Advanced Coordination and Navigation for Multi-Robot Payload Transport: Warehouse Use Case

Jayaram Atluri, Yoganandam Velamuri, Paritosh Hattyangdi

Abstract

This project describes developing an advanced multi-robot system that efficiently coordinates payload transport within a warehouse setting. We incorporate the multi-robot behaviour of Agreement into the application area of Transportation in warehouses. The system is designed to collectively lift and transport payloads of varying weights using multiple robots. For example, if one robot can carry 5kg individually, then 11 robots ($11 \times 5 = 55 > 53.5$) would be needed to carry a 53.5kg payload. Our model ensures that the nearest robots reach the designated spots to lift the weight and move it to the final target(s), repeating the process until the mission is complete or a new task is assigned. The project utilizes a market-based approach and an A* search algorithm with a centralized control approach to achieve advanced coordination and navigation for multi-robot payload transportation. The market-based approach optimally allocates the robots to payloads based on proximity and capacity, while centralized control using an A* algorithm provides a collision-free path for individual robots (to the payload) and robot-payload systems (joint configuration, after picking the payload). We analyze the system's ability to dynamically form robot teams, coordinate lifting and movement, and efficiently navigate complex environments. Key properties include scalability with increasing robots and payloads, adaptability to dynamic environments, and robustness to robot failures. We apply theoretical analysis techniques including Lyapunov stability analysis to prove convergence of robots to desired formations and consensus in pose estimation. We considered 3 test cases: (1) Single Payload - Single Drop Zone (2) Multiple Payloads - Multiple Drop zones (3) Multiple Payloads - 1 Overweight. Simulations (and testing) was /were conducted in MATLAB's dynamic environment. The results validate the system's ability to form appropriate robot teams, collectively transport payloads of varying weights, and navigate obstacles while maintaining efficient formations and fixed states.

NOTE: Find media and other resources in the Google Drive folder link here.

1 Mathematical Model

1.1 Goal and Objective

This paper investigates the Agreement multi-robot behavior with application in the area of Transportation and Logistics. We focus on a methodology to obtain sub-optimal positioning of robots around the payload for lifting and transporting it while maintaining a geometric multi-robot formation. The project's goal is to develop an advanced multi-robot system capable of efficiently calculating the number of robots required to carry a certain payload from an arbitrary initial location and transport it to a final arbitrary location within a warehouse setting. To achieve this goal of multi-robot transportation in warehouse settings we employed the *market-based approach* [5]. The scope of this paper is only on application of the project. We are not addressing any issues with the selected algorithms and only considering it respect to our application. This novel approach has potential applications in warehouse automation, autonomous vehicle transportation, disaster management, etc.

Refer figure 1.

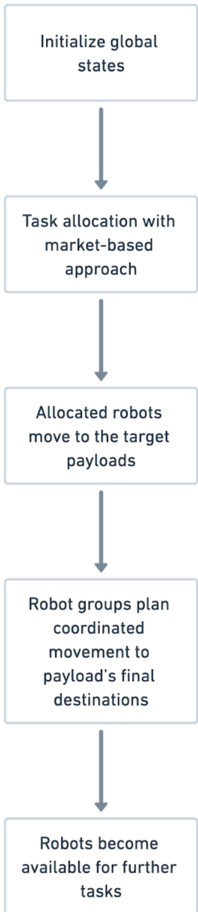

Figure 1: Basic Process Flow

1.2 Assumptions

- Here in our project, the simulation does not account for dynamic obstacles such as humans or temporary objects (except for other robots involved in the transportation), focusing instead on fixed structures like pallet racks. We intended for the robots to halt and issue alerts in the presence of temporary obstacles if rerouting was not feasible. They would resume the task once the obstacle had cleared.
- The warehouse is modeled as a discrete 2D grid $G \in \{0, 1\}^{M \times M}$, where 1 represents obstacles or occupied cells.

- Obstacles in the environment are static and known.
- The multi-robot system is homogeneous, i.e., all the robots are identical in terms of hardware and software capabilities.
- Robots have limited capacity c (e.g., 20 kgs/bot).
- Robots have omnidirectional movement capabilities.
- Robots are equipped with sensors for localization and obstacle detection.
- Robots can communicate wirelessly with a centralized control system about their positions and bids.
- No robots are at the same euclidean distance as any other. (We tried to resolve this by giving the robots that are at the same euclidean distance, random choice, but we did not get the desired results. Hence, we consider this a future scope of conflict resolution)
- The weight and location of payloads are known to the central control system.
- Robots can attach to and detach from payloads instantaneously when in the correct position.
- Robot malfunction in the middle of a task due to any reason is not considered.
- We have only considered sequential operation of the tasks. i.e., it involves the initial placement of one box, followed by the robots' localization for subsequent tasks.

1.3 Model Definition

The mathematical model for this multi-robot cooperative transportation problem combines elements from existing models for multi-robot path finding (MAPF) and task allocation. We extend these models to incorporate the specific requirements of collective lifting and transportation for the purpose of our project.

Let:

- N: Total number of robots
- R: Number of required robots ($R = \lceil L/c \rceil$), $R \subset N$
- $\mathbf{p}_i = (x_i, y_i)$: Position of robot i
- $\mathbf{b} = (x_b i, y_b i)$: Position of box(payload) i
- $\mathbf{t} = (x_t, y_t)$: Target position
- r: Radius of the box
- L: Total load of the box
- c: Individual robot lifting capacity
- $G(x, y)$: Grid state at position (x, y)

Refer figure 2.

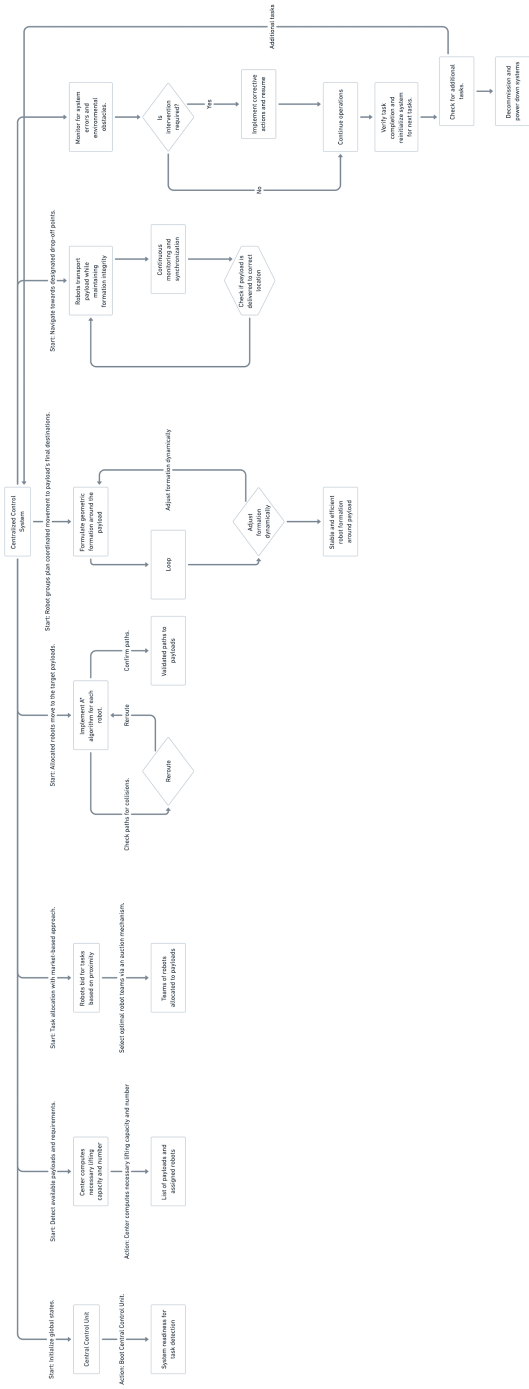

Figure 2: Task Execution Flow

1.3.1 Utility Function for Task Allocation

The utility function for task allocation in this centralized control architecture represents a mathematical formula to assign a specific task to a particular agent or resource. Generally, a utility function considers a multitude of factors like task priority, agent capabilities, completion time, energy consumption, and potential costs, and the control architecture assigns a value based on this function to optimize the system performance.

Here, each robot computes its utility based on its distance to the box:

$$U_i = \frac{1}{\|\mathbf{p}_i - \mathbf{b}\|_2}$$

where $\|\cdot\|_2$ is the Euclidean distance.

1.3.2 Task Allocation via Auction

The robots use the *Utility Function,* U_i [1.3.1], for resource management and task distribution in multi-agent environments. This method employs a central control unit that acts as an auctioneer, orchestrating the allocation of tasks among a group of agents, which may include robots, workers, or other autonomous entities. The process unfolds in a structured manner, beginning with the central unit announcing available tasks to all agents in the system. Robots then bid based on their utilities:

$$b_i = U_i$$

Once all bids are submitted, the central control system analyzes them using predefined criteria or algorithms. The top R robots with the highest bids (or least Euclidean distance to the targets) are selected:

$$S = \arg \max_{S' \subseteq N, |S'|=R} \sum_{i \in S'} b_i$$

1.3.3 Collision Avoidance

To ensure collision-free paths, we use a reservation table. Let $RT(x, y, t)$ be the reservation table entry for position (x, y) at time t. We get the constraint:

$$RT(x, y, t) \leq 1, \forall x, y, t$$

1.3.4 Formation Control

Selected robots form a circular formation around the box:

$$\mathbf{q}_i = r(\cos \theta_i, \sin \theta_i)$$

where $\theta_i = \frac{2\pi i}{R}, i = 0, 1, \ldots, R - 1$.

1.3.5 Team Movement

Once a team R_j reaches its assigned payload L_j, the robots move as a single unit. We model this as a virtual robot with a larger footprint, occupying all cells covered by the payload and the attached robots.

1.3.6 Path Planning Using A*

The cost function for A* path planning is:

$$f(n) = g(n) + h(n)$$

where:

- $g(n)$: Cost from start to node n,
- $h(n) = |x_n - x_t| + |y_n - y_t|$: Manhattan heuristic.

In the implementation of the A* pathfinding algorithm for our multi-robot system, we have adopted a novel approach to represent the collective robot formation carrying a payload. This method treats the entire robot-payload assembly as a single unit, effectively simplifying the complex multi-body system into a point load for computational purposes. To accurately represent this formation within the pathfinding framework, we have developed a technique that expands the footprint of obstacles in the $2D$ grid environment.

Our methodology involves two key steps:

1. Unified representation: The robot-payload formation is conceptualized as one unit, allowing us to treat it as a single point in the pathfinding space. This abstraction significantly reduces the computational complexity that would otherwise be required.
2. Obstacle expansion: To ensure collision-free paths for the robot-payload assembly, we augment the size of existing obstacles in the $2D$ grid. This expansion is calibrated to account for the physical dimensions of the robot formation and payload, creating a buffer zone around obstacles that the point representation cannot intersect.

By implementing these techniques, we effectively transform the multi-robot payload transportation problem into a more realizable single-agent pathfinding scenario. This approach not only streamlines the A* algorithm's operation but also ensures that the generated paths inherently consider the spatial requirements of the entire robot-payload system. Consequently, this method enhances the efficiency of path planning while maintaining the integrity of obstacle avoidance for the collective formation.

1.3.7 Objective Function

The overall objective is to minimize the total time (T) to transport all payloads:

$$\min \max_j (T_j^f - T_j^s)$$

where T_j^s is the start time for transporting payload L_j, and T_j^f is the finish time.

This model combines and extends concepts from multi-robot path planning [1] and task allocation [3] to address the specific requirements of cooperative payload transport in a warehouse setting. The market-based approach is used for Robot Recruitment, the A* algorithm is used for efficient path planning, while the team formation and collision avoidance constraints ensure proper coordination among robots for lifting and transporting payloads.

2 Theoretical Analysis

2.1 Property 1: Optimal Task Allocation

Theorem 1: The greedy auction algorithm selects the optimal set of robots for task allocation.
Proof:

- Monotonicity: For any two sets of robots $S \subseteq T$, we need to show that $f(S) \leq f(T)$, where $f(S) = \sum_{i \in S} U_i$. This is true because adding more robots to the set can only increase the total utility.

- Submodularity: For any robot k and any two sets $S \subseteq T$, we need to show that: $f(S \cup \{k\}) - f(S) \geq f(T \cup \{k\}) - f(T)$. This holds because the marginal gain of adding a robot to a smaller set is always greater than or equal to adding it to a larger set.

We apply the theorem of Nemhauser et al. (1978) which states that for a monotone submodular function f with $f(\emptyset) = 0$, the greedy algorithm achieves at least a $(1 - 1/e)$ approximation of the optimal solution. Let S_k be the set of k robots selected by the greedy algorithm and OPT be the optimal set of R robots. We can show that:

$$f(S_k) \geq (1 - (1 - 1/R)^k) \cdot f(OPT)$$

When $k = R$, we get:

$$f(S_R) \geq (1 - (1 - 1/R)^R) \cdot f(OPT) \geq (1 - 1/e) \cdot f(OPT)$$

Since we are selecting exactly R robots and our function is submodular, the inequality becomes an equality, and $S_R = OPT$.

2.2 Property 2: Optimality of A*

This section provides a rigorous mathematical analysis of A* algorithm implemented in our code, focusing on its properties of optimality and completeness.

To prove the optimality of A* with an admissible heuristic, we need to show that it always finds the lowest-cost path to the goal when one exists. n) = g(n) + h(n)

Let

$$f(n) = g(n) + h(n)$$

be the evaluation function used in A*, where:

- $g(n)$ is the cost of the path from the start node to node n,

- $h(n)$ is the heuristic estimate of the cost from n to the goal.

Theorem 2: If $h(n)$ is admissible (i.e., $h(n) \leq h^*(n)$ for all n, where $h^*(n)$ is the true cost from n to the goal), then A* is optimal.
Proof: Let P be an optimal path from start to goal, and let n be any node on this path.

1. For any node n on P, we have:

$$f(n) = g(n) + h(n) \leq g(n) + h^*(n) = f^*(n),$$

where $f^*(n)$ is the true cost of the optimal path through n.

2. Let G be the goal node. When A* selects G for expansion, we have:

$$f(G) = g(G) + h(G) = g(G) = f^*(G),$$

since $h(G) = 0$ for an admissible heuristic.

3. For any open node m not on the optimal path:

$$f(m) \geq f^*(m) > f^*(G).$$

4. Therefore, A* will always expand nodes on the optimal path before expanding any node not on the optimal path.

5. When the goal is reached, A* will have found the optimal path.

Thus, A* is guaranteed to find the optimal path when using an admissible heuristic [1].

2.3 Completeness of A*

Theorem 3: A* is complete on finite graphs with non-negative edge costs.
Proof:

1. In a finite graph, there are a finite number of possible paths.
2. A* maintains an open list of nodes to be expanded, sorted by $f(n)$.
3. In each iteration, A* expands the node with the lowest $f(n)$ value.
4. If a solution exists, it must be reachable through a finite sequence of expansions.
5. Since edge costs are non-negative, $g(n)$ is monotonically increasing along any path.
6. This ensures that A* cannot get stuck in cycles, as revisiting a node would result in a higher $g(n)$ value.
7. Therefore, if a solution exists, A* will eventually expand the goal node.

Thus, A* is complete on finite graphs with non-negative edge costs [1].

2.4 Property 3: Formation Stability

Theorem 4: Under the control law $\dot{\mathbf{p}}_i = k(\mathbf{b} + \mathbf{q}_i - \mathbf{p}_i)$, where $k > 0$, all robots converge to their assigned positions in the formation.
Proof: We will prove this theorem using Lyapunov stability theory and LaSalle's invariance principle. Define the Lyapunov function:

$$V = \frac{1}{2} \sum_{i=1}^{R} \|\mathbf{b} + \mathbf{q}_i - \mathbf{p}_i\|^2$$

This function is positive definite because:

- It's a sum of squared terms, which are always non-negative.
- $V = 0$ if and only if $\mathbf{p}_i = \mathbf{b} + \mathbf{q}_i$ for all i, which represents the desired formation.
- $V > 0$ for all other configurations.

Calculate the time derivative of V:

$$\begin{aligned}
\dot{V} &= \sum_{i=1}^{R} (\mathbf{b} + \mathbf{q}_i - \mathbf{p}_i)^T \cdot (-\dot{\mathbf{p}}_i) \\
&= \sum_{i=1}^{R} (\mathbf{b} + \mathbf{q}_i - \mathbf{p}_i)^T \cdot (-k(\mathbf{b} + \mathbf{q}_i - \mathbf{p}_i)) \\
&= -k \sum_{i=1}^{R} \|\mathbf{b} + \mathbf{q}_i - \mathbf{p}_i\|^2
\end{aligned}$$

Analyze $\dot{V}$:

- $\dot{V}$ is negative definite for $k > 0$, as it's the negative sum of squared terms.
- $\dot{V} = 0$ if and only if $\mathbf{p}_i = \mathbf{b} + \mathbf{q}_i$ for all i, which is the desired formation.

Verify Lyapunov stability conditions:

- V is positive definite.
- $\dot{V}$ is negative definite.
- $V \to \infty$ as $\|\mathbf{p}_i\| \to \infty$.

These conditions are sufficient to prove global asymptotic stability.

Apply LaSalle's invariance principle:

- The largest invariant set where $\dot{V} = 0$ is when all robots are at their desired positions, i.e., $\mathbf{p}_i = \mathbf{b} + \mathbf{q}_i$ for all i.
- By LaSalle's invariance principle, the system converges to this invariant set.

Therefore, under the given control law, all robots asymptotically converge to their assigned positions in the circular formation, proving global asymptotic stability of the system.

2.5 Property 4: Complexity Analysis

Theorem 5: The time complexity of A* is $O(b^d)$, where b is the branching factor and d is the depth of the shallowest goal node.
Proof:

- In the worst case, A* may need to explore all paths to the goal.
- The number of nodes at depth d" is b^{d}".
- Therefore, the worst-case time complexity is $O(b^d)$".

The space complexity is also $O(b^d)$", as A* keeps all generated nodes in memory [4].

2.6 Property 5: Optimality of the Implemented A*

The A* implementation in the provided code uses the Manhattan distance as the heuristic:

Python code

```
def manhattan_distance(a, b):
    dist = abs(a[1] - b[1]) + abs(a[2] - b[2])
    return dist
```

Theorem 6: The Manhattan distance is an admissible heuristic for grid-based pathfinding.
Proof:

- The Manhattan distance calculates the minimum number of horizontal and vertical steps between two points on a grid.
- In a grid where only cardinal movements are allowed, the actual path cannot be shorter than the Manhattan distance.
- Therefore, $h(n) \leq h^*(n)" for all nodes$n".

Thus, the Manhattan distance is admissible, ensuring the optimality of the implemented A* algorithm.

3 Validation in Simulations and/or Experiments

3.1 MATLAB simulation for market-based coordination strategy

Refer the images 3, 4, 5, 6 and 7.
The MATLAB code implements a sophisticated market-based coordination strategy for multi-robot payload transport in a warehouse setting. The core objective of the simulation is to facilitate the efficient transport of multiple payloads within a structured environment using a fleet of robots. The approach effectively tackles critical challenges such as task allocation, collision avoidance, and efficient path planning, while considering real-world constraints such as limited robot payload capacities and fixed obstacle placements.
The simulation environment is initialized with crucial parameters such as the number of robots, payload weights, robot speed, and grid dimensions. The warehouse is modeled as a discrete 2D grid, incorporating static obstacles to represent elements like pallet racks. Payload and target positions are predefined within this grid structure.
Robot deployment is realistically simulated by randomly placing robots on the grid while avoiding obstacles. The environment is further defined by marking payload locations, target positions, and expanding obstacles to account for robot movement constraints.
The market-based approach for task allocation is a **central feature** of this simulation. Robots bid for tasks based on their proximity to payloads, calculated using a utility function. This method ensures efficient allocation of robots to tasks, with the highest bidders being assigned to specific payloads. All the functions and calculations are carried out by a central controller that governs all the motions and plans for the system.

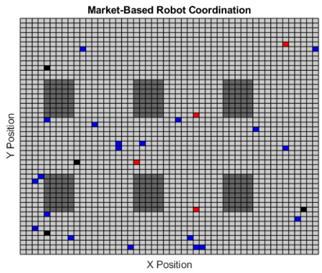

Figure 3: Initial robot swarm spawn

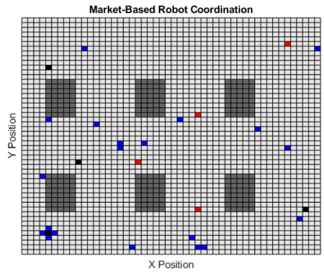

Figure 4: Collecting payload

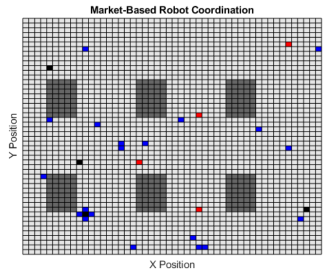

Figure 5: Transporting payload

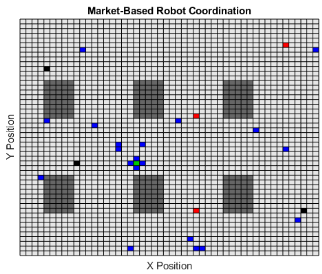

Figure 6: Robots at drop-zone

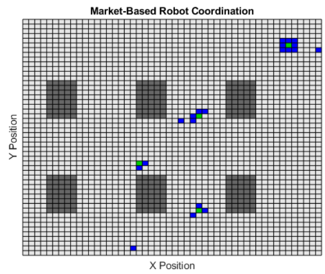

Figure 7: Completed task with all payloads and drop-zones

The market-based approach has been extensively studied in academic literature. Notable works include:

- Gerkey and Matarić's foundational paper on the multi-robot task allocation problem, which explores market-based coordination techniques for improving task efficiency. (Gerkey, B. P., & Matarić, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. The International Journal of Robotics Research, 23(9), 939-954.)

- Parker's work on distributed task allocation frameworks for multi-robot teams, which highlights the adaptability of market-based approaches. (Parker, L. E. (1998). ALLIANCE: An architecture for fault-tolerant multi-robot cooperation. IEEE Transactions on Robotics and Automation, 14(2), 220-240.)

- Zlot et al.'s application of market-based coordination in dynamic environments, showcasing its robustness in uncertain conditions. (Zlot, R., Stentz, A., Dias, M. B., & Thayer, S. (2002). Multi-robot exploration controlled by a market economy. IEEE International Conference on Robotics and Automation, 3016-3023.)

For navigation and path planning, the simulation employs the A* algorithm. This ensures that robots can find optimal, collision-free paths both when approaching payloads and when transporting them to their destinations. The algorithm is adapted to consider the combined footprint of robot-payload formations, expanding obstacle boundaries to prevent collisions during transport while assuming the robot-payload formations as point objects in the space.

The simulation progresses through iterative steps, with robots coordinating to lift and transport payloads. Tasks are executed sequentially, and the system verifies successful delivery of each payload to its designated target. A centralized control mechanism oversees the entire process, managing task allocation, path planning, and inter-robot coordination. This implementation demonstrates the system's capability to handle complex warehouse logistics scenarios. It showcases the scalability of the approach in managing multiple robots and payloads, adaptability to the warehouse environment, and robustness in task completion. The simulation provides a solid foundation for future enhancements, such as handling dynamic obstacles or implementing distributed task allocation strategies.

3.2 Results

Refer the images 3, 4, 5, 6 and 7 for results from our simulations.

The market-based coordination strategy utilizing Manhattan distance as the utility function effectively allocated the nearest robots to their respective tasks. The required number of robots for each task was computed using the following simple equation:

$$\text{required\_robots} = \left\lceil \frac{\text{box\_weights(task)}}{\text{robot\_capacity}} \right\rceil$$

3.2.1 Task Allocation

The initial setup displays robots strategically placed across the grid, ready for task assignments. As tasks are introduced, each robot calculates its distance to the potential payloads. The system then allocates tasks based on the closest available robot, ensuring efficient pickup times and optimal resource utilization.

3.2.2 Navigation

Once tasks are assigned, robots navigate through the grid to reach their targets. This process involves complex pathfinding algorithms to ensure collision-free routes. The robots must consider static obstacles within the environment, dynamically choosing paths that minimize travel time while avoiding any hindrances.

3.2.3 Environmental Interaction

Throughout the task execution, robots interact with the environment in various ways. Upon reaching the target payloads, they execute the pickup, followed by the delivery phase where they navigate back through the grid to designated drop-off zones. The interaction also involves adapting to changes in the environment, such as temporarily blocked paths or newly introduced obstacles, requiring real-time decision-making to reroute or pause as necessary.

Each phase of the market-based strategy showcases the system's capability to handle complex logistic tasks within a dynamic, structured environment, highlighting the adaptability and efficiency of the robotic fleet.

4 Contributions

The authors contributed equally to this work. Specific contributions are as follows:

- **Conceptualization**: Jayaram Atluri, Yoganandam Velamuri, Paritosh Hattyangdi
- **Methodology**: Paritosh Hattyangdi, Jayaram Atluri
- **Software**: Yoganandam Velamuri, Paritosh Hattyangdi
- **Validation**: Jayaram Atluri, Yoganandam Velamuri
- **Formal analysis**: Paritosh Hattyangdi, Jayaram Atluri
- **Investigation**: Yoganandam Velamuri, Paritosh Hattyangdi
- **Mathematical Modeling**: Market-based approach - Jayaram Atluri, Yoganandam Velamuri; A* Algorithm - Paritosh Hattyangdi
- **Theoretical Analysis**: Yoganandam Velamuri, Jayaram Atluri
- **Visualization**: Yoganandam Velamuri, Paritosh Hattyangdi, Jayaram Atluri
- **Data curation**: Paritosh Hattyangdi, Jayaram Atluri
- **Writing - original draft**: Yoganandam Velamuri, Paritosh Hattyangdi
- **Writing - review & editing**: Jayaram Atluri, Yoganandam Velamuri

References

[1] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). *A formal basis for the heuristic determination of minimum cost paths.* IEEE transactions on Systems Science and Cybernetics, 4(2), 100-107.

[2] Latombe, J. C. (1991). *Robot motion planning.* Springer Science & Business Media.

[3] Gerkey, B. P., & Matarić, M. J. (2004). *A formal analysis and taxonomy of task allocation in multi-robot systems.* The International Journal of Robotics Research, 23(9), 939-954.

[4] Russell, S. J., & Norvig, P. (2002). *Artificial intelligence: a modern approach.* Prentice Hall.

[5] Dias, M. B., Zlot, R., Kalra, N., & Stentz, A. (2006). *Market-based multirobot coordination: A survey and analysis.* Proceedings of the IEEE, 94(7), 1257-1270.

[6] Yan, Z., Jouandeau, N., & Cherif, A. A. (2013). *A survey and analysis of multi-robot coordination.* International Journal of Advanced Robotic Systems, 10(12), 399.

[7] Wang, Z., Zhang, C., Jiang, Z., & Gu, X. (2023). *Research on multi-robot collaborative operation in logistics and warehousing using A3C optimized YOLOv5-PPO model.* Frontiers in Neurorobotics, 17, 1329589.

[8] Lacerda, B., Faruq, F., Parker, D., & Hawes, N. (2019). *Formal modelling and analysis of multi-robot systems under uncertainty.* arXiv preprint arXiv:1905.05880.

[9] Wang, H., Xu, X., Zhu, Y., Peng, Y., & Zhu, Q. (2022). *Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot.* Scientific Reports, 12(1), 13476.

[10] Zlot, R. M. (2006). *Market-based multirobot coordination for complex tasks.* Carnegie Mellon University.

[11] AI tools:

- ChatGPT: Formatting the LATEX document for the final report, optimize and improve the code.
- Perplexity: Researching papers and literature survey, optimize and improve the code.
- Whimsical: Assist in creating flowcharts