towards more highly fit solutions since those with greater fitness should have a better possibility of selection than those with lower fitness. However, it is also essential to consider lower fitness solutions during the evolution process, as they can potentially contribute to the generation of better solutions through the recombination process. In this particular work, the new population should be composed of several elements. Firstly, a portion of the best parents is included. Additionally, a portion of the best offspring, generated through recombination, is also incorporated into the new population. This allows for the preservation and propagation of favorable genetic material. Furthermore, the population should also contain some mutants, which are individuals with randomly altered genetic information. Mutations introduce diversity into the population and help explore new areas of the solution space. Lastly, a random complement of lower-fitness parents and offspring is included in the new population. While these individuals may have lower fitness, they still play a role in the evolutionary process by potentially contributing to the generation of improved solutions through recombination or mutation.

By combining these different elements in the new population, the GA can strike a balance between exploiting highly fit solutions and exploring the solution space for potentially better alternatives.

D. CROSSOVER AND MUTATION

Chromosomes chosen from a population are recombined to create the next solution through the process of recombination. Crossover and mutation are the two basic genetic operators that make up recombination. The behavior of genetic operators is nondeterministic. Each has a distinct chance of happening, and the precise result of the crossover or mutation is equally unpredictable. In order to create one or two successor chromosomes, the crossover operator simulates the merging of genetic material from two chosen parent chromosomes. In a GA, the crossover is typically used with a high rate of happening τ_c. In this work, the crossover operator (Figure 5) is mainly done in the four following steps:

1) Randomly select two genes from each chromosome.
2) Identify corresponding genes in each chromosome that have the same measure. There may be multiple corresponding genes, so one is randomly chosen from each individual.
3) Swap the assignments of the robots between the selected gene and its counterpart in the other chromosome. This modification alters the assignment of tasks to robots.
4) Swap the positions of the two selected genes within the same individual chromosome. This modification changes the routing of tasks within the chromosome.

On the other hand, the mutation operator introduces minor, arbitrary changes to a chromosome. It operates by modifying information in one or more of its genes to obtain a new solution. It is usually used with a low rate of happening τ_m to maintain diversity in the genetic population (GA is reduced to a random search if the mutation rate is high). In this paper,

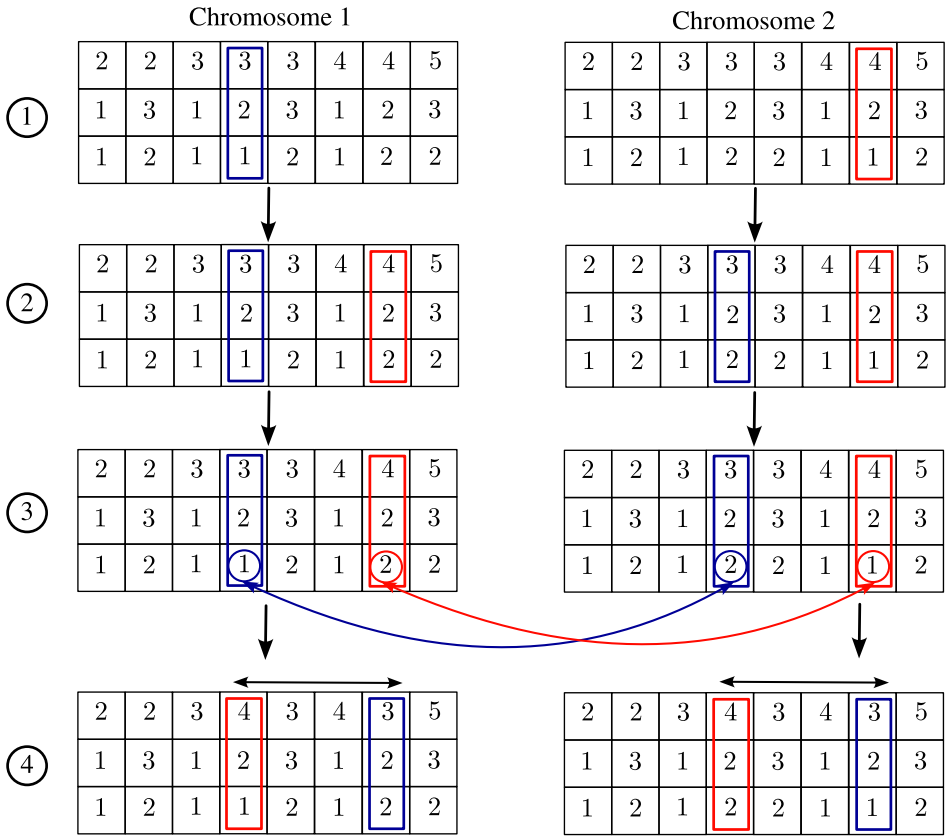


FIGURE 5. Crossover operation between two chromosomes.

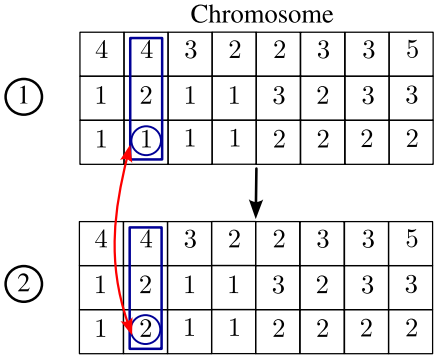


FIGURE 6. Mutation operation.

the mutation operator illustrated in Figure 6 consists of the following steps:

1) Randomly select a gene from the chromosome.
2) Change the robot assigned to perform the task represented by the selected gene to another robot equipped with the necessary sensor. This modification ensures that the task is assigned to a suitable robot.

By applying crossover and mutation operators to selected chromosomes, the GA explores different combinations of genetic material and introduces diversity into the population, facilitating the search for optimal or near-optimal solutions.

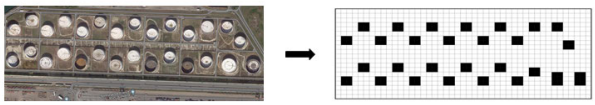


FIGURE 7. Binarization (right) of an industrial area (left) in Le Havre Port, France.

V. RESULTS AND DISCUSSION

A. SIMULATION SCENARIO

To demonstrate the results of the proposed method, we have selected an industrial area in the port of Le Havre City, France, known for its high-risk activities (Figure 7 left). The objective of the multi-robot mission in this area is to collect measurements for surveillance and intervention missions in case of a disaster. The mission involves three Unmanned Ground Vehicles (UGVs). In order to simulate the robot's mission, we converted an aerial view of the area into a 2D grid map, as shown in Figure 7 right.

The grid dimensions are $N_x = 45$ and $N_y = 20$, so, it is divided into 900 cells $\mathcal{V} = \{v_1, v_2, \ldots, v_{900}\}$ that

TABLE 5. Tasks required in each position.

location \ measure	m_1	m_2	m_3	m_4	m_5
a_1 (v_{878})	✗	✗	✗	✗	✗
a_2 (v_{634})	✗	✓	✓	✗	✗
a_3 (v_{772})	✓	✓	✓	✗	✓
a_4 (v_{637})	✓	✓	✓	✗	✗
a_5 (v_{733})	✗	✓	✗	✓	✓
a_6 (v_{509})	✗	✓	✓	✓	✗
a_7 (v_{562})	✗	✓	✗	✗	✗
a_8 (v_{666})	✓	✓	✓	✓	✓
a_9 (v_{310})	✓	✓	✓	✓	✗
a_{10} (v_{129})	✗	✗	✗	✗	✓
a_{11} (v_{248})	✓	✓	✓	✓	✗
a_{12} (v_{371})	✗	✗	✗	✗	✓
a_{13} (v_{233})	✗	✓	✗	✗	✓

TABLE 6. Capabilities of each robot.

robot \ sensor	m_1	m_2	m_3	m_4	m_5
r_1	✗	✓	✓	✗	✓
r_2	✓	✓	✓	✗	✓
r_3	✓	✗	✗	✓	✓

represent the environment. White cells represent open spaces where mobile robots can navigate and take measurements, whereas black cells represent structures or obstacles where robots cannot stay. Each cell is $30m \times 30m$ in size. One can observe that this setting allows for solving collision issues between robots within each cell at the navigation level. Furthermore, it is assumed that the elementary cost $\tilde{c}(j, j', k)$ equals 1 between each of two adjacent cells v_j and $v_{j'}$, otherwise $\tilde{c}(j, j', k) = \infty$ when $v_{j'}$ is an obstacle. Once the environment is converted to a graph, the Dijkstra algorithm will compute the costs and the trajectories between each pair of sites including the depot.

The tasks to be assigned to robots are distributed over twelve particular sites and $\mathcal{A} = \{a_1, \ldots, a_{13}\}$ where a_1 is the depot. Five types of measurements $\mathcal{M} = \{m_1, m_2, m_3, m_4, m_5\}$ are considered as follows:

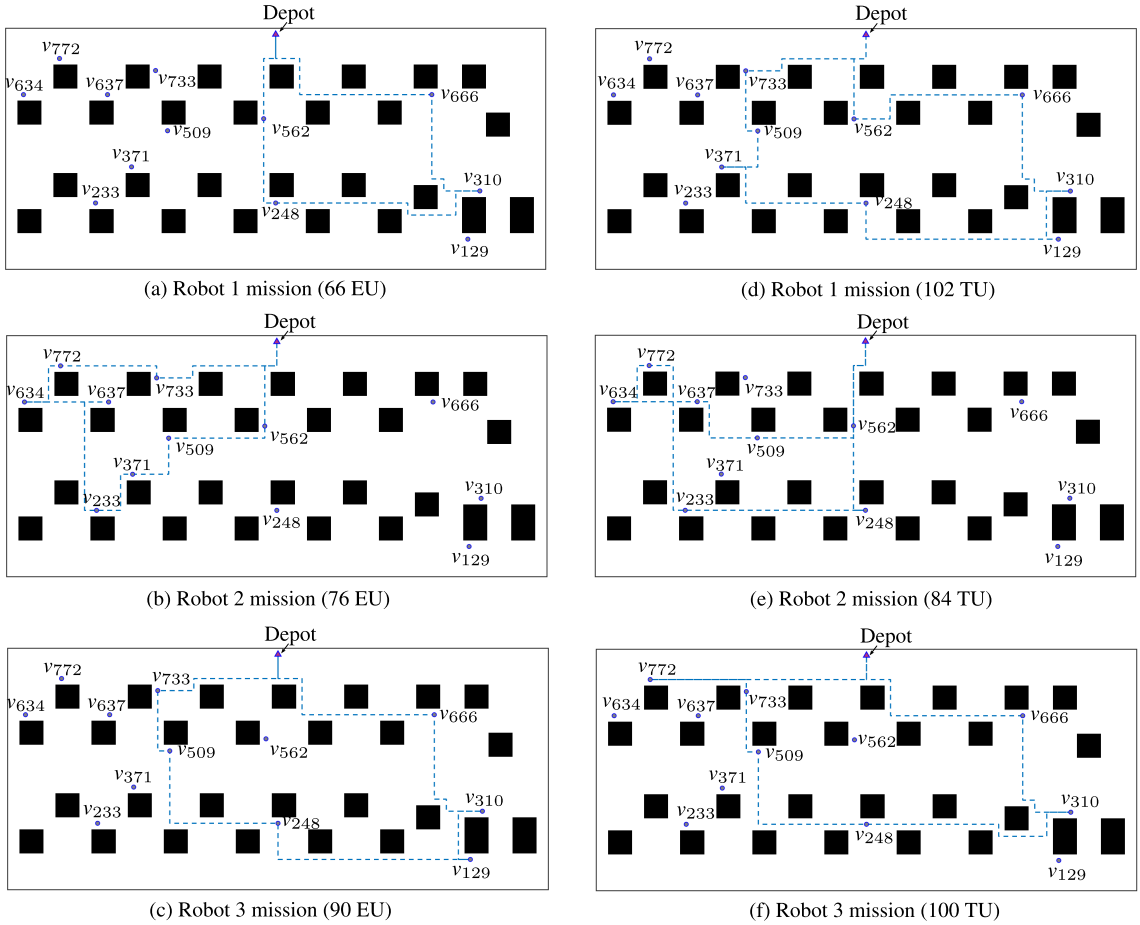
1) Detection of gas leak m_1: Non-Dispersive Infrared Sensors have been extensively used for gas leak detection. This device operates by detecting various light wavelengths absorbed by gases using infrared light [57].
2) Detection of a fire source m_2: a common method for the early detection of hot areas that could cause a fire danger is a thermal imaging camera technology [58].
3) Air quality measurement m_3: sensors that measure air quality are used to find airborne contaminants. This includes substances that could be hazardous to human health, such as particles, pollutants, and noxious gases (CO, CO_2, ...).
4) Temperature measurement m_4: the most popular kind of temperature sensors are thermocouples. They are employed in commercial, transportation, and residential purposes. Thermocouples offer short response times, can be useful over a wide temperature range, and are self-powered.
5) Shooting a specific area m_5: vision sensors that employ image processing are used for a variety of tasks, including sorting objects, identifying characters, measuring object size, and detecting defective objects.

This scenario is composed of thirty-three tasks $\mathcal{J} = \{j_1, \ldots, j_{33}\}$. Table 5 defines the multi-robot mission. Further, three UGVs $\mathcal{R} = \{r_1, r_2, r_3\}$ equipped with different sensors that can perform the measurements $\mathcal{M}$ constitute the team. The robot r_1 is equipped with three sensors that can perform m_2, m_3, and m_5. Besides, robot r_2 is equipped with four sensors for m_1, m_2, m_3, and m_5 respectively. The last robot r_3 is equipped with sensors for m_1, m_4, and m_5. Table 6 shows the capabilities of each robot (which defines the function $\mathcal{P}$).

GA (Genetic Algorithm) is utilized to determine the optimal sequence of tasks for each robot based on calculated costs, preliminary data of the state space (Table 5), and the characteristics of the robots (Table 6). The objective is to allocate tasks in order to minimize either the cumulative energy consumption of the fleet (MinSum objective function) or the mission time (MinMax objective function) in different phases of the mission.

In the first phase, the mission involves regular monitoring, and the objective is to minimize the cumulative energy consumption (The MinSum objective function is used). The robots start and end their missions at the same location v_{878} (symbolized as a triangle in Figure 8), and their paths are illustrated in blue dotted lines. Robot r_1 path includes v_{666}, v_{310}, v_{248}, and v_{562}. In these locations, it performs specific measurements according to the task sequence, such as $\{m_2, m_3, m_5)\}$ in v_{666}, $\{m_2, m_3)\}$ in v_{310}, $\{m_2, m_3)\}$ in v_{248}, and m_2 in v_{562}. The cumulative cost of this sequence is 66 EU, considering that its maximum battery range is $\mathcal{B}_1 = 95$ EU. Similarly, the computed sequence for r_2 is constituted of the following sites: v_{733}, v_{772}, v_{634}, v_{637}, v_{233}, v_{371},and v_{509} where measurements $\{m_2, m_5\}$ in v_{733}, $\{m_1, m_2, m_3, m_5\}$ in v_{772}, $\{m_2, m_3\}$ in v_{634}, $\{m_1, m_2, m_3\}$ in v_{637}, $\{m_2, m_5\}$ in v_{233}, m_5 in v_{371}, and $\{m_2, m_3\}$ in v_{509} are successively performed (the sequence cost is 76 EU with $\mathcal{B}_2 = 115$ EU). The last robot r_3 has a sequence that includes visits to v_{733}, v_{509}, v_{248}, v_{129}, v_{310}, and v_{666}. The performed measurements in each location are specified, such as m_4 in v_{733} and v_{509}, $\{m_1, m_4\}$ in v_{248}, m_5 in v_{129}, $\{m_1, m_4\}$ in v_{310} and v_{666}. The total cost for this sequence is 90 EU, which does not exceed the maximum battery range of robot r_3, $\mathcal{B}_3 = 115$ EU.

For the MinMax optimization phase, which focuses on completing the mission in the shortest possible time (in a disaster scenario for example), a solution that respects the autonomy constraints of the robots is computed. The solution, depicted in Figure 8, shows the trajectories of the robots and the visited sites. Further, Table 7 provides information about the execution time, fitness value, and corresponding GA parameters for each solution obtained during the optimization process.



(a) Robot 1 mission (66 EU)
(d) Robot 1 mission (102 TU)
(b) Robot 2 mission (76 EU)
(e) Robot 2 mission (84 TU)
(c) Robot 3 mission (90 EU)
(f) Robot 3 mission (100 TU)

FIGURE 8. Sequence of tasks found by GA for each robot: MinSum optimization (left), MinMax optimization (right).

TABLE 7. Execution time and fitness value for the two solutions.

Objective function	GA Parameters (τ_p,τ_g)	Execution time (s)	Cost
MinSum	(450, 100)	6.66	232 EU
MinMax	(450, 100)	6.76	102 TU

B. COMPARATIVE ANALYSIS

A comprehensive comparative study of three optimization algorithms, namely GA, Hybrid Filtered Beam Search (HFBS) [32], and Cplex, applied to solve the problem studied is presented in Table 8. The study encompasses various scenarios with differing numbers of tasks, robots, measurement types, and sites. Table 8 provides the cost and running time for each solver in different scenarios. By analyzing the results, we compare the performance of the solvers in terms of cost and observe how the cost values differ among the solvers, and assess the computational efficiency of each approach based on their respective running times.

First, let us introduce the HFBS approach: HFBS method is a heuristic derived from the A^* algorithm. It solves efficiently optimization problems in which the solutions are organized by a tree composed of several nodes. Each node represents a part of a potential solution. The algorithm operates step by step by estimating the cost at each step using a heuristic function. So, the evaluation function is the following: $f = g + h$, where g is the cost from the initial node to the actual one, and h is the heuristic function that estimates the cost from the actual node to the closest node that satisfies the problem criteria. Exploring only some specific branches of the search tree saves computational resources but returns sub-optimal solutions. The authors in [32] have used the HFBS which is based on a local filter β_l that retains only the best successors for each expanded node and a global filter β_g that restricts the number of nodes at each level of the search tree. Although this algorithm gives solutions in a short amount of time, it also has some disadvantages. Choosing the appropriate β_g and β_l is the main challenge in this approach, increasing these parameters doesn't guarantee the improvement of the solution. So, several tests must be done to find the more suitable parameters which vary according to the problem specifications. These tests will increase the execution time necessary to find a good solution. Moreover, the heuristic function must be chosen carefully since the quality of the solution depends principally on it.

TABLE 8. Comparative study.

Scenario				GA result			BS result			MILP result	
Robots (R)	Measurements (M)	Tasks (T)	Sites (A)	GA Parameters (τ_p, τ_g)	Cost (EU)	Time (s)	BS Parameters (β_g, β_l)	Cost (EU)	Time (s)	Cost (EU)	Time (s)
1	3	13	10	(20,30)	74	0.03	(20,4)	78	0.1	74	0.5
1	4	15	10	(20,30)	74	0.06	(3,3)	74	0.012	74	0.66
1	4	30	13	(200,80)	132	2.1	(23,19)	142	0.4	130	0.5
1	5	33	13	(300,80)	132	3.2	(23,6)	142	0.35	130	0.92
2	3	13	10	(30,30)	138	0.04	(11,8)	148	0.18	138	1.04
2	4	15	10	(30,30)	140	0.05	(5,4)	150	0.07	140	2.33
2	4	30	13	(250,80)	234	2.62	(22,20)	248	1.08	232	10.58
2	5	33	13	(350,80)	236	3.4	(27,10)	252	1.6	232	7.82
3	3	13	10	(40,30)	136	0.11	(21,5)	152	0.6	136	588
3	4	15	10	(40,30)	140	0.15	(16,8)	174	0.35	140	3.09
3	4	30	13	(450,100)	232	6.94	(25,21)	240	1.53	214	9.35
3	5	33	13	(550,100)	242	8.86	(12,7)	234	0.85	214	17.93
4	3	13	10	(80,50)	124	0.47	(18,2)	138	0.49	124	375.19
4	4	15	10	(80,50)	140	0.45	(28,25)	172	1.38	140	5.42
4	4	30	13	(650,100)	236	11.47	(9,7)	234	0.7	214	21.95
4	5	33	13	(750,100)	236	11.92	(24,9)	246	2.4	214	29.94

Table 8 shows the performance of the three methods. Simulations were run on an Intel(R) Core(TM) CPU i5 @ 2.6GHz. For each example, HFBS is executed several times with a random choice of the parameters β_g and β_l until the execution time of the GAs solution found is exceeded. Then, the best solution among them is selected and compared to the one generated by GA and Cplex. Note that the GA results were obtained with specific values of the mutation and crossover rates: $\tau_m = 20\%$ and $\tau_c = 80\%$. These values were determined through extensive experimentation, and they consistently produced the best solutions.

The GA results in Table 8 show that GA consistently generates optimal solutions for medium-sized problems, such as examples with 13 and 15 tasks. Afterward, increasing the size of the problem influences the quality of the solution of GA (scenarios with 30 and 33 tasks). GA struggles to handle larger and more complex scenarios efficiently. The GA's system becomes slower, its capacity for space search decreases, and GA can take a long time to find the best solution. In such cases, GA performs well in finding near-optimal solutions with relatively low-cost values across different scenarios. So, having a sub-optimal solution is the best alternative since finding the optimal one may require a huge population size τ_p and computation time.

The HFBS algorithm provides competitive results compared to GA. In many scenarios, the cost values obtained by HFBS are far from those of GA. However, there are a few situations where the cost values obtained by HFBS are similar to those of GA or better. It's worth noting that HFBS requires more execution time compared to GA to find competitive results.

The Cplex solver, which is a MILP solver, consistently produces low-cost solutions compared to GA and HFBS. This confirms that obviously, an exhaustive search based on MILP outperforms the other strategies in terms of the quality of solutions. However, Cplex tends to require significantly more execution time compared to the other two algorithms, especially for larger problem instances, as evident from the higher time values reported in Table 8.

The observed trends emphasize the trade-off between solution quality (cost) and computational efficiency (execution time) when dealing with larger and more complex problem instances. It suggests that finding optimal or near-optimal solutions becomes more difficult and computationally expensive as the problem size increases. Therefore, careful consideration and analysis of the problem characteristics and computational resources are necessary when selecting an appropriate method for solving large-scale optimization problems.

HFBS and GA both use parameter tuning. GA is characterized by its population size τ_p and the number of generations τ_g. Generally, larger values of τ_p and τ_g tend to yield better solutions. In Table 8, τ_p and τ_g were selected through an empirical process where different combinations of parameter values are tested on a set of problem instances. However, because of the fact that higher values for these parameters can indeed lead to good solutions, this test wasn't exhaustive. On the other hand, HFBS parameter tuning may require a lot of time in order to find good solutions. Larger β_g and β_l values do not always offer better solutions, there can be considerable divergence and thus the algorithm returns bad-quality solutions. A good choice of HFBS parameters remains an uncertain and challenging task that limits the use of this approach in general. In particular, repeated runs with different combinations of β_g and β_l without a time limitation can meet computational problems. Nevertheless, it is pretty unreasonable since the computing run would require a lot of time, and the algorithm will lose its efficiency.

Figure 9 illustrates the distribution of costs and running times for each method in the 16 scenarios of Table 8. It can be observed that the costs for the HFBS method are generally higher, followed closely by the GA method, while the Cplex

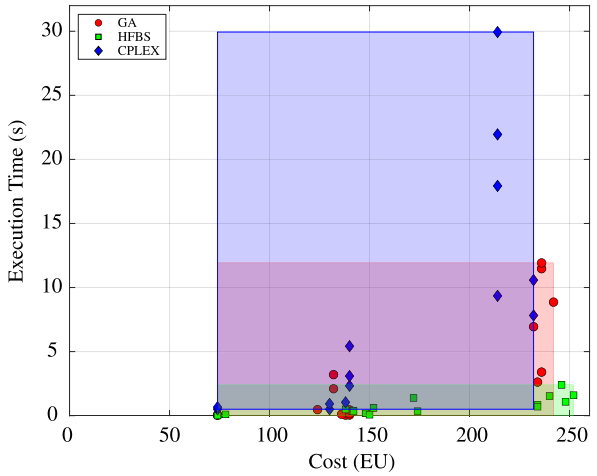


FIGURE 9. Distribution of cost and execution time for the three Different solvers across Various scenarios.

method has the lowest costs. On the other hand, the execution times for the GA and HFBS methods are relatively small compared to the Cplex method, which exhibits significantly higher execution times. Table 8 and Figure 9 allow us to evaluate the performance of the 3 solvers as follows:

When evaluating the trade-off between cost and execution time in the context of the solvers used in Table 6, it is clear that the GA consistently provides a balance between these two factors. GA solutions exhibit relatively low costs compared to the solutions generated by HFBS. Moreover, GA demonstrates shorter execution times compared to Cplex, indicating its efficiency in finding solutions within a reasonable time frame. GA's advantage in terms of execution time is particularly valuable in scenarios where time constraints are considered. The shorter running times of GA indicate its computational efficiency, making it well-suited for situations where obtaining a reasonably good solution in a timely manner is the primary objective. This efficiency is crucial in real-world applications where quick decision-making or responsiveness is required.

VI. CONCLUSION

This study aims to solve a multi-robot task allocation problem. The multi-robot mission consists of assigning optimally a set of tasks to a set of mobile sensing agents according to their capabilities to gather information in an industrial area. We presented a centralized evolutionary approach that solves the problem for two objective functions considered in this paper. The efficiency of the algorithm was illustrated through a simulation scenario and compared with a heuristic approach and MILP solver Cplex. It was shown that the proposed approach provides the best trade-off between optimality and execution time.

Future works will address several aspects including uncertainty like sensor malfunctions, motion planning, collision avoidance between robots, and the influence of each cost function on the performance of the method. Moreover, high-quality path planning techniques [59], [60] can significantly enhance the effectiveness of mission planning for unmanned equipment. These approaches can provide more efficient and adaptable trajectory planning solutions, which are particularly crucial when navigating through complex environments. Furthermore, the proposed method can be extended to deal with dynamic environments and unexpected events such as robot failures, and new coming tasks during the mission. So, an online assignment will be also one of the next topics of our research. Finally, developing algorithms to enable the robots to safely navigate to target positions in the environment should also be treated.

REFERENCES

[1] N. Paltrinieri and G. Reniers, "Dynamic risk analysis for Seveso sites," *J. Loss Prevention Process Industries*, vol. 49, pp. 111–119, Sep. 2017.
[2] C. Ju, J. Kim, J. Seol, and H. I. Son, "A review on multirobot systems in agriculture," *Comput. Electron. Agricult.*, vol. 202, Nov. 2022, Art. no. 107336.
[3] D. Portugal and R. Rocha, "A survey on multi-robot patrolling algorithms," in *Technological Innovation for Sustainability*, vol. 349, M. Luis and C. Matos, Eds. Berlin, Germany: Springer, 2011, pp. 139–146.
[4] B. Mishra, D. Garg, P. Narang, and V. Mishra, "Drone-surveillance for search and rescue in natural disaster," *Comput. Commun.*, vol. 156, pp. 1–10, Apr. 2020.
[5] H. G. Tanner, "Switched UAV-UGV cooperation scheme for target detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2007, pp. 3457–3462.
[6] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Mag.*, vol. 29, no. 1, p. 9, 2008.
[7] G. P. Das, T. M. McGinnity, S. A. Coleman, and L. Behera, "A distributed task allocation algorithm for a multi-robot system in healthcare facilities," *J. Intell. Robotic Syst.*, vol. 80, no. 1, pp. 33–58, Oct. 2015.
[8] D. W. Pentico, "Assignment problems: A golden anniversary survey," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 774–793, Jan. 2007.
[9] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logistics Quart.*, vol. 2, pp. 83–97, Mar. 1955.
[10] O. Cheikhrouhou and I. Khoufi, "A comprehensive survey on the multiple traveling salesman problem: Applications, approaches and taxonomy," *Comput. Sci. Rev.*, vol. 40, May 2021, Art. no. 100369.
[11] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *Eur. J. Oper. Res.*, vol. 88, no. 3, pp. 464–474, Feb. 1996.
[12] 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.
[13] 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.
[14] 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.
[15] J. K. Verma and V. Ranga, "Multi-robot coordination analysis, taxonomy, challenges and future scope," *J. Intell. Robotic Syst.*, vol. 102, no. 1, p. 10, May 2021.
[16] N. Seenu, R. M. K. Chetty, M. M. Ramya, and M. N. Janardhanan, "Review on state-of-the-art dynamic task allocation strategies for multiple-robot systems," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 47, no. 6, pp. 929–942, Sep. 2020.
[17] H. Chakraa, F. Guérin, E. Leclercq, and D. Lefebvre, "Optimization techniques for multi-robot task allocation problems: Review on the state-of-the-art," *Robot. Auto. Syst.*, vol. 168, Oct. 2023, Art. no. 104492.
[18] B. Miloradovic, B. Çürüklü, M. Ekström, and A. V. Papadopoulos, "Optimizing parallel task execution for multi-agent mission planning," *IEEE Access*, vol. 11, pp. 24367–24381, 2023.
[19] B. Miloradovic, B. Çürüklü, M. Ekström, and A. V. Papadopoulos, "GMP: A genetic mission planner for heterogeneous multirobot system applications," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 10627–10638, Oct. 2022.

[20] X. Gao, L. Wang, X. Yu, X. Su, Y. Ding, C. Lu, H. Peng, and X. Wang, "Conditional probability based multi-objective cooperative task assignment for heterogeneous UAVs," *Eng. Appl. Artif. Intell.*, vol. 123, Aug. 2023, Art. no. 106404.

[21] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, pp. 8091–8126, Oct. 2020.

[22] M. Karlsson, F. Ygge, and A. Andersson, "Market-based approaches to optimization," *Comput. Intell.*, vol. 23, no. 1, pp. 92–109, Feb. 2007.

[23] E. Nunes and M. Gini, "Multi-robot auctions for allocation of tasks with temporal constraints," *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2110–2216.

[24] L. Luo, N. Chakraborty, and K. Sycara, "Distributed algorithms for multirobot task assignment with task deadline constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 876–888, Jul. 2015.

[25] 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.

[26] 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.

[27] F. Ye, J. Chen, Q. Sun, Y. Tian, and T. Jiang, "Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints," *J. Supercomput.*, vol. 77, no. 1, pp. 111–132, Jan. 2021.

[28] L. Rabiner, "Combinatorial optimization: Algorithms and complexity," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-32, no. 6, pp. 1258–1259, Dec. 1984.

[29] S. Ismail and L. Sun, "Decentralized hungarian-based approach for fast and scalable task allocation," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 23–28.

[30] X. Zhou, H. Wang, B. Ding, T. Hu, and S. Shang, "Balanced connected task allocations for multi-robot systems: An exact flow-based integer program and an approximate tree-based genetic algorithm," *Exp. Syst. Appl.*, vol. 116, pp. 10–20, Feb. 2019.

[31] X.-F. Liu, B.-C. Lin, Z.-H. Zhan, S.-W. Jeon, and J. Zhang, "An efficient ant colony system for multi-robot task allocation with large-scale cooperative tasks and precedence constraints," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–8.

[32] M. Gam, A. J. Telmoudi, and D. Lefebvre, "Hybrid filtered beam search algorithm for the optimization of monitoring patrols," *J. Intell. Robotic Syst.*, vol. 107, no. 2, Feb. 2023.

[33] F. Nekovár, J. Faigl, and M. Saska, "Multi-tour set traveling salesman problem in planning power transmission line inspection," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6196–6203, Oct. 2021.

[34] J. Chen, J. Wang, Q. Xiao, and C. Chen, "A multi-robot task allocation method based on multi-objective optimization," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1868–1873.

[35] J. David and T. Rögnvaldsson, "Multi-robot routing problem with min–max objective," *Robotics*, vol. 10, no. 4, p. 122, Nov. 2021.

[36] F. Janati, F. Abdollahi, S. S. Ghidary, M. Jannatifar, J. Baltes, and S. Sadeghnejad, "Multi-robot task allocation using clustering method," in *Robot Intelligence Technology and Applications 4*. Cham, Switzerland: Springer, 2017, pp. 233–247.

[37] W. P. N. dos Reis, G. L. Lopes, and G. S. Bastos, "An arrovian analysis on the multi-robot task allocation problem: Analyzing a behavior-based architecture," *Robot. Auto. Syst.*, vol. 144, Oct. 2021, Art. no. 103839.

[38] L. E. Parker, "ALLIANCE: An architecture for fault tolerant multirobot cooperation," *IEEE Trans. Robot. Autom.*, vol. 14, no. 2, pp. 220–240, Apr. 1998.

[39] B. B. Werger and M. J. Mataric, "Broadcast of local eligibility: Behavior-based control for strongly cooperative robot teams," in *Proc. 4th Int. Conf. Auton. Agents*, 2000, pp. 21–22.

[40] A. Beynier and A.-I. Mouaddib, "Decentralized Markov decision processes for handling temporal and resource constraints in a multiple robot system," in *Distributed Autonomous Robotic Systems 6*. Tokyo, Japan: Springer, 2007, pp. 191–200.

[41] J. McCall, "Genetic algorithms for modelling and optimisation," *J. Comput. Appl. Math.*, vol. 184, no. 1, pp. 205–222, Dec. 2005.

[42] J. M. Palomo-Romero, L. Salas-Morera, and L. García-Hernández, "An island model genetic algorithm for unequal area facility layout problems," *Exp. Syst. Appl.*, vol. 68, pp. 151–162, Feb. 2017.

[43] G. Pinto, I. Ainbinder, and G. Rabinowitz, "A genetic algorithm-based approach for solving the resource-sharing and scheduling problem," *Comput. Ind. Eng.*, vol. 57, no. 3, pp. 1131–1143, Oct. 2009.

[44] A. Hiassat, A. Diabat, and I. Rahwan, "A genetic algorithm approach for location-inventory-routing problem with perishable products," *J. Manuf. Syst.*, vol. 42, pp. 93–103, Jan. 2017.

[45] A. Khan, Z. U. Rehman, M. A. Jaffar, J. Ullah, A. Din, A. Ali, and N. Ullah, "Color image segmentation using genetic algorithm with aggregation-based clustering validity index (CVI)," *Signal, Image Video Process.*, vol. 13, no. 5, pp. 833–841, 2019.

[46] Y. Lee, T. Hara, H. Fujita, S. Itoh, and T. Ishigaki, "Automated detection of pulmonary nodules in helical CT images based on an improved template-matching technique," *IEEE Trans. Med. Imag.*, vol. 20, no. 7, pp. 595–604, Jul. 2001.

[47] B. Lorenzo and S. Glisic, "Optimal routing and traffic scheduling for multihop cellular networks using genetic algorithm," *IEEE Trans. Mobile Comput.*, vol. 12, no. 11, pp. 2274–2288, Nov. 2013.

[48] K. Jose and D. K. Pratihar, "Task allocation and collision-free path planning of centralized multi-robots system for industrial plant inspection using heuristic methods," *Robot. Auton. Syst.*, vol. 80, pp. 34–42, Jun. 2016.

[49] H.-J. Choi, Y.-D. Kim, and H.-J. Kim, "Genetic algorithm based decentralized task assignment for multiple unmanned aerial vehicles in dynamic environments," *Int. J. Aeronaut. Space Sci.*, vol. 12, no. 2, pp. 163–174, Jun. 2011.

[50] P. K. Muhuri and A. Rauniyar, "Immigrants based adaptive genetic algorithms for task allocation in multi-robot systems," *Int. J. Comput. Intell. Appl.*, vol. 16, no. 4, Dec. 2017, Art. no. 1750025.

[51] P. Panchu K., M. Rajmohan, R. Sundar, and R. Baskaran, "Multi-objective optimisation of multi-robot task allocation with precedence constraints," *Defence Sci. J.*, vol. 68, no. 2, p. 175, Mar. 2018.

[52] S. Saeedvand, H. S. Aghdasi, and J. Baltes, "Robust multi-objective multi-humanoid robots task allocation based on novel hybrid metaheuristic algorithm," *Int. J. Speech Technol.*, vol. 49, no. 12, pp. 4097–4127, Dec. 2019.

[53] R. J. Alitappeh and K. Jeddisaravi, "Multi-robot exploration in task allocation problem," *Int. J. Speech Technol.*, vol. 52, no. 2, pp. 2189–2211, Jan. 2022.

[54] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[55] E. Schneider, E. I. Sklar, and S. Parsons, "Mechanism selection for multi-robot task allocation," in *Towards Autonomous Robotic Systems*. Berlin, Germany: Springer, 2017, pp. 421–435.

[56] T. Bektas, "The multiple traveling salesman problem: An overview of formulations and solution procedures," *Omega*, vol. 34, no. 3, pp. 209–219, Jun. 2006.

[57] S. Soldan, G. Bonow, and A. Kroll, "RoboGasInspector—A mobile robotic system for remote leak sensing and localization in large industrial environments: Overview and first results," *IFAC Proc. Volumes*, vol. 45, no. 8, pp. 33–38, 2012.

[58] J.-H. Hwang, S. Jun, S.-H. Kim, D. Cha, K. Jeon, and J. Lee, "Novel fire detection device for robotic fire fighting," in *Proc. ICCAS*, Oct. 2010, pp. 96–100.

[59] X. Wang, B. Li, X. Su, H. Peng, L. Wang, C. Lu, and C. Wang, "Autonomous dispatch trajectory planning on flight deck: A search-resampling-optimization framework," *Eng. Appl. Artif. Intell.*, vol. 119, Mar. 2023, Art. no. 105792.

[60] X. Wang, Z. Deng, H. Peng, L. Wang, Y. Wang, L. Tao, C. Lu, and Z. Peng, "Autonomous docking trajectory optimization for unmanned surface vehicle: A hierarchical method," *Ocean Eng.*, vol. 279, Jul. 2023, Art. no. 114156.

HAMZA CHAKRAA (Student Member, IEEE) received the M.S. degree in automation and robotics from the University of Montpellier, France, in 2021. He is currently pursuing the Ph.D. degree with the Research Group on Electrical Engineering and Automatic Control (GREAH), University of Le Havre Normandy, France. His research interests include planning, learning, and task allocation within the domain of multi-robot systems.

EDOUARD LECLERCQ received the B.S. degree in physics and mathematics from Paris Educational District, in 1987, the M.S. degree in electronics from the University of Rouen Normandy, France, in 1994, and the Ph.D. degree in automatic from the University of Le Havre Normandy, France, in 1999. Since 1999, he has been a Lecturer with the Department of Electronic, Electro-Technology and Automatic, Faculty of Sciences and Technology, University of Le Havre Normandy. He is also with the Research Group on Electrical Engineering and Automatic Control (GREAH), University of Le Havre Normandy. His current research interests include modeling, control, and fault detection using dynamical neural networks and Petri nets.

FRANÇOIS GUÉRIN received the Ph.D. degree in robotics and automatic control from the University of Le Havre Normandy, in 2004. He is currently an Assistant Professor with the Research Group on Electrical Engineering and Automatic Control (GREAH) and the Department of Electrical Engineering, Institute of Technology, University of Le Havre Normandy. His current research interest includes advanced automatic control and its applications to mobile robotics and electrical engineering.

DIMITRI LEFEBVRE (Senior Member, IEEE) received the degree from the École Centrale of Lille, France, in 1992, the Ph.D. degree in automatic control and computer science from the University of Sciences and Technologies, Lille, in 1994, and the HDR degree from the University of Franche-Comté, Belfort, France, in 2000. Since 2001, he has been a Professor with the Institute of Technology and the Faculty of Sciences, University of Le Havre Normandy, Le Havre, France. He is currently with the Research Group on Electrical Engineering and Automatic Control (GREAH), University of Le Havre Normandy, where he was the Head, from 2007 to 2012. His current research interests include Petri nets, learning processes, adaptive control, and fault detection and diagnosis.