EPJ Web of Conferences **354**, 03003 (2026) https://doi.org/10.1051/epjconf/202635403003
GCMM 2025

Study on Energy Aware Task Allocation and Optimization Strategies in Swarm Robotic Systems

Mallikarjun A Katageri[1], and *Senthil* Kumar Subramaniam[1*]
[1]School of Mechanical Engineering, Vellore Institute of Technology, Vellore, India

Abstract. This work focuses on implementing an energy-aware task allocation and navigation system for a swarm of e-puck robots in a simulated environment using Webots. Each robot monitors its battery level to make the required decisions like selecting tasks in one of multiple aisles, performing assigned operations, waiting at designated points, and returning to a charging station when energy falls below a critical threshold. The system optimizes energy consumption across the swarm while ensuring task completion, reducing idle time, and preventing collisions. This approach demonstrates how autonomous robots can efficiently manage resources in a collaborative multi-robot system, providing insights into real-world applications of energy-aware swarm intelligence in logistics, warehouse management, and industrial automation

1 Introduction

Swarm robotics is a domain in robotics inspired by the collective behavior of social organisms such as ants, bees, and birds [1-3]. Researchers focus on co-ordinating multiple robots to accomplish complex tasks through local communication and decentralized control with distributed energy management task allocation strategy [4-6]. The present work focuses on developing an energy-aware task allocation [7] within the Webots simulation environment [8, 9]. The robots are programmed using Python controllers to perform task allocation based on their current battery levels. Each robot dynamically selects an aisle to perform tasks, proceeds to a target location, and returns to a charging station when its battery falls below a threshold. By integrating decision making based on energy levels [10], the proposed approach emphasizes adaptive task selection and charging behavior to balance energy utilization across the swarm.

1.1 Overview of the work

- A Webots-based multi-robot setup with three aisles, a shared final block, and a charging station.

---

* Corresponding author: ssenthilkumar@vit.ac.in

- A decentralized, battery-aware task selection policy with a configurable SoC threshold.
- A Python controller implementation with collision avoidance using e-Puck [11] robots with IR sensors.

2 Experimental Setup

2.1 Robots and Sensors

The standard Webots e-Puck robot model is used, which has wheel encoders, eight IR distance sensors for obstacle proximity, GPS for global position, and a compass for heading is used. The Battery SoC is modeled as a scalar that decreases with wheel actuation and idle overhead.

2.2 Environment

The arena contains three aisles (A, B, C), a final block (FB) where tasks are acknowledged, and a charging station (CS). Waypoints are defined by (x, y) coordinates accessible via GPS.

2.3 Software Stack

Control algorithms were written in Python and run per robot. A simple message bus (shared variables or Webots supervisor-free broadcast) is used only for optional status logging; no centralized allocator is required. Figure 1 shows the simulation environment for the swarm robots.

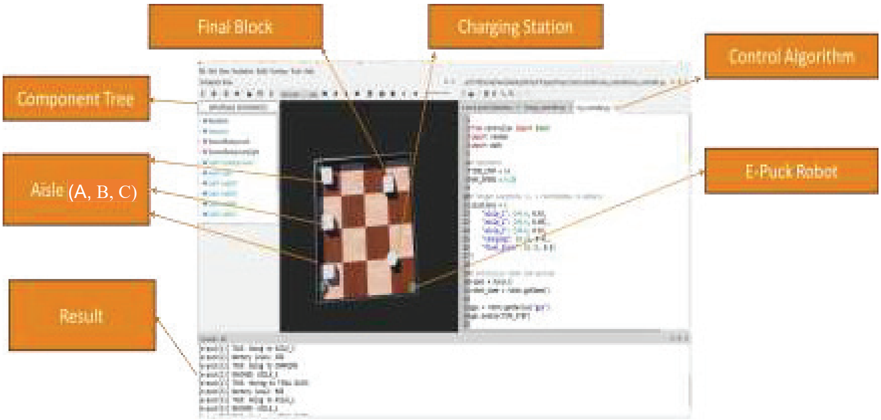


Fig. 1. Simulation Setup.

3 Methodology

The proposed system employs Webots-based e-puck robots equipped with differential drives, GPS, compass, and simulated battery sensors to ensure realistic swarm behavior. The workspace consists of three aisles (A, B, C) containing components of different weights, a final task block, and a charging station.

3.1 Initialization

At startup, each robot is assigned a unique ID, a randomized initial battery level, and a starting position near the central zone. Battery level acts as the key parameter guiding subsequent decision-making.

3.2 Battery-Aware Task Allocation

Robots select task aisles based on their current battery percentage as listed in Table 1.

Table 1. Battery-Based Task Allocation.

Battery Level (%)	Task Allocation
≥ 70	Aisle-A (High weight components)
50–69	Aisle-B (Medium weight components)
30–49	Aisle-C (Low weight components)
≤ 30	Charging Station

3.3 Navigation

After selecting a task aisle, the robot performs navigation and path planning to reach the assigned location [12]. Position and orientation data from GPS and compass modules are used to calculate motion commands toward the aisle. The navigation is implemented using predefined waypoints that guide the robot along a safe and accurate path. A simple proportional controller adjusts the robot's wheel speeds based on heading error, helping it to maintain correct trajectory while minimizing deviations. This method supports reliable navigation without requiring complex path planning algorithms.

3.4 Task Execution at Final Block

Upon reaching the final block at the end of the aisle, the robot transitions to task execution mode. In this phase, the robot simulates performing a task such as object inspection, scanning, or item pickup by remaining stationary for a fixed duration of five seconds. The waiting period allows the time required to complete assigned real-world tasks. After the simulated operation, the robot prepares to return to the central region and reassess its battery level.

3.5 Battery Re-Evaluation and Reallocation

Following task completion, the robot undergoes battery re-evaluation to determine the next course of action. If the remaining battery percentage is above the defined minimum threshold of 30%, the robot returns to the central area and selects a new aisle for subsequent task cycles based on its updated energy level. However, if the battery falls below the threshold, the robot interrupts its normal task loop and switches to charging mode to prevent total energy depletion. This repeated evaluation ensures that robots can continue functioning without unexpected shutdowns.

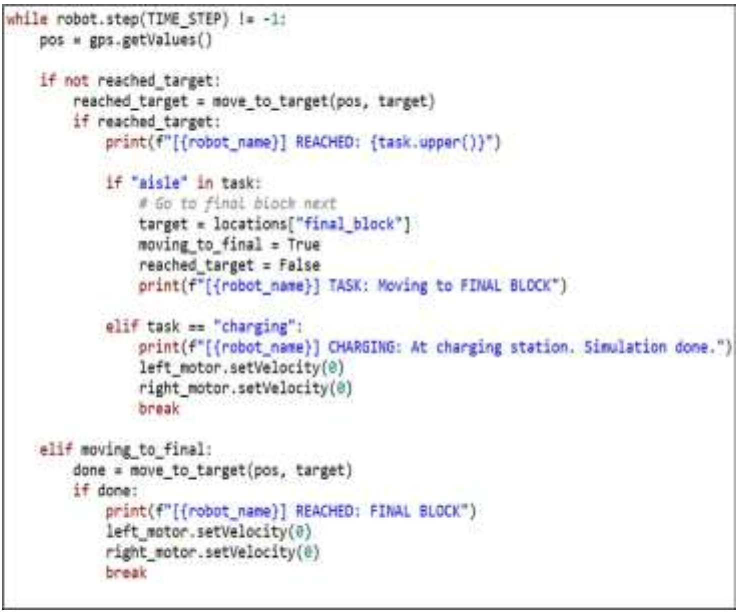
3.6 Charging Behavior

When a robot's energy level falls below 30%, it enters autonomous charging mode. In this mode, the robot navigates to the charging station using waypoint-based navigation like that used for task aisle movement. Upon arrival, the robot docks at the charging location and begins replenishing its battery. The charging may be modeled as instantaneous or gradual, but in both cases, the robot resumes normal operation once the battery level reaches an acceptable threshold such as 95% –100%.

3.7 Decentralized Control

A key aspect of the proposed methodology is its decentralized control logic [13,14]. Each robot independently monitors its own battery level, chooses its task, navigates the workspace, and decides when to recharge without requiring coordination from a centralized controller. This decentralized approach increases system robustness as overall function is unaffected even if one or more robots fail or shut down. The uniform algorithm deployed across robots enables simple swarm scalability and makes the system more flexible for real-world applications.

4 Main Control Loop and State Transition



```
while robot.step(TIME_STEP) != -1:
    pos = gps.getValues()

    if not reached_target:
        reached_target = move_to_target(pos, target)
        if reached_target:
            print(f"[{robot_name}] REACHED: {task.upper()}")

            if "aisle" in task:
                # Go to final block next
                target = locations["final_block"]
                moving_to_final = True
                reached_target = False
                print(f"[{robot_name}] TASK: Moving to FINAL BLOCK")

            elif task == "charging":
                print(f"[{robot_name}] CHARGING: At charging station. Simulation done.")
                left_motor.setVelocity(0)
                right_motor.setVelocity(0)
                break

    elif moving_to_final:
        done = move_to_target(pos, target)
        if done:
            print(f"[{robot_name}] REACHED: FINAL BLOCK")
            left_motor.setVelocity(0)
            right_motor.setVelocity(0)
            break
```

Fig 2. Main Control Code.

In main control loop and state transitions, the controller executes a discrete-time loop (driven by the simulator TIME STEP) that reads the robot GPS and progresses a simple state machine. The move to target routine returns when a target coordinate is reached while navigating it applies a baseline forward velocity and reduces one wheel's speed to steer toward the heading error. After reaching an assigned aisle the controller switches to a final block target; if the battery is below threshold the controller instead navigates to a charging location. On arrival at either a charging or final block the controller sets motor velocities to zero and terminates. The minimal state-based controller isolates high-level task-choice and energy-aware behavior from low-level path planning or obstacle avoidance.

5 Results and Discussion

The proposed energy-aware swarm control strategy was evaluated in the Webots simulation environment using four E-puck robots operating simultaneously under different initial battery conditions. The results demonstrate that the battery-driven decision-making model significantly influences energy consumption, and overall task throughput. The observations show that the robots consistently followed the intended decision hierarchy, thereby validating the correctness of the decentralized energy logic. Robots initialized with high battery levels (greater than or equal to 70%) were able to select and traverse to the aisle containing more weighted products without interruption. These robots reached the task regions, executed the required operations, and successfully moved to the final block, maintaining continuous workflow. Their behavior demonstrates that the controller efficiently exploits available energy, maximizing task coverage before battery levels fall below the threshold. Robots with medium battery levels (between 35% and 70%) demonstrated distinctly optimized behavior. Instead of attempting to obtain more weighted products, which might cause premature energy depletion, they reliably selected the aisle with medium weighted products. This resulted in reduced traversal distances, lower motion power consumption, and stable completion of the assigned tasks [15]. This behavior effectively balances workload distribution across the swarm, ensuring that medium-energy robots remain productive rather than frequently traveling to the charging station.

Robots initialized with low battery levels (less than 30%) immediately navigated to the charging station. This prevented any risk of robot shutdown mid-task, which would otherwise lead to task abandonment and potential obstruction of other robots' paths. The charging-first behavior not only maintains operational continuity but also reduces collision probability since low-energy robots do not unnecessarily occupy aisles or intersections. Moreover, the average downtime per robot decreased because only critically low-energy robots approached the charging station. Robots with moderate energy continued tasks without unnecessary charging, preventing charger congestion, which is a common issue in multi-robot systems. The results highlight that energy-awareness does not simply prolong robot operation but also improves the collective efficiency of the swarm. By intelligently distributing tasks and allocating paths based on available energy, the system achieves better coverage, higher throughput, and improved robustness. This makes the approach promising for real-world warehouses or inspection environments where robots must self-manage energy while performing sequential tasks.

6 Conclusion and Future Work

The presented work effectively demonstrated an energy-aware task allocation strategy for swarm robots using Webots and E-puck platforms. By assigning tasks based on battery level and routing low-energy robots to a charging station, the system ensured continuous work execution without unexpected robot shutdowns. The decentralized nature of the algorithm allowed each robot to independently evaluate its energy state, select appropriate tasks, navigate to target locations, complete the assigned work, and return for recharging when required. The implementation exhibited that simple rule-based logic can be used to effectively balance task distance and available battery power, reducing unnecessary travel and helping robots complete more tasks per operational cycle. The waypoint-based navigation combined with proportional steering was sufficient for reliable movement in structured environments. Results confirm that the proposed method increases overall system robustness, minimizes idle time, and provides a scalable approach to multi-robot task distribution.

References

1. E. Sahin, Swarm robotics: From sources of inspiration to domains of application. In Int. workshop on swarm robotics. Berlin, Heidelberg: Springer Berlin Heidelberg.10-20 (2004).
2. P.G.F. Dias, M.C. Silva, G.P. Rocha Filho, P.A. Vargas, L.P. Cota, G. Pessin, Swarm robotics: A perspective on the latest reviewed concepts and applications. Sensors, 21(6) 2062 (2021).
3. S. Rooban, M. Javaraiu, P. P. Sagar, P.P., 2022, April. A detailed review of swarm robotics and its significance. In: International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) IEEE. 797-802 (2022).
4. A. Djenadi, B. Mendil, Energy-aware task allocation strategy for multi robot system. Int. J. Mod. & Sim. 42(1) 153-167 (2022).
5. J.N. Yasin, H. Mahboob, M.H. Haghbayan, M.M. Yasin, J. Plosila, Energy-efficient navigation of an autonomous swarm with adaptive consciousness. Remote Sensing, 13(6), 1059 (2021).
6. V. Dabass, S. Sangwan, Strategic allocation: exploring optimization techniques in multi-robot systems. Int. J. Intelligent Robotics and Appn. 1-23 (2025).
7. E. Latif, Y. Gui, A. Munir, and R. Parasuraman, Energy-aware multi-robot task allocation in persistent tasks. arXiv preprint arXiv:2112.15282 (2021).
8. L. Turkler, T. Akkan, and L. O. Akkan, Control of Swarm Robotics in Webots with PSO. In 2021 3rd Int. Cong. on Human-Computer Interaction, Optimization and Robotic Applications (HORA) IEEE 1-6 (2021).
9. O. Michel, Cyberbotics Ltd. Webots: Professional Mobile Robot Simulation, Int. J. Adv. Robotic Syst. 1(1), 5(2004).
10. G. Miyauchi, M.S. Talamali, R. Grob. A comparative study of energy replenishment strategies for robot swarms. In International Conference on Swarm Intelligence, Cham: Springer Nature Switzerland. (15) 3-15 (2024).
11. F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.C. Zufferey, D. Floreano, A. Martinoli, The e-puck, a robot designed for education in engineering. In Proceedings of the 9th conference on autonomous robot systems and competitions 1 (1) 59-65 (2009).
12. A. Ayari, S. Bouamama, Evolutionary Swarm Robotics: A Methodological Approach for Task and Path Planning. In IEEE Afro-Mediterranean Conference on Artificial Intelligence (AMCAI) IEEE. 1-6 (2023).
13. R. B. Amor, S. Elloumi, 2017, December. Decentralized model reference adaptive control for interconnected robotic systems. In 2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA),

IEEE, 235-240 (2017).

14. T. Abukhalil, M. Patil, T. Sobh, A comprehensive survey on decentralized modular swarm robotic systems and deployment environments. Int. J. Engg. 7(2) 44 (2013).
15. S. Singh, R. Vaishnav, P. Goswami, Evaluating Area Coverage Efficiency in Swarm Robotics: A Comparative Study of Different Approaches. In 2025 International Conference on Computer, Electrical & Communication Engineering (ICCECE) IEEE. 1-7 (2025).