Energy-Aware Optimization of Multi-Robot Systems with Task Allocation and Partial Recharge Scheduling

[*]*Germain Junior AVOSSEVOU*
CESI LINEACT
Campus Le Mans, France
gjavossevou@cesi.fr

[†]*Ahmed NAIT CHABANE*
CESI LINEACT
Campus Saint-Nazaire, France
anaitchabane@cesi.fr

[‡]*Ouahib GUENOUNOU*
CESI LINEACT
Campus Le Mans, France
oguenounou@cesi.fr

[§]*Karim BEDDIAR*
CESI LINEACT
Campus Nantes, France
kbeddiar@cesi.fr

***Abstract*—This paper presents a task allocation and scheduling model for multi-robot systems operating under energy constraints. The proposed model integrates key factors such as energy consumption, battery charging management, and task execution efficiency. To address this problem, we employ both an exact solver-based method and a bio-inspired algorithm, enabling a comparative analysis of their performance in terms of solution quality and computational efficiency. Experimental results highlight the trade-offs between execution time, energy consumption, and task scheduling efficiency. These insights contribute to optimizing task scheduling in multi-robot systems while minimizing energy expenditure. The proposed framework offers a promising solution for balancing energy constraints and task efficiency in collaborative robotic missions.**

Keywords— Multi-Robot Systems, Task Allocation, Energy Optimization, Scheduling, Mixed-Integer Linear Programming, Genetic Algorithm, Partial Recharge Management

I. Introduction

Autonomous multi-robot systems are increasingly deployed in applications such as industrial inspection, disaster response, and environmental monitoring [1], [2]. These systems must not only allocate tasks efficiently but also manage energy consumption and ensure timely mission completion, particularly when operating under tight battery constraints and limited recharging infrastructure. Traditionally, multi-robot task allocation (MRTA) approaches focus on minimizing total distance or aggregate execution time. However, in mission-critical environments, the key performance indicator is often the makespan, i.e., the time by which all inspection tasks are completed and the robots return to base. This shifts the optimization focus from aggregate cost to the earliest possible completion of all required measurements.

Managing energy consumption becomes especially critical when each task—such as on-site measurements—consumes additional energy beyond locomotion. Robots must therefore not only plan feasible paths but also anticipate energy depletion, determine whether and when to recharge, and ensure that they maintain enough energy to return to the depot. In some cases, partial recharges may offer a better trade-off between inspection speed and energy autonomy.

In this work, we propose an energy-aware task scheduling framework for multi-robot inspection missions. Our goal is to minimize the makespan while ensuring energy feasibility throughout the operation. Each robot is initialized at a central depot and must:

- Visit assigned inspection sites and perform energy-consuming measurements,
- Recharge selectively at predefined stations with variable charging rates,
- Return to the depot with sufficient battery reserves.

To address this problem, we propose:

- A Mixed-Integer Linear Programming (MILP) model integrating task allocation, energy constraints, partial recharges, and a makespan objective,
- A tailored genetic algorithm (GA) incorporating specialized operators for recharge handling and inspection compatibility,
- A comparative experimental analysis highlighting the benefits of energy-aware scheduling on mission completion time.

II. Related Work

In the field of multi-robot task allocation (MRTA) under energy constraints, several works have explored the challenge of coordinating multiple agents while managing limited energy resources. Tolmidis and Petrou [3] employ genetic algorithms to enable real-time adaptation in multi-robot coordination scenarios, highlighting the importance of dynamic scheduling. Similarly, Nait Chabane and Guenounou [4] address an inspection-oriented MRTA problem, proposing an optimization model that minimizes the total distance traveled. Their comparison between a MILP-based exact method and an improved genetic algorithm (IGA) shows that bio-inspired methods offer satisfactory trade-offs between solution quality and computation time. Comparable strategies for multi-objective optimization in human-robot collaboration and industrial scheduling have also been explored in [5], [6], reinforcing the relevance of these approaches for energy-aware resource allocation.

Bio-inspired optimization approaches, particularly genetic algorithms (GA) and ant colony optimization (ACO), have been widely applied to address MRTA and related scheduling problems. Wang et al. [7] propose a multi-objective ACO algorithm that reduces total flight time and operational cost, emphasizing the versatility of swarm intelligence techniques.

The aforementioned studies by Tolmidis and Petrou [3], further demonstrate the benefits of using GAs and IGAs with custom operators tailored for robotic coordination. Additionally, Yang and Yoo explore hybrid GA-ACO approaches to optimize drone trajectories in IoT sensor networks, showing the potential of combining metaheuristics to exploit complementary strengths.

Battery modeling and recharging strategies constitute another key research direction for energy-aware robotic systems. Di Nisio et al. and Poskart et al. [8] emphasize the importance of predictive battery discharge models for mobile robots and drones, enabling better planning of energy usage during operations. Keskin et al. [9] investigate electric vehicle routing problems (EVRPs) under uncertainty, specifically accounting for stochastic waiting times at charging stations. In a broader review, Kucukoglu et al. identify critical research gaps in EVRP formulations, highlighting the need for improved algorithmic solutions and recharge strategies. More recently, Kumar et al. [10] propose an ILP model for computing the optimal initial partial charge configuration of heterogeneous robots, with the objective of minimizing the number of required charging stations and improving overall resource allocation.

These contributions collectively lay the foundation for the energy-aware scheduling framework we develop in this paper, which integrates task assignment, recharge management, and makespan minimization into a unified optimization model.

III. Problem Description and Assumptions

A. Optimization Problem for a Collaborative Robot System

We consider a multi-robot system assigned to perform distributed inspection tasks at specific sites. Each robot starts and ends its mission at a central depot, operates at a constant speed, and is initialized with a fully charged battery. Energy consumption is modeled as proportional to the distance traveled, with an additional fixed cost incurred for each measurement performed at inspection sites. Both charging and discharging follow linear dynamics with respect to time and distance.

Robots are subject to battery constraints and can partially recharge at predefined charging stations, each with a limited charging duration and robot-specific recharge rates. Every robot must return to the depot with a minimum energy threshold to ensure safe mission completion.

The optimization objective is to minimize the makespan, defined as the time when all robots have completed their assigned tasks and returned to the depot with the required energy reserve. Our model explicitly integrates all relevant constraints: task-robot compatibility, energy-aware scheduling, partial recharging, and path feasibility. This approach contrasts with conventional full-recharge strategies by jointly optimizing task allocation, routing, and controlled partial charging to enhance overall mission efficiency under realistic operational conditions.

IV. Mathematical Model

A. Nomenclature and Decision Variables

The main indices, sets, parameters, and decision variables employed in the mathematical model are summarized in Table I for clarity and ease of reference.

TABLE I: Summary of main indices, parameters, and decision variables used in the mathematical model.

Symbol	Description
Indices and Sets	
$k \in K$	Robot index and set, $K = \{1,\ldots,n_r\}$
$i, j \in N$	Site indices and set, $N = \{0,1,\ldots,n,n_{\text{arr}}\}$
$R \subseteq N$	Set of sites with charging stations
$l \in \mathscr{M}$	Measurement index and set, $\mathscr{M} = \{1,\ldots,m\}$
Parameters	
c_{ij}	Distance between site i and site j
t_{ij}	Travel time between site i and site j
s_i	Task duration at site i ($s_0 = s_{n_{\text{arr}}} = 0$)
$E_k^{\max}$	Maximum battery capacity of robot k
B_{init}^k	Initial battery level of robot k
α_k	Energy consumption per unit distance (robot k)
β_k	Fixed energy cost per task (robot k)
ρ_k	Recharge rate at station (robot k)
γ	Minimum required energy fraction for return
M	Large constant (big-M)
C_{kl}	1 if robot k can perform measurement l, 0 otherwise
m_{il}	1 if site i requires measurement l, 0 otherwise
Decision Variables	
$x_{ij}^k \in \{0,1\}$	1 if robot k moves from site i to j; 0 otherwise
$T_i^k \geq 0$	Arrival time of robot k at site i
$B_i^k \geq 0$	Battery level of robot k upon arrival at site i
$r_{k,i} \geq 0$	Recharge time of robot k at site i (if $i \in R$; else 0)

B. Constraints

1) Task and Coverage Constraints

a) Measurement coverage

Each required measurement must be performed at least once by a capable robot:

$$\sum_{k\in K} C_{kl} \cdot \left(\sum_{\substack{j\in N \\ j\neq i}} x_{ij}^k \right) \geq m_{il}, \quad \forall i \in N, l \in \mathscr{M}. \tag{1}$$

b) constraints

A robot does not visit a site where it has no task to perform or no recharging to do:

$$\begin{aligned} \sum_{\substack{h\in N \\ h\neq i}} x_{hi}^k = 0&, \\ \forall k \in K, \forall i \in N \setminus \{R, 0, i, n_{arr}\}&, \\ i \in \{i \mid \sum_{l\in\mathscr{M}} (C_{kl} \cdot m_{il}) = 0\}&. \end{aligned} \tag{2}$$

c) Flow Conservation

For each robot, the number of incoming and outgoing arcs at any site (except the depot) must be equal:

$$\sum_{\substack{j\in N\\ j\neq i}} x_{ij}^k = \sum_{\substack{j\in N\\ j\neq i}} x_{ji}^k, \quad \forall i \in N\setminus\{0, n_{arr}\}, k \in K. \tag{3}$$

d) No Simultaneous Tasks for a Robot

A robot cannot perform multiple tasks at the same time.

$$\sum_{\substack{j\in N\\ j\neq i}} x_{ij}^k \leq 1, \quad \forall k \in K, \forall i \in N \tag{4}$$

2) Temporal and Scheduling Constraints

a) Time Propagation

If a robot moves from i to j, it must arrive at j after traveling and executing a task at i:

$$\begin{aligned} T_j^k \geq (T_i^k + t_{ij} + s_i + r_{k,i}) * x_{ij}^k,\\ \forall i, j \in N, i \neq j, k \in K. \end{aligned} \tag{5}$$

b) Initial Conditions for Time

$$T_0^k = 0, \quad \forall k \in K. \tag{6}$$

c) Calculation of $T_{\max}$

With this constraint, we determine the makespan

$$T_{\max} \geq T_i^k, \quad \forall k \in K, \forall i \in N \tag{7}$$

3) Energy and Battery Management Constraints

a) Battery Consumption and Recharge

Battery level update with recharge management:

$$\begin{aligned} B_j^k \geq B_i^k + \rho_k r_{k,i} - \alpha_k c_{ij} - \beta_k - M(1 - x_{ij}^k),\\ B_j^k \leq B_i^k + \rho_k r_{k,i} - \alpha_k c_{ij} - \beta_k + M(1 - x_{ij}^k),\\ \forall i, j \in N, i \neq j, k \in K, j \neq 0. \end{aligned} \tag{8}$$

b) Minimum Battery Level at Return

Ensure that the robot returns to the depot with sufficient remaining energy:

$$\begin{aligned} B_{n_{arr}}^k \geq \gamma E_k^{\max}\\ \forall k \in K. \end{aligned} \tag{9}$$

c) Battery Capacity Limit

$$\begin{aligned} 0 \leq B_i^k \leq E_k^{\max},\\ 0 \leq B_i^k + r_{k,i} \leq E_k^{\max},\\ \forall k \in K, i \in N. \end{aligned} \tag{10}$$

d) Recharge Only at Stations

$$r_{k,i} = 0, \quad \forall k \in K, i \notin R. \tag{11}$$

e) Initial Conditions for Batteries

$$B_0^k = B_{init}^k, \quad \forall k \in K. \tag{12}$$

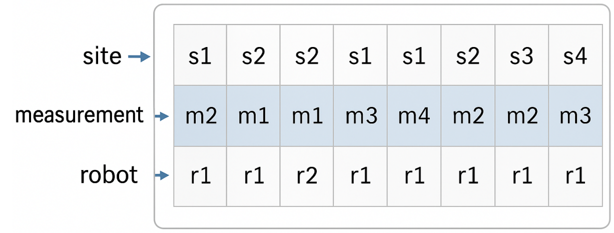

Fig. 1: Chromosome encoding structure used in this work: each gene specifies a (site, measurement, robot) assignment for multi-robot inspection. Horizontal arrows label the semantic roles of each row.

C. Objective Function

The goal is to minimize the makespan, defined as the maximum time among all robots:

$$\min: \quad T_{\max} \tag{13}$$

V. Proposed approach

A. Genetic Algorithm for Task Allocation and Planning

Genetic algorithms (GAs) are optimization techniques inspired by the process of natural selection. They iteratively evolve a population of candidate solutions through selection, crossover, and mutation to find an optimal or near-optimal solution to a given problem. In this section, we describe the genetic algorithm used to solve the task allocation and planning problem in a multi-robot system.

The Genetic algorithm was chosen here for its flexibility in solving complex combinatorial problems with numerous constraints. It is also a widely adopted approach in the literature for addressing Multi-Robot Task Allocation (MRTA) problems. The approach in this paper follows the general structure of a genetic algorithm, including initialization, evaluation, selection, crossover, and mutation. However, we introduce specific adaptations in the encoding procedure and genetic operators to take into account in particular the representation of sites and measurements, the charging and discharging of batteries or even the allocation of measurements to the robot capable.

1) Chromosome Structure

Each chromosome, as shown in Figure 1, is represented as a matrix with three rows. For each measurement of each site:

- The first row represents the site where a measurement is taken.
- The second row represents the measurement to be performed at that site.
- The third row represents the robot assigned to perform the measurement. It is chosen from among the robots capable of making this measurement.

2) Initial Population

Instead of generating a purely random initial population, we developed a heuristic to construct more efficient initial solutions. The heuristic follows these steps:

1) Create a list of tasks, where each task corresponds to a measurement required at a site.
2) Initialize the available battery levels for all robots.
3) Shuffle the list of tasks to introduce diversity.
4) For each task (site, measurement):
    a) Identify the robots capable to perform the task.
    b) If the site has a charging station:
        i) Assign the robot with the lowest battery.
        ii) Allow it to recharge.
    c) Otherwise:
        i) Assign the robot with the highest battery.
    d) Update the robot's battery level based on the travel and task execution costs.
    e) If recharging was allowed, reset the robot's battery to its maximum capacity.
5) Repeat the process for a predefined number of individuals to construct the initial population.

In practice, for a medium-sized instance, this heuristic results in approximately 10% of infeasible chromosomes (due to insufficient battery levels) compared to 60% in the purely random approach.

3) Genetic Operator

a) Crossover Operator

The crossover operator follows these steps:

- Select a crossover point along the parent chromosomes.
- The columns before the crossing point are assigned to the child.
- The remaining is completed by the data of the second parent taking only the site-measurement combinations which are not present in the first parent and preserving the order of the genes.

b) Mutation Operator

The mutation operator follows these steps:

- Select two random columns (tasks) in the chromosome.
- Swap the two selected columns while ensuring the feasibility of the new assignment.
- Then, randomly change the assigned robot, making sure to only choose from capable robots.

B. Fitness function

The fitness function evaluates the quality of a chromosome based on the total time required to complete all assigned tasks, considering feasibility constraints related to energy consumption and battery levels.

For each robot, the function simulates the execution of its assigned sequence of tasks. The procedure follows these steps:

Each robot starts at the depot with an initial battery level. For each assigned task:

- Compute travel time and energy from the previous site to the current site.
- Update the robot's arrival time and battery level.
- If the site has a charging station, calculate the charging time required to reach the next charging site without exceeding the battery capacity.

After completing all tasks, the robot must return to the depot:

- Compute the return travel and task execution costs.
- Update the arrival time and battery level accordingly.

The fitness value is the maximal arrival time among all robot. If the battery level becomes negative or the remaining battery is below a required threshold during the steps, the solution is penalized

VI. Simulation Setup

A. Scenario Details

The experiments are based on a scenario in which three robots have to perform tasks at different sites while optimizing their operating time under energy constraints. The datasets used in the simulations are structured as follows:

The robots move at constant speed and consume energy linearly as a function of the distance travelled (parameter α) and the task execution (parameter β). The energy recharge at the charging stations also follows a linear model.

B. Dataset Description

The experiments are based on synthetic benchmark instances reflecting the complexity of energy-aware multi-robot inspection. Each instance defines a set of sites, with spatial coordinates and required measurements (Table II), and a team of heterogeneous robots with varying battery capacities and capabilities (Table III). Multiple scenarios, detailed in Table IV, vary in the number of robots, sites, tasks, and charging stations, enabling an extensive evaluation of scalability and performance.

TABLE II: List of sites and corresponding measurements.

Site	X	Y	Measurements Required
0	22	29	depot
1	23	28	m1, m2, m3, m4
2	4	26	m1, m2
3	43	29	m3, m4
4	16	27	m2, m3
5	37	27	m1, m2, m3, m4
6	5	23	m1, m4
7	41	24	m1
8	2	19	m2, m3
9	32	20	m2, m3, m4
10	8	19	m1, m4
11	38	19	m2
12	10	15	m1, m2, m3, m4
13	43	18	m4
14	34	15	m2, m3
15	3	12	m3
16	13	11	m1, m2
17	33	13	m3, m4
18	42	10	m1, m3, m4
19	9	8	m1, m2, m4
20	6	5	m1, m2, m3, m4
21	13	4	m2, m3, m4
22	29	7	m1, m2
23	36	4	m3, m4
24	25	3	m2, m4
25	18	2	m2, m4
26	40	25	m2, m3
27	45	30	m1, m4
28	12	8	m1, m2, m3
29	27	14	m2, m4
30	33	22	m1, m3

TABLE III: Capabilities of the robots used for inspection.

Robot	Battery Capacity	Compatible Measurements
1	100	m1, m3
2	80	m2, m4
3	60	m1, m2

TABLE IV: Summary of the test instances.

Instance	Robots	(Sites, Tasks)	Charging Sites
1	3	(5,14)	3, 5
2	3	(8,19)	4, 8
3	3	(10,24)	3, 6, 10
4	3	(15,34)	4, 8, 12, 15
5	3	(20,48)	4, 8, 12, 16, 20
6	3	(25,58)	4, 8, 12, 16, 20, 25
7	3	(30,69)	4, 8, 12, 16, 20, 24, 30

VII. Results and Discussion

A. Comparison of MILP and GA

This section presents the results of the optimization process, comparing the performance of Mixed Integer Linear Programming (MILP) and the Genetic Algorithm (GA) based on computation time, objective value, and energy usage.

The implementation was carried out using Python, and the MILP model was solved using Gurobi Optimizer. All experiments were performed on a computer with an IntelCore i5 @ 1.60GHz, 8GB RAM, running Windows 10.

For MILP, the solver maintains a lower bound (LB) and an upper bound (UB) throughout the branch-and-bound process. When the time limit is reached, it reports the *MIP gap*:

$$\text{MIP Gap\%} = 100 \times \frac{|\text{UB} - \text{LB}|}{|\text{UB}| + \varepsilon} \tag{14}$$

where $\varepsilon = 10^{-10}$ is the default tolerance used by Gurobi to avoid division by zero when UB is close to zero. A gap of 0% indicates proven optimality, while a non-zero value (e.g., 3%) reflects the possible deviation from the optimal solution.

This metric enables a fairer comparison with GA, particularly under time-limited scenarios where exact methods may not converge.

Tables V and VI summarize the optimization results for the considered problem instances. Table V presents the performance of the MILP approach, while Table VI corresponds to the Genetic Algorithm (GA). Each table reports the computational time, the resulting optimal makespan, and the total recharge time for all robots.

TABLE V: Performance of MILP across problem instances.

Instance	Time (s)	Makespan (Mip Gap %)	Recharge Time (s)
1	0.67	62.75 (0.0%)	0.00
2	1188	96.11 (0.0%)	0.78
3	3600	102.45 (7.24%)	2.47
4	3600	123.69 (70.9%)	10.12
5	3600	189.64 (82.88%)	58.32
6	3600	245.80 (84.51%)	100.29
7	3600	283.78 (86.58%)	122.45

TABLE VI: Performance of GA across problem instances.

Instance	Time (s)	Best Makespan	Worst Makespan	Mean ± Std Makespan	Recharge Time (s)
1	3.53	62.75	62.75	62.75 ± 0.00	0.00
2	9.01	96.11	100.92	96.61 ± 1.07	0.79
3	16.90	102.54	108.34	104.61 ± 1.41	2.44
4	76.05	120.50	132.20	124.93 ± 2.29	8.01
5	148.05	191.31	218.86	201.80 ± 7.27	56.80
6	268.31	234.08	297.76	257.87 ± 19.45	90.03
7	797.32	261.14	340.98	292.06 ± 18.78	115.44

Computation Time: MILP provides optimal or near-optimal solutions for small instances in a very short time (e.g., 0.67s for Instance 1), but quickly becomes impractical with increasing problem size, reaching the 1-hour time limit from Instance 3 onward. GA, on the other hand, maintains reasonable computation times across all instances, remaining scalable even for the largest problems (e.g., 797.32s for Instance 7).

Makespan Quality: Interestingly, GA produces better makespan results than MILP in all but one instance (Instance 5). This can be explained by the high optimality gaps observed with MILP on large problems. With better tuning of genetic operators (e.g., crossover) and population size or number of generations, GA performance could be improved even further.

B. Scalability and Recharge Behavior

Recharge Time: On average, GA requires less recharge time than MILP, which is consistent with its generally better makespan. More importantly, the total recharge time increases with the number of tasks to be executed, significantly influencing the overall makespan. This trend highlights the need for partial charging strategies when early measurements or task completions are required.

Trade-Off: MILP ensures optimality on small instances but does not scale well, while GA offers better performance on large problems with reasonable time and recharge costs. Given the nature of the results, hybrid strategies are not considered here, as GA already proves efficient when appropriately tuned.

C. Trajectory and Battery Analysis

For the seventh instance with 30 sites and 69 tasks, two additional graphical analyses are presented to better illustrate the behaviour of the optimization methods (results from GA):

- Robot trajectories: A visual representation of the paths followed by each robot to complete their assigned tasks efficiently.
- Charge and discharge cycles: A time series plot showing the variations in battery for each robot.

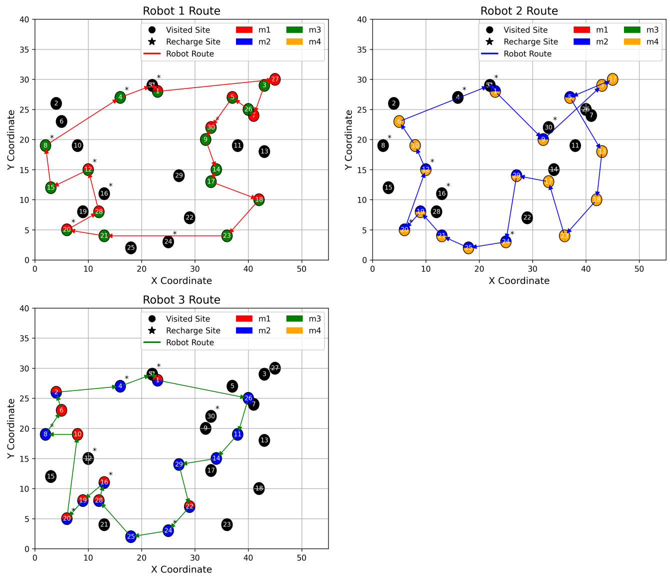

Fig. 2: Robot tours for instance 7 (30,69).

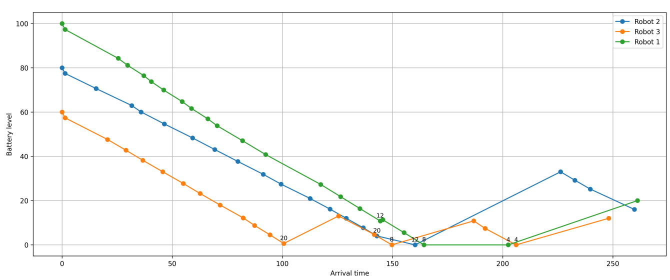

Fig. 3: Battery level evolution for instance 7 (30,69).

We observe a relatively balanced distribution of tasks, with spatial and energy constraints seemingly taken into account: the charging sites, marked with stars, are well integrated into the routes, suggesting proactive planning of charging actions. Figure 3 presents the evolution of battery levels over time. It highlights a generally linear energy consumption, interspersed with charging phases allowing the robots to continue their mission.

VIII. Conclusion and Future Work

In this study, we proposed an optimization framework for task allocation and scheduling in a multi-robot system operating under energy constraints. The results demonstrate the effectiveness of the proposed approach in balancing workload distribution and energy consumption while ensuring the timely completion of tasks within operational constraints.

The comparative analysis between MILP and the genetic algorithm highlights the trade-offs between computational efficiency and solution quality. The findings emphasize the importance of optimizing charging station utilization to minimize downtime and enhance overall mission efficiency.

Future research should explore several key aspects to further refine the proposed approach. Investigating the impact of parameters such as robot speed, charging rates, and discharging dynamics could provide deeper insights into system optimization. Additionally, optimizing the number and placement of charging stations would enhance energy efficiency and mission feasibility. Finally, incorporating a non-linear battery model for charging and discharging processes would allow for a more accurate representation of energy consumption, improving the overall performance and reliability of multi-robot systems.

Acknowledgment

This work was supported by the Pays de la Loire Region, France, under the Tandem PhD fellowship program (Grant No. 2024_07125).

References

[1] B. Ranjbar-Sahraei, K. Tuyls, and I. Caliskanelli, "Bio-inspired multi-robot systems," *Elsevier Robotics and Autonomous Systems*, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780081002490000136

[2] J. Almeida, R. Nakashima, and F. Neves-Jr, "Bio-inspired on-line path planner for cooperative exploration of unknown environment by a multi-robot system," *Robotics and Autonomous Systems*, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889018303567

[3] A. T. Tolmidis and L. Petrou, "Multi-objective optimization for dynamic task allocation in a multi-robot system," *Engineering Applications of Artificial Intelligence*, vol. 26, pp. 1458–1468, 5 2013.

[4] A. Nait Chabane and O. Guenounou, "Task Allocation and Planning for Multi-Robot System Using an Improved Genetic Algorithm," in *IEEE*, Tlemcen, Algeria, Nov. 2024. [Online]. Available: https://hal.science/hal-04774814

[5] P. Hémono, A. Nait Chabane, and M. Sahnoun, "Multi objective optimization of human-robot collaboration: A case study in aerospace assembly line," *Computers & Operations Research*, 2025, https://doi.org/10.1016/j.cor.2024.106874.

[6] P. Hémono, M. Sahnoun, and A. Nait Chabane, "Optimizing resource allocation in the flexible job shop problem: Assessing the impact of rest breaks on task strenuousness reduction," in *2023 International Conference on Decision Aid Sciences and Applications (DASA)*, 2023, pp. 320–325.

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

[8] B. Poskart, G. Iskierka, K. Krot, R. Burduk, P. Gwizdal, and A. Gola, "Multi-parameter predictive model of mobile robot's battery discharge for intelligent mission planning in multi-robot systems," *Sensors 2022, Vol. 22, Page 9861*, vol. 22, p. 9861, 12 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/24/9861/htmhttps://www.mdpi.com/1424-8220/22/24/9861

[9] M. Keskin, B. Çatay, and G. Laporte, "A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations," *Computers Operations Research*, vol. 125, p. 105060, 1 2021.

[10] N. Kumar, J. J. Lee, S. Rathinam, S. Darbha, P. B. Sujit, and R. Raman, "The persistent robot charging problem for long-duration autonomy," 2024.