# A Greedy Randomized Adaptive Search Procedure Variant for MRTA Problems with Multiple Depots

Chaima Baccouche[1,2], Edouard Leclercq[2], Achraf Jabeur Telmoudi[1], and Dimitri Lefebvre[2]

***Abstract*— This study presents a comprehensive approach to solving an advanced Multi-Robot Task Allocation (MRTA) problem, in which heterogeneous agents, initially stationed at different depots, are required to perform a set of spatially distributed tasks. A key challenge lies in determining which agent performs each task and defining the order in which each agent executes the assigned tasks, while optimizing travel costs and respecting energy constraints. To address this problem, we propose an approach that combines the Greedy Randomized Adaptive Search Procedure (GRASP) with 2-Opt local search. A comparative analysis was conducted against a Mixed Integer Linear Programming (MILP) solver and standard GRASP variants. The results demonstrate that the GRASP + 2-Opt approach strikes a favorable balance between optimality and execution time. This work provides practical insights for applications such as industrial inspection and environmental monitoring, where autonomous multi-robot coordination and energy constraints are essential for sustained and reliable operation.**

***Index Terms*— Multi-Robot Task Allocation (MRTA), Greedy Randomized Adaptive Search Procedure(GRASP), 2-Opt Local Search, Mixed-Integer Linear Programming (MILP).**

## I. INTRODUCTION

In the current era, the demand for autonomous robotic systems capable of collaborating in complex environments has grown significantly. From environmental monitoring [1] and industrial inspection [2] to exploration [3] and urban surveillance [4], modern applications demand coordinated and efficient deployment of multiple mobile robots. These systems are often composed of different types of agents, each suited to specific operational environments. Land-based robots, such as Automated Guided Vehicles (AGVs) and Unmanned Ground Vehicles (UGVs), are well suited for structured settings like warehouses and industrial sites. Aerial robots, commonly referred to as Unmanned Aerial Vehicles (UAVs), provide high mobility and rapid deployment capabilities, making them ideal for surveillance and monitoring tasks. Meanwhile, marine and underwater robots such as Unmanned Surface Vehicles (USVs) and Autonomous Underwater Vehicles (AUVs) play a crucial role in aquatic operations including maritime surveillance and environmental sampling. The increasing prevalence of Multi-Robot Systems (MRS) in these domains is driven by their numerous advantages, including greater operational flexibility, scalability, resilience to individual failures, and improved task throughput [5]. However, these benefits are closely tied to the ability to address the fundamental challenge of Multi-Robot Task Allocation (MRTA).

MRTA consists of optimally assigning a series of tasks to a team of robots while considering problem-specific constraints, with the objective of optimizing the overall efficiency of the system [6]. These constraints can include energy limitations notably battery capacity, agent compatibility, time windows, or environmental uncertainty. Determining which agent should perform each task and in what sequence, is a non-trivial problem, especially in large-scale, real-world scenarios. To address this challenge, a variety of methods have been explored in the literature. One widely adopted approach consists of formulating the MRTA problem as an Integer Linear Programming (ILP) or Mixed-Integer Linear Programming (MILP) model [6]. While these techniques yield optimal solutions, their computational complexity becomes a significant limitation as problem size increases. Due to the NP-hard nature of MRTA, research has increasingly focused on approximate methods to achieve near-optimal solutions more efficiently. In particular, heuristics and metaheuristics have shown strong potential for yielding high-quality solutions within acceptable computation times, enabling them to be well-suited for real-world applications. Population-based algorithms such as Particle Swarm Optimization (PSO) [7], Bee Colony Optimization (BCO) [8], Ant Colony Optimization (ACO) [9] and Genetic Algorithms (GA) [10] have demonstrated competitive performance across diverse MRTA formulations. Additionally, Simulated Annealing (SA) [11] and Tabu Search (TS) [12] employ stochastic exploration and memory-based search mechanisms. Recent trends in MRTA research highlight the growing adoption of hybrid strategies that leverage the complementary strengths of different algorithms to enhance solution quality, scalability, and robustness. These strategies include combinations of metaheuristics with exact methods, local search techniques, or fuzzy logic systems [13].

In this study, we investigate a scenario involving a fleet of heterogeneous agents, each initially stationed at a distinct depot and equipped with specialized sensors to perform dispersed measurement tasks within an industrial environment. The main objective is to minimize the total energy consumption resulting from agent movements while ensuring

[1] Chaima Baccouche and Achraf Jabeur Telmoudi are with The National Higher Engineering School of Tunis (ENSIT), University of Tunis, Laboratoire d'Ingenierie des Systemes Industriels et des Energies Renouvelables, Tunis BP 56-1008, Tunisia. Email: `achrafjabeur.telmoudi@ensit.rnu.tn`

[2] Chaima Baccouche, Edouard Leclercq, and Dimitri Lefebvre are with Université Le Havre Normandie, GREAH, 75 Rue Bellot, Le Havre, 76600, France. Emails: `chaima.baccouche@etu.univ-lehavre.fr`, `edouard.leclercq@univ-lehavre.fr`, `dimitri.lefebvre@univ-lehavre.fr`

the successful completion of all tasks. This setting introduces several challenges, notably deciding which agent should perform each task and determining the order in which each agent executes the assigned tasks, all while accounting for the limited energy autonomy of each agent. Unlike existing methods that address simplified variants of the MRTA problem, such as assuming homogeneous agents [14] or single depot configurations [15], our approach simultaneously considers heterogeneous agents, multiple depots, and energy constraints.

To tackle this problem, we propose an approach that combines the Greedy Randomized Adaptive Search Procedure (GRASP) [16] with a 2-Opt local search heuristic. Compared to population-based metaheuristics, the GRASP framework offers faster convergence and fewer parameters to tune, making it a lightweight yet effective choice for combinatorial optimization. However, it may require local improvement strategies, such as 2-Opt, to avoid entrapment in local optima, particularly in constrained or highly structured environments. To the best of our knowledge, this is the first application of GRASP to an MRTA formulation that integrates heterogeneous agents, multiple depots, and energy constraints. In addition, a Mixed-Integer Linear Programming (MILP) model is formulated and solved to evaluate solution quality and serve as a reference for small- to medium-scale instances.

The key contributions of this study are outlined below:

- Formulation of a novel variant of the MRTA problem that incorporates agent heterogeneity, multiple depots, and energy constraints, reflecting realistic requirements in industrial and environmental missions;
- Development of a MILP model to obtain exact solutions for small- to medium-sized instances, serving as a baseline for performance comparison;
- Design of a GRASP algorithm combined with 2-Opt local search to enable scalable and efficient solution generation for larger problem instances;
- Validation across diverse scenarios, highlight the effectiveness, robustness, and practical relevance of the proposed approach in addressing a novel variant of the MRTA problem.

The remainder of this paper is organized as follows: the second section introduces the problem and presents the MILP formulation. The third section details the materials and methodology employed in this work. The fourth section discusses the experimental findings. Lastly, the final section highlights the main contributions and proposes directions for future research.

## II. MODELING ASPECTS

In response to the challenges outlined previously, this section introduces a Mixed-Integer Linear Programming (MILP) model to formally represent the MRTA problem, characterized by heterogeneous agents, multiple depots, and energy constraints.

### A. Problem description

This paper addresses the challenge of assigning a set of measurement tasks to a group of agents, each initially positioned at a distinct depot and subject to specific energy limitations. The objective is to optimize the total energy consumption, defined as the cost function.

The environment is modeled as a mesh grid composed of individual cells. Each cell $(x, y)$ represents a area where agents may be positioned. As a result, the environment is represented into a set of $V$ cells, where the dimensions of the cells are defined according to the specific requirements of the problem. This set is denoted as $V = v_1, \ldots, v_j, \ldots, v_{|V|}$. Furthermore, the environment may contain various obstacles or one-way paths that limit agent mobility. Within this environment, certain cells, referred to as sites, are designated for specific functions. Among these, some are reserved as depots, where each agent starts and completes its mission; this set is denoted by $D$. Others are designated for specialized activities such as the execution of measurement tasks; this set is denoted by $P_m$. The complete set of sites is therefore defined as $S = D \cup P_m$. Let $A$ denote the set of agents. The agents can be heterogeneous, with their heterogeneity arising from the specific sensors they are equipped with, as well as from differences in their maximum battery capacities $C^k_{max}$. Each depot $d_k \in D$ is associated with a specific agent $a_k \in A$, uniquely defined by its sensor configuration and energy capacity (Battery Swapping). As a result, each agent must return to its corresponding depot at the end of its mission. Each sensor corresponds to a particular measurement type, and the set of measurement types is denoted by $M$. The capabilities of the agents are defined by a matrix $Agents_{k,q}$, where $Agents_{k,q} = 1$ if the agent $a_k \in A$ is provided with the sensor necessary to perform the measurement of type $m_q \in M$ and $Agents_{k,q} = 0$ otherwise. A task is explicitly outlined as the execution of a specific measurement of type $m_q \in M$ at a designated location $p_j \in Pm$, represented as $t = (m_q, p_j)$. The set of tasks to be performed at each site is represented by the matrix $Tasks_{j,q}$, where $Tasks_{j,q} = 1$ indicates that a measurement of type $m_q$ is required at site $p_j \in P_m$, and $Tasks_{j,q} = 0$ otherwise. The minimum cost $c(i, j)$ required for any agent to move from site $p_i$ to site $p_j$ is calculated using the Dijkstra algorithm [17].

The assumptions considered in this study are as follows:

- The environment is fully known in advance and remains static during the execution of the missions, no dynamic changes or uncertainties are considered;
- Each agent begins and concludes its mission at the designated depot $d_k$, where $d_k$ is the depot assigned to agent $a_k$;
- Each task is performed by a single agent $a_k$;
- Each agent is constrained by its energy limit $C^k_{\text{max}}$.

### B. MILP formalization

The problem can be formulated as a Mixed-Integer Linear Programming (MILP) model.

The following decision variables are defined:

- $x_{i,j}^k$: binary decision variable such that $x_{i,j}^k \in \{0,1\}$, where $x_{i,j}^k = 1$ if agent $a_k$ travels from site $p_i$ to site $p_j$, and $x_{i,j}^k = 0$ otherwise;
- $u_i^k$: integer decision variable representing the order in which agent $a_k$ visits site $p_i$, defined as $u_i^k \in \mathbb{N}$ with bounds $0 \leq u_i^k \leq |S|$.

The problem is formalized as follows:

$$\text{minimize} \quad \sum_{a_k \in A} \sum_{p_i \in S} \sum_{p_j \in S, j \neq i} x_{i,j}^k \times c(i,j) \tag{1}$$

subject to:

$$\sum_{a_k \in A} \sum_{p_j \in P_m} x_{i,j}^k = \sum_{a_k \in A} \sum_{p_j \in P_m} x_{j,i}^k, \quad \forall p_i \in D \tag{2}$$

$$\sum_{p_j \in P_m} x_{d_k,j}^k \leq 1, \quad \forall a_k \in A \tag{3}$$

$$\sum_{p_j \in P_m} x_{i,j}^k = 0, \quad \forall i \in D, i \neq d_k, \forall a_k \in A \tag{4}$$

$$\sum_{p_i \in P_m} x_{i,d_k}^k \leq 1, \quad \forall a_k \in A \tag{5}$$

$$\sum_{p_i \in P_m} x_{i,j}^k = 0, \quad \forall j \in D, j \neq d_k, \forall a_k \in A \tag{6}$$

$$\sum_{p_i \in S} x_{i,j}^k = \sum_{p_i \in S} x_{j,i}^k, \quad \forall p_j \in P_m, \forall a_k \in A \tag{7}$$

$$x_{i,i}^k = 0, \quad \forall a_k \in A, \forall p_i \in S \tag{8}$$

$$\sum_{a_k \in A} \sum_{p_i \in S} Agents_{k,q} \times x_{i,j}^k \geq Tasks_{j,q}, \quad \forall p_j \in P_m, \forall m_q \in M \tag{9}$$

$$\sum_{p_i \in S} \sum_{p_j \in S} x_{i,j}^k \times c(i,j) \leq C_{max}^k, \quad \forall a_k \in A \tag{10}$$

$$u_i^k - u_j^k + x_{i,j}^k \leq (|S|-1)(1 - x_{i,j}^k), \quad \forall a_k \in A, \forall p_i \in S, \forall p_j \in P_m \tag{11}$$

The objective, defined in equation (1), aims to minimize the total displacement cost across all agents throughout their missions. Equation (2) enforces flow conservation at the depots by ensuring that the number of agents departing from each depot equals the number returning to it. Equations (3) and (4) guarantee that each agent starts its mission from its designated depot and is not allowed to depart from any other. Similarly, equations (5) and (6) ensure that each agent returns to its designated depot and cannot end its mission at any other depot. Continuity of movement is imposed by equation (7), which requires that if an agent visits a task site, it must also leave it. Equation (8) eliminates self-loops by preventing agents from visiting the same site consecutively. Equation (9) ensures that measurement task is completed by at least one capable agent. To account for energy limitations, equation (10) restricts the total travel cost of each agent's route such that it does not exceed its maximum energy capacity $C_{max}^k$. Finally, equation (11) eliminates the possibility of sub-circuits within the set of sites $S$, thereby ensuring route continuity and correctness.

## III. MATERIALS AND METHODOLOGY

To address the complexity of the problem setting, this study adopts a approach by combining the Greedy Randomized Adaptive Search Procedure (GRASP) with the 2-Opt local search algorithm. GRASP involves a greedy randomized construction phase, which is followed by a local search phase. Within the local search, two distinct neighborhood operators swap and mutation are implemented and tailored to MRTA context to evaluate their respective impacts on solution quality. The following subsections present the materials of the proposed approach in detail, along with the methodology used for implementation.

### A. *Greedy Randomized Adaptive Search Procedure (GRASP)*

GRASP, first introduced by Feo and Resende in 1989 [16], is a metaheuristic designed to tackle combinatorial optimization problems [18]. It is classified as a single-solution metaheuristic, meaning that at any given time, the algorithm maintains a single solution, which it attempts to iteratively improve by exploring its neighborhood.

The pseudo-code of the GRASP algorithm is provided in Algorithm 1 (main loop), Algorithm 2 (construction phase), and Algorithm 3 (local search). The main loop (Algorithm 1) is governed by two key parameters: the maximum number of iterations $N$ and the greediness threshold $\alpha \in [0,1]$, which governs the trade-off between greediness and randomness during the solution construction process (construction phase). In fact, $\alpha$ governs Algorithm 2. The number of iterations $N$ is a key factor in the algorithm's performance, as it enables both the refinement of solutions within each iteration and the exploration of different regions of the solution space. This process increases the probability of escaping local optima and discovering globally optimal or near-optimal solutions. Therefore, the choice of this parameter is very important. In each iteration, the algorithm performs two phases: Greedy Randomized Construction (GRC) and Local Search (LS). The GRC phase generates a feasible solution, which is subsequently enhanced by the LS phase through neighborhood exploration. If the improved solution is better than the current best solution, it replaces the latter.

**Algorithm 1** GRASP($N, \alpha$)

```
Best_Solution ← ∅
for k = 1 to N do
    Solution ← Greedy_Randomized_Construction(α)
    Solution ← Local_Search(Solution)
    Best_Solution ← Update_Solution(Solution, Best_Solution)
end
return Best_Solution
```

The construction phase begins with an empty solution and progressively incorporates components until a complete solution is formed. The pseudo-code for the GRC phase is presented in Algorithm 2. Let $e$ denote an element of the problem, and let $E$ be the finite set of all such elements. The set $C$ denotes the candidate set, and the cost associated with adding an element $e$ is represented by $c(e)$. The Restricted Candidate List (RCL) consists of elements from $C$ whose costs are relatively low, specifically those satisfying a threshold condition based on minimum cost $c^{\min}$, maximum cost $c^{\max}$, and the greediness parameter $\alpha$.

**Algorithm 2** Greedy_Randomized_Construction($\alpha$)

```
Solution ← ∅
Initialize the candidate set: C ← E
Evaluate the incremental cost c(e) for all e ∈ C
while C ≠ ∅ do
    c^min ← min{c(e) | e ∈ C}
    c^max ← max{c(e) | e ∈ C}
    RCL ← {e ∈ C | c(e) ≤ c^min + α(c^max − c^min)}
    Select randomly an element e from RCL
    Solution ← Solution ∪{e}
    Update the candidate set C
    Reevaluate the incremental costs c(e) for all e ∈ C
end
return Solution
```

In the context of our problem, the element $e$ represents a feasible assignment of a measurement task to an agent, taking into account energy constraints, task definition, and agent compatibility. The candidate set $C$ is formed by selecting unassigned tasks that the agent can feasibly execute. The cost function $c(e)$ represents the travel cost from the agent's current location to the location of the candidate task, optionally accounting for the return cost to the depot if required. The values $c^{\min}$ and $c^{\max}$ are computed from these travel costs, and reflect the minimum and maximum incremental costs among all candidates in $C$. The Restricted Candidate List (RCL) contains tasks that are both feasible for the agent and among the least costly.

**Algorithm 3** Local_Search(Solution)

```
while Solution is not at a local optimum do
    Find s′ ∈ N(Solution) with f(s′) < f(Solution)
    Update Solution to s′
end
return Solution
```

Algorithm 3 presents the pseudo-code for the LS phase. This algorithm iteratively explores the neighborhood of a given solution, denoted as $\mathcal{N}$(Solution), which includes all nearby solutions that can be reached through small modifications. The objective function $f(\cdot)$ represents a quantity to be minimized, such as total energy consumption or travel cost. In each iteration, the algorithm checks if a neighboring solution $s' \in \mathcal{N}$(Solution) exists with a better objective value ($f(s') < f$(Solution)). If such a solution is found, it becomes the new current solution. This process repeats until no additional improvements can be achieved, signifying that a local optimum has been attained. To evaluate optimality, the algorithm compares the objective value of the current solution with those of its neighbors. Optimality is considered achieved when no neighboring solution yields a better objective value. The improved solution derived at the end of this phase is then returned.

In the literature, various neighborhood structures have been proposed for local search procedures, each offering different trade-offs between exploration and exploitation. In this study, two distinct neighborhood structures swap and mutation, were utilized and adapted to the MRTA problem with energy constraints, where tasks can be performed by multiple agents. While the swap operator follows a conventional exchange-based approach, the mutation operator modifies task assignments through unilateral reassignment, resulting in structurally different neighborhood transitions.

- Swap: this operator selects two tasks assigned to different agents and attempts to exchange them, provided the swap maintains feasibility.
- Mutation: this operator reassigns a task from one agent to another agent with the same capability. The mutation operator alters the assignment by moving a task unidirectionally from one agent to another.

For both operators, feasibility is ensured by verifying agent compatibility and energy constraints. A swap or mutation is accepted only if it results in a reduction of the total solution cost.

### B. 2-Opt Local Search

The 2-Opt method, introduced by Croes in 1958 [19], is a local search technique designed for the Traveling Salesman Problem (TSP). Starting with an initial solution, whether provided or randomly generated, the algorithm iteratively swaps two edges with another pair to eliminate path crossings and decrease the total tour length. The process continues until no further improvement is possible, that is, when all possible edge swaps have been tested and no shorter tour is found.

**Algorithm 4** 2-Opt Algorithm (tour, $c(\cdot,\cdot)$)

```
n ← size(tour)
while an improving 2-Opt move exists do
    for i ← 1 to n − 3 do
        for j ← i + 2 to n − 1 do
            if c(i, i+1) + c(j, j+1) > c(i, j) + c(i+1, j+1) then
                Reverse sub-tour between i+1 and j
                Mark improvement found
            end
        end
    end
end
return tour
```

This method is used as a local refinement step within the GRASP framework. It does not alter the task assignments determined during the GRASP phases but focuses on reordering the sequence of tasks assigned to each agent in order to minimize their individual displacement costs. This integration helps eliminate inefficiencies such as suboptimal task sequences or crossed paths, thereby improving the overall solution quality without compromising feasibility.

## IV. RESULTS AND DISCUSSION

To demonstrate the effectiveness of the proposed methodology, we evaluate its performance with numerical experiments across two different industrial zones [10], [6], each composed of a varying number of sites $|S|$ . Ten scenarios are considered for each environment to assess the robustness of the approaches under energy constraints. Each environment is characterized by the number of agents $|A|$, the number of tasks, and different values of $C^k_{\max}$. A key objective of these experiments is to verify that the energy constraint is properly enforced. Specifically, each agent's displacement cost must not exceed its individual maximum battery capacity. To assess this, the value of $C^k_{\max}$ was systematically varied, starting with large values and gradually decreasing them. This enabled the evaluation of each approach under both relaxed and stringent energy constraints and helped identify the threshold beyond which the problem becomes infeasible.

To provide a comprehensive evaluation of solution strategies for the proposed MRTA problem, four approaches are presented. GRASP with a swap operator and GRASP with a mutation operator were implemented to assess the effectiveness of different neighborhood transition mechanisms, each introducing distinct perturbations to diversify the search. The third approach, GRASP with mutation combined with 2-Opt local search, offers a more refined search process. Finally, an exact MILP solver is used as a reference method for small- and medium-scale instances, enabling the assessment of metaheuristic solution quality relative to optimal or near-optimal results. Table I presents the total energy cost obtained by each method across all scenarios. A dash (–) indicates that no feasible solution was found.

In the first experiment (13 sites), all three methods produced identical results for several scenarios. In Scenario 1, each method yields a total cost of 202 Energy Units (EU), demonstrating consistency across solvers under relaxed energy constraints. In Scenario 3, all methods converge to a cost of 160 EU, and in Scenario 4, each returns 168 EU. However, in Scenarios 2 and 5, none of the methods find a feasible solution. This infeasibility directly results from the strict energy constraint $C^k_{\max}$, confirming that all methods correctly enforce the requirement that each agent respects its maximum battery capacity. Additionally, it highlights that the only agent with enough energy is unable to complete all the missions, further reinforcing the importance of energy limits in task allocation. These results suggest that both standard GRASP variants (with swap and mutation operators) and the GRASP(Mut.) + 2-Opt approach are capable of producing high-quality and potentially optimal solutions under energy constraints. On the other hand, in this particular setting, no conclusion can be drawn regarding which method is more effective, as all approaches yield identical costs when feasible. In the second experiment (24 sites), the methods show more differences in performance. In Scenario 6, all GRASP variants return feasible solutions, GRASP(Mut.) + 2-Opt yielding the lowest total cost of 220 EU, compared to 232 EU for the other GRASP variants. The MILP solver, however, fails to find a solution, highlighting the scalability limitations of exact approaches for larger instances. In Scenario 8, the performance gap becomes more pronounced: GRASP(Mut.) + 2-Opt again outperforms both standard GRASP configurations, returning a cost of 206 EU compared to 230 EU (improvement of about 10%). These results suggest that the local refinement provided by 2-Opt plays a crucial role in guiding the search toward more cost-effective solutions. Scenario 9 follows a similar pattern, with GRASP(Mut.) + 2-Opt achieving a cost of 210 EU, outperforming the standard GRASP at 234 EU (improvement of about 10%). However, in Scenarios 7 and 10, none of the methods succeeds in finding a feasible solution. This again reflects the strict enforcement of the energy constraint $C^k_{\max}$, consistent with the observations from the first experiment.

The results for both industrial experiments confirm the validity and robustness of the proposed methodology. All methods strictly adhered to the energy constraint $C^k_{\max}$ and GRASP(Mut.) + 2-Opt yields better-quality solutions as the complexity of the scenario increases. The MILP approach, although optimal in theory, is limited to smaller instances due to its computational demands. These findings validate the effectiveness of metaheuristics in handling MRTA problems, especially when exact methods become impractical.

TABLE I
COST FOR VARIOUS SCENARIOS

| Experiments | | | | | GRASP | | GRASP(Mut.)+2Opt | MILP |
|---|---|---|---|---|---|---|---|---|
| $\|S\|$ | $\|A\|$ | Tasks | Sc. | $C^k_{\max}$ | Swap | Mut. | | |
| 13 | 2 | 11 | 1 | 1000<br>1000<br>1000 | 202 | 202 | 202 | 202 |
| | | | 2 | 1000<br>10 | - | - | - | - |
| | 3 | 10 | 3 | 1000<br>1000<br>1000 | 160 | 160 | 160 | 160 |
| | | | 4 | 1000<br>1000<br>10 | 168 | 168 | 168 | 168 |
| | | | 5 | 1000<br>10<br>10 | - | - | - | - |
| 24 | 2 | 22 | 6 | 1000<br>1000 | 232 | 232 | 220 | - |
| | | | 7 | 1000<br>10 | - | - | - | - |
| | 3 | 21 | 8 | 1000<br>1000<br>1000 | 230 | 230 | 206 | - |
| | | | 9 | 1000<br>1000<br>10 | 234 | 234 | 210 | - |
| | | | 10 | 1000<br>10<br>10 | - | - | - | - |

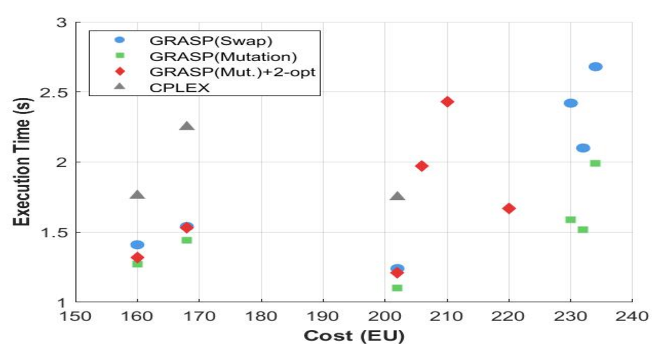


Fig. 1. Performance comparative study

To offer a thorough comparison of the strategies examined, Figure 1 depicts the trade-off between execution time and solution cost for each method across all test scenarios.

The results reveal several key insights. First, the GRASP(Mut.) + 2-Opt approach demonstrates a consistently favorable balance between solution cost and computational time. Although its execution time is slightly higher than that of the other metaheuristic variants, it consistently yields lower cost values, indicating improved solution quality. Compared to the MILP approach, GRASP(Mut.) + 2-Opt delivers near-optimal solutions in significantly shorter runtimes, particularly for larger problem instances where MILP struggles or fails to obtain acceptable solutions within the specified time constraints. This highlights the effectiveness of combining GRASP(Mut.) with 2-Opt, as the latter enhances local exploitation capabilities. In contrast, the GRASP(Swap) method generally exhibits longer execution times and higher solution costs, especially in larger problem instances. This suggests that its neighborhood exploration mechanism may be less effective in identifying high-quality solutions within the available computational time. By comparison, the GRASP(Mut.) variant offers faster runtimes but yields higher costs than both MILP and GRASP(Mut.) + 2-Opt. The MILP approach (solved using CPLEX) shows the most variable performance. While it provides competitive results for small- to medium-sized instances, it becomes impractical in larger scenarios due to excessive computation times or its inability to find feasible solutions within the time constraints. These findings underscore the scalability limitations of exact methods in real-world environments. Overall, the GRASP-based methods clearly outperform the MILP approach in terms of execution time. Among them, GRASP(Mut.) + 2-Opt emerges as the most promising compromise between solution quality and computational efficiency. It consistently delivers high-quality solutions within acceptable runtimes, offering a robust and scalable alternative to exact methods, which tend to struggle as problem complexity increases.

## V. CONCLUSION

This work has presented a comprehensive approach to an extended variant of the MRTA problem, characterized by heterogeneous agents, multiple depots, and energy constraints. To address this challenge, a hybrid method was introduced that integrates GRASP with 2-Opt local search, aiming to construct efficient and high-quality solutions.

The proposed approach was evaluated through a comparative analysis against a MILP solver, commonly used in related research, and a standard GRASP variant. Experiments conducted across diverse scenarios confirmed the method's robustness and effectiveness in minimizing energy consumption, while also satisfying task allocation and agent-related constraints within a reasonable time frame. However, the proposed approach is currently applicable only to static environments.

Our future research will focus on extending the current framework to dynamic environments characterized by evolving tasks and uncertainties, as well as exploring advanced metaheuristic strategies such as Particle Swarm Optimization (PSO). Additionally, we aim to investigate distributed approaches to further enhance scalability, adaptability, and solution quality in large-scale, decentralized settings. Finally, we aim to extend our approaches to other classes of robotic systems [20].

## REFERENCES


[1] Matarić, M. J., Sukhatme, G. S., and Østergaard, E. H. (2003). Multi-robot task allocation in uncertain environments. Autonomous Robots, 14, 255-263.

[2] Liu, C., and Kroll, A. (2012). A centralized multi-robot task allocation for industrial plant inspection by using A* and genetic algorithms. In Artificial Intelligence and Soft Computing: 11th International Conference, ICAISC 2012, Zakopane, Poland, April 29-May 3, 2012, Proceedings, Part II 11 (pp. 466-474). Springer Berlin Heidelberg.

[3] Alitappeh, R. J., and Jeddisaravi, K. (2022). Multi-robot exploration in task allocation problem. Applied Intelligence, 52(2), 2189-2211.

[4] Williams, R. K., Gasparri, A., and Ulivi, G. (2017). Decentralized matroid optimization for topology constraints in multi-robot allocation problems. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 293-300). IEEE.

[5] Nie, Z., Chen, K. C. (2021). Hypergraphical real-time multirobot task allocation in a smart factory. IEEE Trans. on Industrial Informatics, 18(9), 6047-6056.

[6] Baccouche, C., Ammar, I. I., Lefebvre, D., and Telmoudi, A. J. (2024). A preliminary study about multi-robot task allocation with energy constraints. In 2024 IEEE 20th International Conference on Automation Science and Engineering (CASE) (pp. 3057-3062). IEEE.

[7] Chen, J., Wang, J., Xiao, Q., and Chen, C. (2018, November). A multi-robot task allocation method based on multi-objective optimization. In 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV) (pp. 1868-1873). IEEE.

[8] Kurdi, H., Al-Megren, S., Aloboud, E., Alnuaim, A. A., Alomair, H., Alothman, R., Youcef-Toumi, K. et al. (2020). Bee-inspired task allocation algorithm for multi-UAV search and rescue missions. International Journal of Bio-Inspired Computation, 16(4), 252-263.

[9] Wang, J., Gu, Y., and Li, X. (2012). Multi-robot task allocation based on ant colony algorithm. J. Comput., 7(9), 2160-2167.

[10] Chakraa, H., Leclercq, E., Guérin, F., and Lefebvre, D. (2023). A centralized task allocation algorithm for a multi-robot inspection mission with sensing specifications. IEEE Access, 11, 99935-99949.

[11] David, J., and Rögnvaldsson, T. (2021). Multi-robot routing problem with min–max objective. Robotics, 10(4), 122.

[12] Nekovář, F., Faigl, J., and Saska, M. (2021). Multi-tour set traveling salesman problem in planning power transmission line inspection. IEEE Robotics and Automation Letters, 6(4), 6196-6203.

[13] Patel, H. R., and Shah, V. A. (2022). Shadowed type-2 fuzzy sets in dynamic parameter adaption in cuckoo search and flower pollination algorithms for optimal design of fuzzy fault-tolerant controllers. Mathematical and Computational Applications, 27(6), 89.

[14] Nikitenko, A., Lavendelis, E., Ekmanis, M., and Rumba, R. (2018). Task allocation methods for homogeneous multi-robot systems: feed pushing case study. Automatic Control and Computer Sciences, 52, 371-381.

[15] Gam, M., Telmoudi, A. J., and Lefebvre, D. (2023). Hybrid filtered beam search algorithm for the optimization of monitoring patrols. Journal of Intelligent and Robotic Systems, 107(2), 26.

[16] Feo, T. A., and Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. Operations research letters, 8(2), 67-71.

[17] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Math., 1(1), 269–271.

[18] Gendreau, M., and Potvin, J. Y. (Eds.). (2010). Handbook of metaheuristics (Vol. 2, p. 9). New York: Springer.

[19] Croes, G. A. (1958). A method for solving traveling-salesman problems. Operations research, 6(6), 791-812.

[20] Nzue, R. M. A., Brethé, J. F., Vasselin, E., and Lefebvre, D. (2013). Comparison of serial and parallel robot repeatability based on different performance criteria. Mechanism and Machine Theory, 61, 136-155.