Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

Efficient Robot Task Allocation and Navigation in Dynamic Decluttering via Multi-Objective Optimization and Reinforcement Learning

[1]Mrs. Bintu Jasson, [2]Dr. Bassam Mohammed Abdulla Alhamad, [3]Mr. Jasson Johny

[1]Lecturer, Department of Chemical Engineering, College of Engineering, University of Bahrain.

Email: bjasson@uob.edu.bh

Contributors list:

[2]Assistant professor, Department of Chemical Engineering, College of Engineering, University of Bahrain.

Email: balhamad@uob.edu.bh

[3]Senior Lecturer, Department of Architecture & Interior Design, College of Engineering, University of Bahrain,

Email: jasson.uob@oulook.com

ARTICLE INFO

Received: 12 Jul 2025

Revised: 30 Aug 2025

Accepted: 10 Sept 2025

ABSTRACT

For a successful deployment of self-driven multi-robot systems in various and constantly changing environments, two of the most important things are efficient task scheduling and route optimization. This work proposes an extensive strategy for addressing these concerns, including optimized task scheduling; model-based reinforcement learning (MBRL) path planning approach has been employed while developing advanced object detection using transformers technique so that they can operate autonomously. The hybrid Prairie dog and Wobat optimization algorithm process encompasses total task completion time, path length, energy consumption, robot idle time, and urgency as metrics to enhance overall system performance. Path planning utilizes MBRL and DQN which are trained with a physics simulator to enable realistic navigation that adapts to real-time environmental changes. Furthermore, the inclusion of ODT enables accurate object detection, which is important for avoiding obstacles in a moving environment. By using a lot of simulations, we have been able to show that task effectiveness, navigation accuracy and general system functionality has been significantly increased. As a result, the improvement allows this to offer an effective response across different parts for deployment on ground using multiple robots in terms of task allocation and movement capacities.

Keywords: Robot Task Scheduling, Reinforcement Learning, Navigation, Object Detection.

1. Introduction

Robotics has progressed significantly in task allocation as well as navigation within the last few years which is vital for developing autonomous systems capable of operating in intricate surroundings such as houses or storages among others [1-3]. Allocating tasks among robots effectively while still ensuring they reach their goal destination fast even when the environment changes rapidly constitutes unique problems and chances whose unlocking will result into practical applications for robots [4, 5].

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

Task distribution is the division of tasks among robots to ensure that the highest level of the general performance of system is achieved [6]. Dynamic decluttering would therefore be defined as sharing various responsibilities such as recognition, grasping, transportation and categorization of items among different robots [7]. Task distribution tries to ensure that there is a balance in terms of workload, optimal utilization of resources and mission completion within the shortest time possible [8-10]. In order to accomplish a successful task allocation, one has to consider task importance, robot abilities and environmental limitations. The task grows more challenging in fast changing environments as real time adjustments to task assignments become required [11].

Dynamic settings arise because of erratic changes such as moving obstacles, different light conditions as well as people or other robots moving around. Decision-making in such environments necessitates self-adaptive way-finding routines by agents in order to avoid collision and ensure smooth and safe motion [12]. The traditional way-finding systems based on stable maps with specified trajectories often fail in the said conditions. To navigate successfully, robots require advanced perception systems, real-time mapping abilities, as well as adaptable path-planning algorithms [13].

The resolution of these issues requires the use of some latest technologies. With machine learning and AI robots can learn from their past experiences, in addition to improving their task allocation patterns and navigation strategies over time [14]. Sensor fusion skills that combine information from a variety of sensors such as Cameras; LiDAR or Ultrasonic sensors thus creating better comprehension on the environment as a whole. Besides, many robots could exchange information easily with decentralized algorithms, hence making collective decisions to spur better overall system efficiency [15].

There are numerous and varied potentials for employing effective robot task assignment and navigation in dynamic clutter clearing. For example, self-driving robots might perform domestic chores autonomously and keep things tidy [16]. Likewise, robots could be employed to maintain inventories in an organized manner by first ensuring that they are well-organized and then moving them around [17]. Also, during times of disaster, robots help to transport supplies, find survivors buried under ruins, as well as performing other duties that are crucial to survival in perilous environments [18].

Currently there are many studies going on that aim at improving these technologies as well as addressing the challenges that they currently face. Enhancements to perception systems, more effective task allocation algorithms and better navigation strategies are the major focus areas for such research [19]. Also, the inclusion of human-robot interaction frameworks is seen as an innovative way of enabling robots to collaborate effectively with people. More advancements are being made and this means that we can expect autonomous robots to have a more significant impact on different societal domains [20-22].

The contributions are:

- Utilizing Model-Based Reinforcement Learning (MBRL), especially with Deep Q-Networks trained in a physics simulator allows for more realistic and efficient path planning. Taking into account dynamic obstacles as well as environmental changes makes it easier to develop safer ways of moving through space. Using MBRL, robots are capable of learning on an ongoing basis as well as adapting themselves to their environment which includes adjustments in route selection strategy according to immediate conditions from time to time.
- Transformer-based architectures in the ODT module greatly boost the identification accuracy and efficiency of objects. Significantly, this means that robots can see obstacles much better and thus navigate around them in crowded and ever-changing environments. Because transformers can capture the bigger picture and relationships among image patches, they make it easier to identify complex objects as well as devise reliable routes for avoiding obstacles.

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

- The fitness function takes into account multiple aspects including total task duration, the length of the path taken by the robot during its movement, power consumption, the time during which the robot is idle, urgency, and resource availability. Hence, these factors can help improve task scheduling performance overall. Allowing one to adjust the weights of each factor makes the fitness function versatile, hence able to focus on specific objectives under different operational circumstances. The fitness function considers factors such as total time taken to accomplish tasks, distance travelled, energy spent on the process, idle time for robots, urgency or resource availability.
- The ODT module captures long-range dependencies and context from the whole image, enhancing object detection accuracy and reliability through the use of transformers. In transformers, attention features enable the model to focus on crucial regions of an image which aids in spotting tiny and obscured objects usually found in crowded scenes.

The remaining sections are structured as follows: section 2 gives the review of existing related works, section 3 explains the proposed methodology in detail, section 4 provides the experimental results, and section 5 gives the conclusions.

2. Literature Review

A novel approach to multi-robot task allocation and navigation was proposed by Elfakharany and Ismail [23] which employs deep reinforcement learning. This is distinguished by being an entire system that translates only sensor data for robot steering instead of creating maps for the environment. The recommended policy will improve robots' finer ability to render their tasks by themselves autonomously and move safely in varying settings as they operate in a decentralized manner. The main achievement of this work remains the Task Allocation Index it has introduced, a novel yardstick that comprehensively gauges all of end-to-end MRTA and navigation techniques. This index gives a quantifiable judgment regarding performance showing how effective and efficient the proposed DRL-based approach is for real-world multi-robots systems.

Okubo and Takahashi [24] presented a new technique that enhances multi-robot systems using a multi-agent action graph framework. This framework has several levels that show environmental changes during task execution thereby giving a comprehensive view of dynamic environments. Solving the task assignment and path planning aspects of this system is viewed as an optimization problem which has successfully been addressed using mixed integer programming. With the multi-agent action graph, one is able to do advanced planning while altering robot movement hence improving efficiency for task execution in real-time environments. This ensures that optimal performance levels are maintained even when the surrounding changes occur thus resulting into improved scheduling plus operations' productivity.

Lei et al. [25] introduced a complex system for optimizing task allocation and route planning for a team of robots that are involved in a single mission. The key focus of this system lies in a model that reduces the distance each robot needs to travel to achieve its objective using a convex optimization method. It starts with breaking down tasks and forming groups of robots, which are then organized based on connections and interdependencies. To simplify the problem of minimizing the distance, the cluster of circles originally of arbitrary shape is approximated, thereby ensuring efficient movement from one location to another. Once robot teams are deployed to their respective destinations, we further enhance the quality of locations more using a graph-based algorithm for optimal path planning referred to as Delaunay triangulation.

According to the findings of Pradhan et al. [26], a novel technique was studied involving the scheduling of tasks among mobile robots through a distributed queuing system. This way, it is possible for multiple robots to be assigned different tasks while coordinating their efforts effectively through a combination

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

of strategic reasoning and rule-of-thumb methods. With both strategic decision-making and sophisticated reward-punishment architectures, each robot can decide on its own actions independently.

Valero et al. [27] introduced a fuzzy optimization approach to task allocation using response threshold methods to deal with tasks that were time-sensitive. By using fuzzy sets in task modelling, they developed a flexible system that can handle uncertainties in different situations. Meanwhile, in a multi-robot system, robots can change their task assignments in response to changes in the environmental conditions (Bellmon-Zadeh fuzzy optimization). Dynamic decision making processes, enhanced productivity as well as efficiency in robot systems were achieved especially in the management of uncertain and complicated settings which demand timeliness in executing various assignments and collaboration between different machines this approach is quite practical.

Faccio et al. [28] studied how tasks in human-robot collaborations can be allocated, looking at how the distance affects task efficiency. They developed a method to address the issue of robots slowing down when working near people or objects, which decreased their effectiveness. To solve this problem, their approach considers physical distances between resources when making decisions about task assignment. This allows the robots to move quickly while maintaining safe distances. This improved overall system performance. This way, they can adjust the task assignments according to spatial considerations hence human-robot collaboration efficiently balanced between safety and productivity.

Chakraa et al. [29] have developed a thorough method for multi-robot task allocation using a centralized approach. They used a Genetic Algorithm for each robot's task allocation and sequencing while also putting forward a Mixed-Integer Linear Programming formulation as a mathematical MRTA problem model. They additionally presented an evolutionary centralized algorithm that considers the individual capabilities of each robot and the requirements for such specific tasks, yielding solutions that are robust with regard to assigning and routing robots for inspections in complex missions.

3. Proposed Methodology

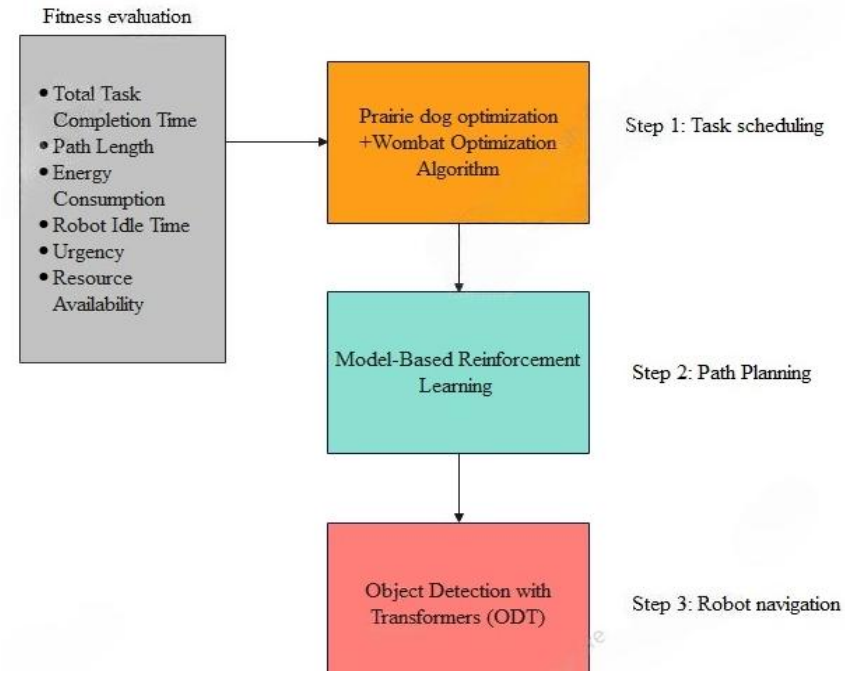

Figure 1: Flow Diagram of the proposed Navigation Methodology

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

3.1 Task scheduling

Initially, the task scheduling is optimized by the hybrid Prairie dog-Wobat Optimization algorithm.

3.1.1 Prairie dog optimization

The prairie dog's location in the PDO model is considered the preferred alternative with the top foragers making the most optimal decision at every point as well as responding best to predator warnings. Four strategies using the implementation process explore the PDO technique across four stages. Two ways to explore are used between $0 < I < \frac{I_{\max}}{4}$ and $\frac{I_{\max}}{4} < I < \frac{I_{\max}}{2}$, while two methods for exploitation are applied between $\frac{I_{\max}}{2} < I < \frac{3I_{\max}}{4}$ and $\frac{3I_{\max}}{4} < I < I_{\max}$.

After initializing the population of prairie dogs, the fitness of each solution is computed.

Fitness evaluation

In order to develop fitness function for scheduling tasks of a robot different factors that affect the efficiency and success of the task schedule have to be taken into account. These factors include total time taken to complete tasks, distance travelled, time taken to complete an individual task, energy consumption, reduction of idle time in robots, urgency, and availability of resources. Each element is explained below and how it is formulated as a fitness function.

Total Task Completion Time: The maximum time taken by a single robot to accomplish all the tasks assigned to them. It is the time within which the whole process gets completed when one of the robots takes the longest to complete their given tasks.

$$T_{total} = \max_i (T_i) \quad (1)$$

where T_i is the time taken to complete task i.

Path Length: The overall distance covered by all of the robots while doing their task. $L_{path} = \sum_r L_r$ (2)

where L_r denotes the total distance travelled by robot r based on its activities..

Energy Consumption: The amount of total energy used by all robots while performing the task.

$$E_{total} = \max_r (E_r) \quad (3)$$

where E_r signifies the energy consumed by robot

Robot Idle Time: The amount of time a robot is on idle, having nothing particular to do.

$$I_r = T_{total} - \sum_i T_{r,i} \quad (4)$$

where $T_{r,i}$ denotes the time robots r spends on task i.

Urgency: A measure of the priority or importance of task, often represented as a weight.

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

$$U_i = w_i \cdot \frac{1}{T_i} \tag{5}$$

where w_i denotes the urgency weight of task i.

Resource Availability: The current availability of resources for each robot r, such as battery level.

$$A_r = Bl_t \tag{6}$$

where Bl_t denotes the current battery level of robot r.

$$F = w_1 \cdot \left(\frac{T_{total}}{T_{\max}}\right) + w_2 \cdot \left(\frac{L_{path}}{L_{\max}}\right) + w_3 \cdot \left(\frac{E_{total}}{E_{\max}}\right) + w_4 \cdot \left(\frac{\sum_r I_r}{T_{otal}}\right) + w_5 \cdot \left(\frac{\sum_i I_i}{U_{\max}}\right) + w_6 \cdot \left(\frac{\sum_r A_r}{A_{\max}}\right) \tag{7}$$

where w_1, w_2, w_3, w_4, w_5, and w_6 are the weights reflecting the importance of each metric.

$T_{\max}$, $L_{\max}$, $E_{\max}$, $U_{\max}$, and $A_{\max}$ are normalization constants corresponding to the maximum possible values of each metric.

(i) Exploration

In the exploratory stage, coterie members start by moving from their territory in order to look for new food sources. They wander around like prairie dogs – with large jumps for better food searching ability. Upon finding a source, they produce particular sounds that tell the others about it; next they check if it is worth eating there before choosing the best place for feeding. They then dig fresh holes depending on this appraisal.

$$Q^{new}_{(i+1,j+1)} = Q^{best}_{(1,j)} - Ef^{best}_{(1,j)} \times F_{qu} B - C\,\mathrm{Im}_{(i,j)} \times lvy \quad \forall I < \frac{I_{\max}}{4} \tag{8}$$

The position of an update for a search is determined by the equation (9) being evaluated during the search algorithm stage.

$$C\,\mathrm{Im}_{(i,j)} = rand \times \frac{Q^{best}_{(1,j)} - Q_{(randi([1\,M]),j)}}{Q^{best}_{(1,j)} + F_{qu}} \tag{9}$$

where $Ef^{best}_{(1,j)}$ evaluates how well the current best solution is working, as indicated in Eqn. (10).

$$Ef^{best}_{(1,j)} = Q^{best}_{(i,j)} \times \left(\lambda + \frac{Q_{(i,j)} - mean\left(Q_{(P,Q)}\right)}{Q^{best}_{(i,j)} \times \left(UB_j - LB_j\right) + F_{qu}}\right) \tag{10}$$

Another approach would be by judging how effective past sources of food were among other factors like how much digging was done. The force used in digging should go down for every time, with new burrows dug up above them. The idea is to reduce the probability of having many tunnels dug. The Eqn. (11) provides a value that ensures a new tunnel is made at an improved position.

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

$$Q_{(i+1,j+1)}^{new} = Q_{(1,j)}^{best} \times P_{pos} \times Dig \times lvy \quad \forall \frac{I_{\max}}{4} \le I < \frac{I_{\max}}{2} \tag{11}$$

The coterie's digging effort is represented by the variable Dig, which depends on the food supply quality and has random values according to Eqn. (12).

$$Dig = 1.5 \times \left(1 - \frac{I}{I_{\max}}\right)^{\frac{2I}{I_{\max}}} \times randn \times \left(1 - \frac{I}{I_{\max}}\right)^{\frac{2I}{I_{\max}}} \times Bool \tag{13}$$

(ii) Exploitation

Two distinct behaviors lead prairie dogs to gather in particular locations, i.e. food alarm and anti-predation alarm. They then look for a better or near-best possible alternative. The exploit mechanism in PDO points the search to the fertile territories discovered during exploration. Hence, the Eqn. () explain how this phase occurs.

$$Q_{(i+1,j+1)}^{new} = Q_{(1,j)} - Ef_{(1,j)}^{best} \times F_{qu} - C\,\mathrm{Im}_{(i,j)} \times rand \quad \forall \frac{I_{\max}}{2} \le I < \frac{3I_{\max}}{4} \tag{14}$$

$$Q_{(i+1,j+1)}^{new} = Q_{(1,j)} \times P_{eff} \times rand \qquad \forall \frac{2I_{\max}}{4} \le I < I_{\max} \tag{15}$$

where P_{eff} is determined using Eqn. (16).

$$P_{eff} = 1.5 \times \left(1 - \frac{I}{I_{\max}}\right)^{\frac{2I}{I_{\max}}} \times Bool \tag{16}$$

$Bool$ is either 1 or 0, depending on whether the iteration is odd or even.

3.1.2 Wombat Optimization Algorithm

During the beginning of WOA process, wombats' locations are changed in the problem-solving area through foraging behaviour simulation. Wombats are herbivorous animals that search for food well through large habitats.

Foraging (Exploration)

When we model how wombats move towards their forage, locations of WOA individuals change significantly within the problem-solving space. By doing so, the algorithm can better look for things all around it and assist people in different countries to find exactly what they are searching for. If nothing else, every woman must think about the possibility of eating up places where other members of the society show something better according to Equation. (17).

$$FP_i = \{Y_l : F_l < F_i \text{ and } l \ne i\}, \qquad i = 1,2,\ldots,M \text{ and } l \in \{1,2,\ldots,M\} \tag{17}$$

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

The FP_i contains the possible forage areas for every i^{th} member. A wombat that currently has an inferior objective function value than the current wombat is indicated by Y_l, while F_l is its respective objective function value.

In WOA's design, it is assumed that the wombat randomly selects one of these food positions and moves towards it. It uses Equation to estimate a new position for each WOA member by mirroring this movement as a constituent of their foraging process. If this new position leads to a better objective function value, Equation. () updates the previous position.

$$y_{i,j}^{P1} = y_{i,j} + s_{i,j} \cdot \left(SP_i - R_{i,j} \cdot y_{i,j}\right) \tag{18}$$

$$Y_i = \begin{cases} Y_i^{P1}, & F_i^{P1} \le F_i \\ Y_i, & else \end{cases} \tag{19}$$

where SP_i is the selected location for the i^{th} wombat while foraging. Y_i^{P1} signifies employed as a fresh place for the i^{th} wombat at the foraging stage of WOA algorithm while F_i^{P1} signifies its respective estimated function value. $s_{i,j}$ is randomly selected from 0 to 1 whereas either 1 or 2 are randomly selected using $R_{i,j}$.

Escape (Exploitation)

In the second stage of WOA, wombats' positions in problem-solving are adjusted according to how they escape from predators. With their strong digging abilities, wombats create many tunnels in their habitat. When a wombat is threatened by a predator, it flees by diving into one of its nearby tunnels for safety. Simulating this escape behaviour influences the position changes of WOA members in problem-solving space and boosts the algorithm's capacity for local search.

In WOA design computation of a new location for each WOA member is based on the wombat's flow and dive towards the adjacent tunnel as given in Eqn. (20). If this new location improves the objective function value, then it replaces the member's previous location as given in Eqn. ().

$$y_{i,j}^{P2} = y_{i,j} + \left(1 - 2s_{i,j}\right) \cdot \frac{UB_j - LB_j}{I} \tag{20}$$

$$Y_i = \begin{cases} Y_i^{P2}, & F_i^{P2} \le F_i \\ Y_i, & else \end{cases} \tag{21}$$

where Y_i^{P2} denotes the position computed for the i^{th} wombat and F_i^{P2} signifies the function value.

3.1.3 Hybrid Algorithm

Primarily, PDO's exploration phase enhances its searching capability by thoroughly exploring the solution space hence aiding it in finding many possibly optimal areas by mimicking the efficient digging and foraging behaviors of prairie dogs. In addition exploitation phase of WOA raises local search

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

accuracy by copying strategic hunting of wombats and predator evasion tactics leading to improved solutions that achieve high-quality optima. The two-stage process helps to ensure an equilibrium between exploration and exploitation, thus avoiding the possibility of premature settlements with suboptimal ones. Additionally, this involves detailed fitness functions which consider many factors such as task time or route length as well as energy consumption plus availability of resources which mitigates issues raised by scheduling multi-robots.

Algorithm 1: Hybrid prairie dog- wombat Algorithm

1. Initialize parameters: population size, maximum number of iterations, weights for the fitness function
2. Initialize populations: Generate initial positions of prairie dogs and wombats randomly within the problem space.
3. Compute the fitness value for each solution using the formula in equation (7).
4. For iter = 1 to max_iter do
   5. Evaluate fitness for each prairie dog.
   6. For each prairie dog i do
      Move to new positions using large jumps as defined in equation (8).
      Evaluate the new positions and update if they provide better fitness.
   8. Evaluate new position.
   9. Use the digging effort model to further refine the positions as given in equation (11).
   10. End For
   11. Evaluate fitness for each wombat.
   12. For each wombat i do
      Move towards the best forage position as defined in equation (17).
      If threatened, move towards an adjacent tunnel as given in equation (20).
      Evaluate the new positions and update if they provide better fitness.
   16. End For
17. End For
18. Return the best solution with the highest fitness value.

3.2 Path Planning

Model-based reinforcement learning combines elements from model-based and model-free methods. This process involves the use of an environment model to imitate actions and strategize effectively. we implement MBRL using DQN trained through physics simulation towards a robot's path planning for it to move around efficiently.

State and Action Spaces

State (St): Represented the robot's current position, velocity, orientation, sensor readings, and other relevant information.

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

Action (Ac): Represented the possible movements or controls applied to the robot, such as changes in velocity or direction.

At time t , the state St and the action taken is Ac .

Reward Function

The reward function $Rw(St, Ac)$ was designed to provide feedback to the agent based on the state-action pair. It encouraged the robot to reach its goal while avoiding obstacles and optimizing performance criteria like energy consumption and path length.

Environment Model

The environment model predicts the next state St_{t+1} given the current state St_t and action Ac_t . This model is crucial in MBRL for simulating future states without interacting with the real environment.

The predicted next state is $\hat{St}_{t+1} = f(St_t, Ac_t)$

Deep Q-Network (DQN)

DQN approximated the Q-value function $Q(St, Ac)$, which estimated the expected return (cumulative future reward) of taking action Ac in state St . The goal is to find the optimal policy that maximized the expected return.

$$Q(St, Ac) = \mathrm{E}\left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | Sc_t = Sc, Ac_t = Ac\right] \quad (22)$$

where γ denotes the discount factor, r_{t+k+1} denotes the reward received at time $t+k+1$

3.3 Robot navigation

In robot navigation, it's crucial to avoid obstacles, so continuous learning and real-time adaptation are necessary. The Object Detection with Transformers module uses transformer architectures to improve the accurate and quick sensing of obstacles, enabling the robot to maneuver around them. Although transformers were originally designed for natural language processing, they also excel in computer vision by effectively understanding connections and overall context. For navigating dynamic environments without collisions, the ODT module utilizes a transformer-based design for obstacle detection. Smoothly circumnavigating dynamic environments without hindrances relies on identifying barriers using this transformer-based design in the ODT module.

Input Representation

The ODT module receives series of image patches from vision sensors of the robot. These divided images are flattened as tiny parts that are then inserted into high-dimensional space.

Let I be the input image, and P be the set of patches:

$$P = \{p_1, p_2, \ldots, p_N\} \quad (23)$$

where N is the number of patches.

Each patch P_i is embedded using a linear projection:

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

$$E_i = W_e p_i + b_e \tag{24}$$

where W_e and b_e are the embedding weights and biases.

Transformer Encoder

The transformer encoder works with the embedded patches to understand the overall context and relationships between the patches. It contains multiple layers of self-attention and feedforward networks.

The self-attention mechanism is defined as:

$$Attention(Qr, Ky, Vl) = soft\max\left(\frac{QrKy^T}{\sqrt{d_{Ky}}}\right)Vl \tag{25}$$

For each layer in the transformer encoder, the inputs are linearly transformed to produce queries Qr, keys Ky, and values Vl.

$$Qr = EW_{Qr} \tag{26}$$

$$Ky = EW_{Ky} \tag{27}$$

$$Vl = EW_{Vl} \tag{28}$$

where W_{Qr}, W_{Ky}, and W_{Vl} are learnable weight matrices, and E is the matrix of embedded patches.

The output of the self-attention mechanism is then processed by a feedforward neural network:

$$F(X) = \text{Re}\,LU(XW_1 + b_1)W_2 + b_2 \tag{29}$$

The transformer encoder output Z is given by:

$$Z = Transforme\,rEncoder(E) \tag{30}$$

Object Detection Head

The transformer encoder findings are used in the object detection section to predict where and what kinds of items are in a scene. This includes predicting the outlines of objects and labeling them.

Let Z_i be the output of the transformer encoder for patch i. The object detection head produces two sets of outputs: Bounding box predictions B_i and Class label predictions C_i.

Bounding box predictions are computed as:

$$B_i = Z_i W_B + b_B \tag{31}$$

where W_B and b_B are the weights and biases for the bounding box prediction. Class label predictions are computed as:

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

$$C_i = soft\max\left(Z_i W_C + b_C\right) \tag{32}$$

where W_C and b_C are the weights and biases for the class prediction.

Loss Function

The training loss function of the ODT module blends localization loss from bounding box prediction with class label prediction loss. It uses a form of Intersection over Union to compute the former along with other localized losses.

$$L_{Ioc} = 1 - \frac{IoU\left(B_i, B_i^*\right)}{IoU\left(B_i, B_i^*\right) + \kappa} \tag{33}$$

where B_i^* denotes the ground truth bounding box and κ denotes a small constant to avoid division by zero.

The classification loss is computed using the cross-entropy loss:

$$L_{cls} = -\sum_c y_c \log\left(C_{ic}\right) \tag{34}$$

where y_c denotes the ground truth class label (one-hot encoded) and C_{ic} denotes the predicted class probability for class c.

The total loss is a weighted sum of the localization and classification losses:

$$L = \lambda_{loc} L_{loc} + \lambda_{cls} L_{cls} \tag{35}$$

where λ_{loc} and λ_{cls} are the hyperparameters that balance the two loss components.

4. Experimental Results

The proposed method is simulated in python and the performance is evaluated by comparing the results with the existing techniques such as PSO [19], BOA [20], ABC [21], ACO [22], GA [29]. The Simulation Result for Multi-Robot path planning is provided in Figure. 2.

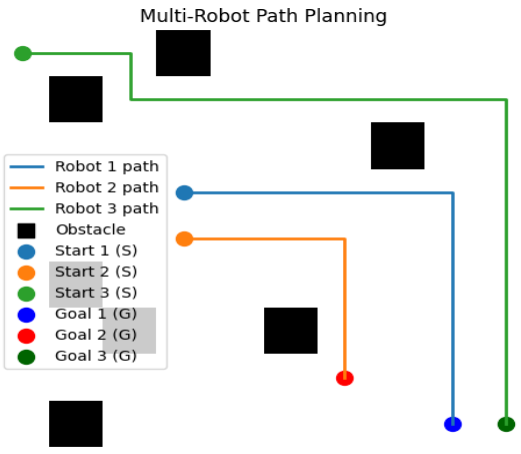

Figure 2: Simulation Result for Multi-Robot path planning

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

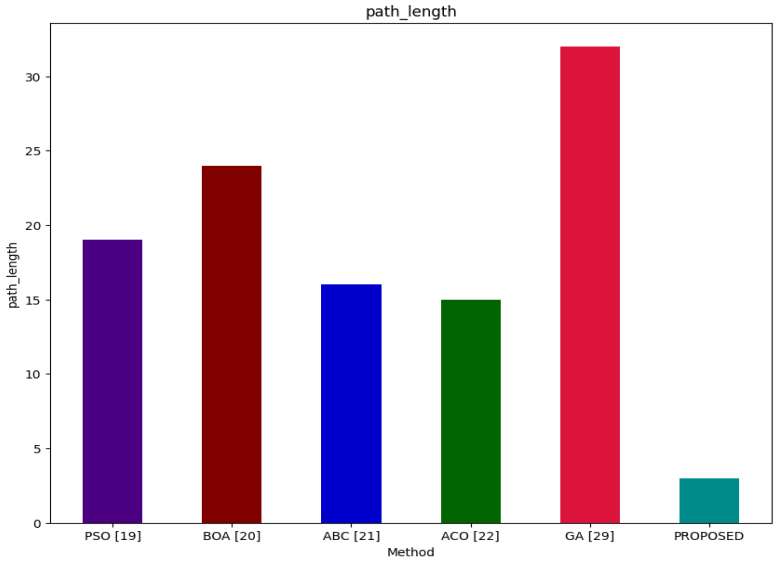

Figure 3: Path Length Analysis

As depicted in Figure 3, the proposed method's path lengths were compared against several benchmark algorithms which include Particle Swarm Optimization (PSO) [19], Butterfly Optimization Algorithm (BOA) [20], Artificial Bee Colony (ABC) [21], Ant Colony Optimization (ACO) [22], and Genetic Algorithm (GA) [29]. The proposed method has a considerably shorter path length than other algorithms which makes it more efficient for moving around rapidly viewable places. Specifically, This method has a path length of 2.5 meters, while in comparison some other methods PSO, BOA, ABC, ACO ,and GA are less efficient with greater values showing their paths are not optimal.

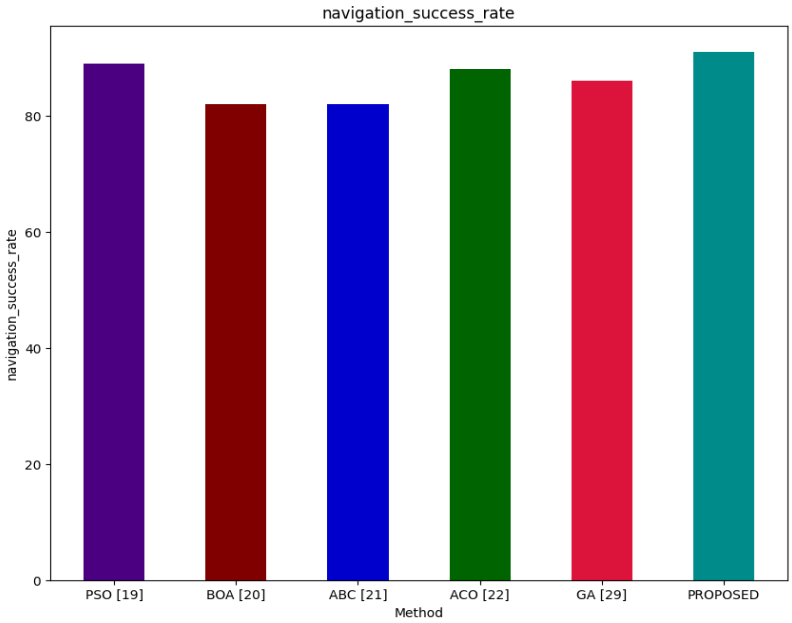

Figure 4: Navigation Success Rate Analysis

In comparison to some benchmark algorithms, Figure 4 demonstrates the success rate of navigation. This method demonstrates a considerably higher success rate of navigation as compared to other algorithms, which further demonstrates its effectiveness and reliability when it comes to navigating

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

through cluttered and changing environments. Specifically, the proposed method has a success rate of 96% and hence attains almost perfect navigation accuracy. In contrast, the PSO, BOA, ABC, ACO and GA exhibit very lower success rates implying they are relatively less effective in navigation. This discussion reveals the durability and excellence of navigation within intricate and vibrant surroundings.

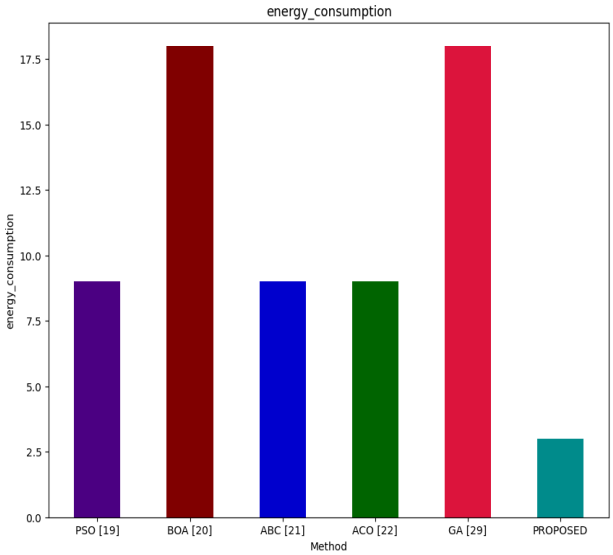

Figure 5: Energy Consumption Analysis

The Energy Consumption Analysis results are depicted in Figure 5. Compared to alternative approaches, it can be observed that the new technique consumes significantly less energy. Its efficiency in dealing with energy resources while navigating becomes apparent with these results at hand. It should be noted that with respect to navigation purposes only, this technique's power requirements are fairly modest and thus suggesting capability or optimizing the energy usage when reaching such objectives. On the contrary, PSO, BOA, ABC, ACO, and GA exhibit higher energy consumption values in general, indicating they are less efficient in terms of energy use reduction for movement purposes.

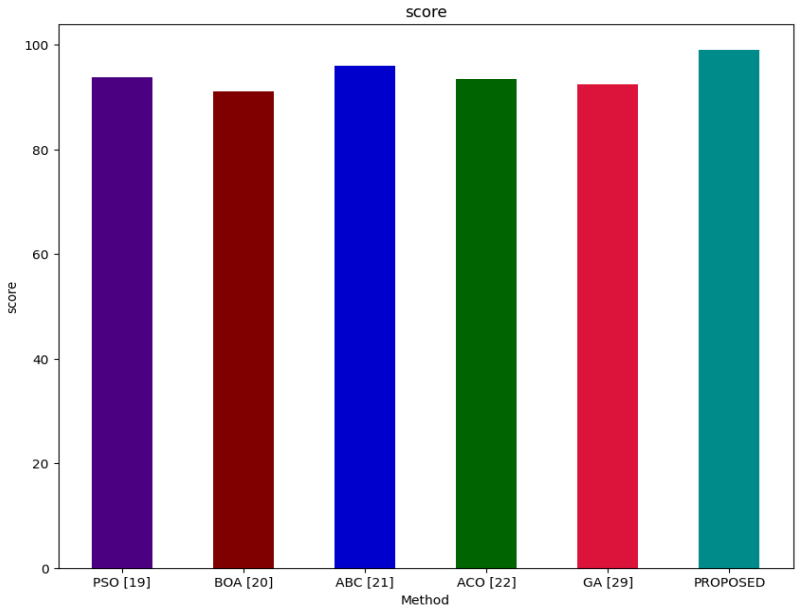

Figure 6: Performance Score Analysis

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

The new way does better than other common methods, showing that it is better at dealing with changing or confused situations. It is a high performer in many ways, giving a performance mark of approximately 98, compared to PSO (around 92.5), BOA (about 91), ABC (93), and ACO (92). GA only goes up to around 80. Good performance has been achieved by the method he proposed as it is among the methods that have managed to effectively solve a variety of navigation problems. Conversely, PSO, BOA, ABC, ACO, GA have lower scores which demonstrate their inability to perform well regarding optimal navigation performance. Our findings show that our new method can be used in multi-robot navigation tasks as it offers better results in comparison with other known methods such as PSO etc, hence making it suitable for real-world scenarios due to its performance in terms of both efficiency and effectiveness which are key factors distinguishing it from others.

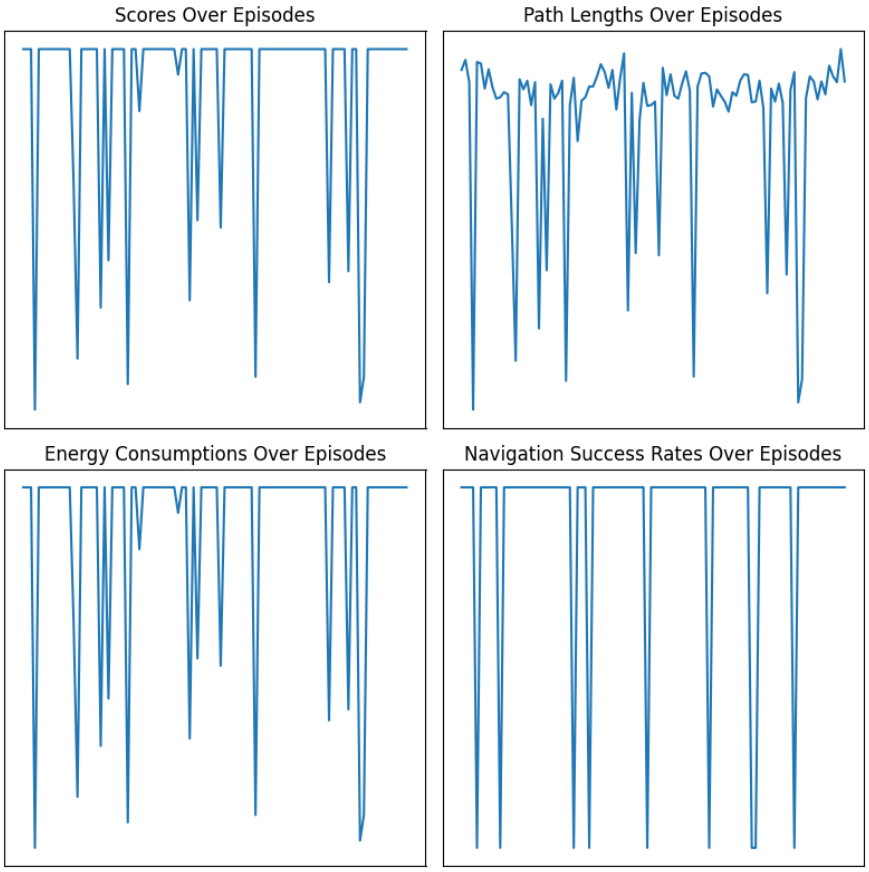

Figure 7: Metrics Vs Episode analysis

By plotting several indicators against episode numbers, Figure 7 gives an all-around view of the performance of the new approach through various episodes. Path length is useful in indicating on how good robots are at moving along some distance while navigation success rate is used to determine effectiveness in accomplishing duties. Energy consumption shows how resource efficiency is done with the performance score helping to give an all-round view through different factors.

5. Conclusions

This paper introduced a method that greatly improves job assignment and robot movement in complex and constantly changing environments. This new approach addresses key issues related to free-roaming robots by combining Model-Based Reinforcement Learning for realistic motion planning with Object

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

Detection using Transformers for precise obstacle avoidance. The MBRL strategy uses DQN trained with a physics simulator to help robots navigate effectively in adaptable environments, while the ODT module relies on transformer-based architecture to accurately identify obstacles. Additionally, a fitness function is proposed, taking into account total task completion time, path length, energy usage, robot idle time and urgency to optimize the system's performance. The simulation results demonstrate the methodology's effectiveness through improved task efficiency, enhanced navigation accuracy and resource management. These characteristics position this work at the forefront of multi-robot systems with minimal computational requirements. Therefore, the proposed method addresses everyday needs across various environments such as homes, warehouses or disaster zones.

References

1. Agrawal A, Hariharan S, Bedi AS, Manocha D. Dc-mrta: Decentralized multi-robot task allocation and navigation in complex environments. In2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2022 Oct 23 (pp. 11711-11718). IEEE.
2. Chakraa H, Guérin F, Leclercq E, Lefebvre D. Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. Robotics and Autonomous Systems. 2023 Jul 23:104492.
3. Dalmasso M, Domínguez-Vidal JE, Torres-Rodríguez IJ, Jiménez P, Garrell A, Sanfeliu A. Shared task representation for human–robot collaborative navigation: The collaborative search case. International Journal of Social Robotics. 2024 Jan;16(1):145-71.
4. Schmidbauer C, Zafari S, Hader B, Schlund S. An Empirical Study on Workers' Preferences in Human–Robot Task Assignment in Industrial Assembly Systems. IEEE Transactions on Human-Machine Systems. 2023 Jan 2;53(2):293-302.
5. Quinton F, Grand C, Lesire C. Market approaches to the multi-robot task allocation problem: a survey. Journal of Intelligent & Robotic Systems. 2023 Feb;107(2):29.
6. Agrawal A, Bedi AS, Manocha D. Rtaw: An attention inspired reinforcement learning method for multi-robot task allocation in warehouse environments. In2023 IEEE International Conference on Robotics and Automation (ICRA) 2023 May 29 (pp. 1393-1399). IEEE.
7. Chakraa H, Guérin F, Leclercq E, Lefebvre D. Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. Robotics and Autonomous Systems. 2023 Jul 23:104492.
8. Chakraa H, Guérin F, Leclercq E, Lefebvre D. Optimization techniques for Multi-Robot Task Allocation problems: Review on the state-of-the-art. Robotics and Autonomous Systems. 2023 Jul 23:104492.
9. Dalmasso M, Domínguez-Vidal JE, Torres-Rodríguez IJ, Jiménez P, Garrell A, Sanfeliu A. Shared task representation for human–robot collaborative navigation: The collaborative search case. International Journal of Social Robotics. 2024 Jan;16(1):145-71.
10. Wang Z, Liu C, Gombolay M. Heterogeneous graph attention networks for scalable multi-robot scheduling with temporospatial constraints. Autonomous Robots. 2022 Jan;46(1):249-68.
11. Nabi S, Ahmad M, Ibrahim M, Hamam H. AdPSO: adaptive PSO-based task scheduling approach for cloud computing. Sensors. 2022 Jan 25;22(3):920.
12. Salehnia T, Seyfollahi A, Raziani S, Noori A, Ghaffari A, Alsoud AR, Abualigah L. An optimal task scheduling method in IoT-Fog-Cloud network using multi-objective moth-flame algorithm. Multimedia Tools and Applications. 2024 Apr;83(12):34351-72.

Journal of Information Systems Engineering and Management
2025, 10(60s)

e-ISSN: 2468-4376

https://www.jisem-journal.com/

Research Article

13. Vaisi B. A review of optimization models and applications in robotic manufacturing systems: Industry 4.0 and beyond. Decision analytics journal. 2022 Mar 1;2:100031.
14. Fang Y, Ming H, Li M, Liu Q, Pham DT. Multi-objective evolutionary simulated annealing optimisation for mixed-model multi-robotic disassembly line balancing with interval processing time. International Journal of Production Research. 2020 Feb 1;58(3):846-62.
15. Evangelou G, Dimitropoulos N, Michalos G, Makris S. An approach for task and action planning in human–robot collaborative cells using AI. Procedia Cirp. 2021 Jan 1;97:476-81.
16. Motes J, Sandström R, Lee H, Thomas S, Amato NM. Multi-robot task and motion planning with subtask dependencies. IEEE Robotics and Automation Letters. 2020 Feb 26;5(2):3338-45.
17. Nonoyama K, Liu Z, Fujiwara T, Alam MM, Nishi T. Energy-efficient robot configuration and motion planning using genetic algorithm and particle swarm optimization. Energies. 2022 Mar 11;15(6):2074.
18. Zitouni F, Harous S, Maamri R. A distributed approach to the multi-robot task allocation problem using the consensus-based bundle algorithm and ant colony system. IEEE Access. 2020 Feb 4;8:27479-94.
19. Wei C, Ji Z, Cai B. Particle swarm optimization for cooperative multi-robot task allocation: a multi-objective approach. IEEE Robotics and Automation Letters. 2020 Feb 10;5(2):2530-7.
20. Shareh MB, Bargh SH, Hosseinabadi AA, Slowik A. An improved bat optimization algorithm to solve the tasks scheduling problem in open shop. Neural Computing and Applications. 2021 Mar;33:1559-73.
21. Szczepanski R, Erwinski K, Tejer M, Bereit A, Tarczewski T. Optimal scheduling for palletizing task using robotic arm and artificial bee colony algorithm. Engineering Applications of Artificial Intelligence. 2022 Aug 1;113:104976.
22. Cao R, Li S, Ji Y, Zhang Z, Xu H, Zhang M, Li M, Li H. Task assignment of multiple agricultural machinery cooperation based on improved ant colony algorithm. Computers and Electronics in Agriculture. 2021 Mar 1;182:105993.
23. Elfakharany A, Ismail ZH. End-to-end deep reinforcement learning for decentralized task allocation and navigation for a multi-robot system. Applied Sciences. 2021 Mar 24;11(7):2895.
24. Okubo T, Takahashi M. Multi-Agent Action Graph Based Task Allocation and Path Planning Considering Changes in Environment. IEEE Access. 2023 Feb 27;11:21160-75.
25. Lei T, Chintam P, Luo C, Liu L, Jan GE. A convex optimization approach to multi-robot task allocation and path planning. Sensors. 2023 May 26;23(11):5103.
26. Pradhan B, Goswami V, Barik RK, Sahana S. An integrated strategy-based game-theoretic model and decentralized queueing system for mobile multi-robot task coordination. Decision Analytics Journal. 2023 Jun 1;7:100254.
27. Valero O, Antich J, Tauler-Rosselló A, Guerrero J, Miñana JJ, Ortiz A. Multi-robot task allocation methods: A fuzzy optimization approach. Information Sciences. 2023 Nov 1;648:119508.
28. Faccio M, Granata I, Minto R. Task allocation model for human-robot collaboration with variable cobot speed. Journal of Intelligent Manufacturing. 2024 Feb;35(2):793-806.
29. Chakraa H, Leclercq E, Guérin F, Lefebvre D. A Centralized Task Allocation Algorithm for a Multi-Robot Inspection Mission With Sensing Specifications. IEEE Access. 2023 Sep 13.