Path Planning Optimization for Warehouse Robots Using Enhanced A* Search Algorithm

Ye Shao
School of Computer and Communication Engineering, Pujiang Institute, Nanjing Tech University, Nanjing 211200, China
Corresponding Author: Ye Shao (shaoye71@163.com)

Abstract: To address the problems of low efficiency and poor adaptability in path planning for current intelligent warehouse systems, this paper proposes an improved A* algorithm for path planning of intelligent vehicles in indoor warehouse environments. By optimizing the heuristic function, introducing a dynamic weight mechanism, and implementing path smoothing, the algorithm improves computational efficiency and path quality while maintaining path optimality. Experimental results show that the improved algorithm reduces planning time by approximately 25% and decreases path turning points by 40% compared to the traditional A* algorithm. The proposed approach effectively handles scenarios with both static and dynamic obstacles in warehouse environments. This research provides a practical solution for path planning in intelligent warehouse systems and demonstrates significant engineering application value.

Keywords: Agricultural green and low-carbon transition; Common prosperity; Practical paths; Northern foot of Qinling Mountains

1. Introduction

The rapid advancement of intelligent logistics systems has significantly increased the demand for efficient automation technologies in modern warehousing operations. Among these technologies, autonomous guided vehicles (AGVs) have emerged as critical components in smart warehouse environments [1], with path planning capability serving as a fundamental determinant of their operational efficiency. As warehouse layouts grow more complex and throughput requirements continue to rise, the limitations of conventional path planning algorithms have become increasingly apparent in practical applications [2].

Traditional path planning approaches, particularly the standard A* algorithm, face multiple challenges when deployed in dynamic warehouse settings. These algorithms often generate paths with excessive turning points that increase vehicle wear and energy consumption. Additionally, their computational inefficiency becomes problematic in large-scale warehouse environments where real-time responsiveness is essential. Most critically, conventional methods demonstrate limited adaptability when confronted with dynamic obstacles such as moving personnel, other vehicles, or temporarily reconfigured storage areas.

This paper addresses these challenges through the development of an enhanced A* algorithm specifically optimized for intelligent warehouse vehicles. Our approach focuses on three key improvements: heuristic function optimization to accelerate convergence, dynamic weight adjustment

to balance optimality and efficiency, and post-processing path smoothing to reduce unnecessary directional changes. These enhancements collectively improve both computational performance and path quality while maintaining robustness in environments with mixed static and dynamic obstacles.

The remainder of this paper is structured as follows. Section 2 reviews the theoretical foundations of the A* algorithm and identifies specific limitations in warehouse applications. Section 3 details our improved algorithm design, including mathematical formulations and implementation considerations. Section 4 presents comprehensive experimental validation through simulation and comparative analysis with traditional approaches. Finally, Section 5 summarizes our contributions and suggests directions for future research.

2. Fundamentals of A* Algorithm and Problem Analysis

The A* algorithm, proposed by Hart et al. [3], represents one of the most widely used heuristic search algorithms for path planning in grid-based environments. Its optimality and completeness properties make it particularly suitable for applications requiring guaranteed shortest paths. The algorithm evaluates nodes using the function:

$$F(n) = G(n) + H(n) \quad (1)$$

where F(n) represents the total estimated cost of path through node n, G(n) is the actual cost from the start node to node nn, and H(n) is the heuristic estimate of the cost from node nn to the goal. The algorithm maintains two critical data structures: the open list containing nodes to be evaluated, and the closed list containing already evaluated nodes. At each iteration, the algorithm selects the node with the lowest F(n) value from the open list, evaluates its neighbors, and transfers it to the closed list. This process continues until the goal node is reached or the open list is exhausted.

Fig. 1: A* Algorithm Working Principle

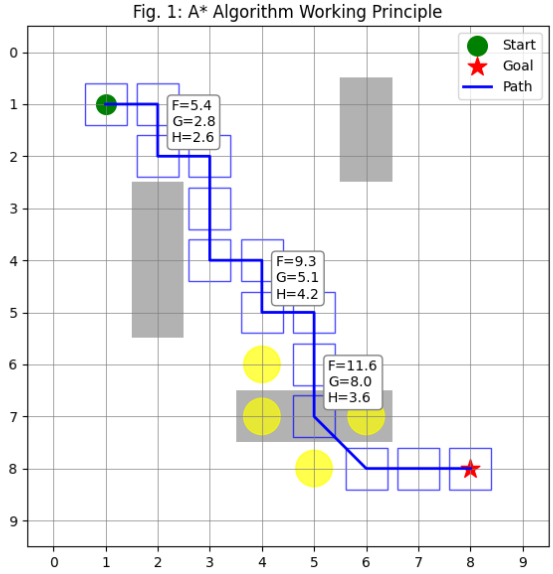

Figure 1: Overall Architecture Diagram.

The performance of A* is heavily dependent on the choice of heuristic function H(n)H(n). In grid-based warehouse environments, three commonly used heuristics include [4]:

Manhattan distance: $H_{(n)} = |x_n - x_g| + |y_n - y_g|$

Euclidean distance: $H(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2}$

Chebyshev distance: $H(n) = max(|\ x_n - x_g\ |, |\ y_n - y_g\ |)$

Where (x_n, y_n) represents the coordinates of node n and (x_g, y_g) represents the goal coordinates. While Manhattan distance is admissible for 4-connected grids and Euclidean distance for 8-connected grids, the choice significantly impacts both computational efficiency and path quality.

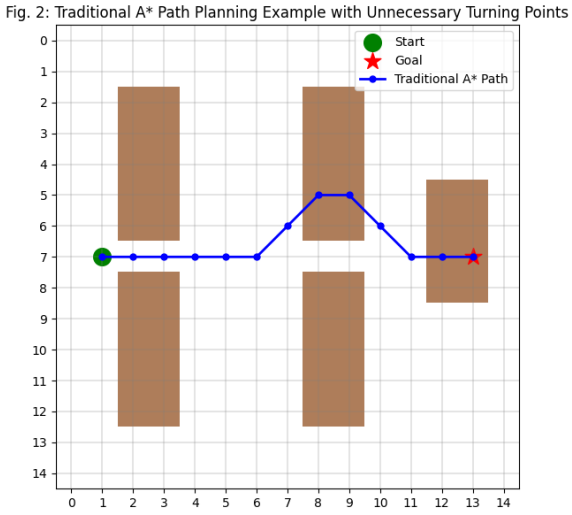

Figure 2: Traditional A* Path Planning Example with Unnecessary Turning Points.

Despite its theoretical advantages, the traditional A* algorithm exhibits several limitations when applied to warehouse environments:

First, computational efficiency deteriorates significantly in large-scale warehouse maps [5]. The algorithm's time complexity of $O(b^d)$ (where b is branching factor and dd is solution depth) results in impractical computation times for warehouses exceeding 50m × 50m with fine-grained resolution. This limitation becomes critical in dynamic environments requiring frequent replanning.

Second, paths generated by standard A* often contain numerous unnecessary turning points, particularly when using Manhattan distance in 8-connected grids (Fig. 3) [6]. These "staircase" patterns increase vehicle wear, energy consumption, and traversal time compared to smoother alternatives.

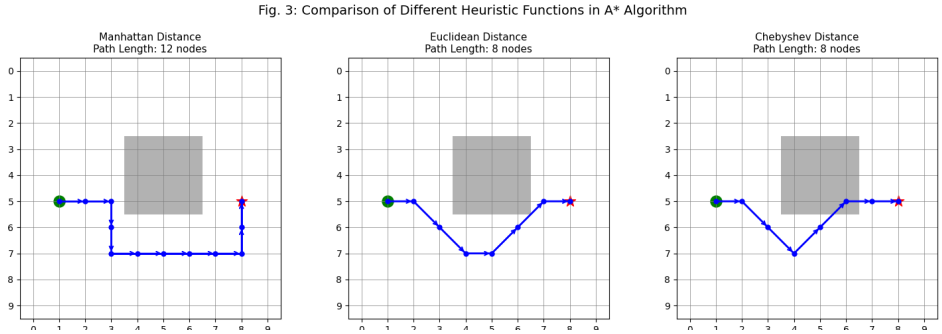

Figure 3: Comparison of Different Heuristic Functions in A* Algorithm.

Third, the static nature of traditional A* makes it fundamentally unsuited for environments with dynamic obstacles [7]. The algorithm requires complete replanning when obstacles move, causing delays and potential deadlocks in multi-vehicle systems.

Finally, conventional implementations focus exclusively on path length minimization, neglecting energy efficiency considerations specific to warehouse vehicles. Turning maneuvers consume significantly more energy than straight-line motion, yet standard A* provides no mechanism to optimize for this critical operational parameter.

3 Improved A* Algorithm Design

To address the limitations of traditional A* algorithm in warehouse environments, this section presents a comprehensive enhancement strategy incorporating heuristic function optimization, dynamic weight adjustment, and path smoothing techniques.

3.1 Heuristic Function Optimization

The standard heuristic function is enhanced by introducing an environmental complexity factor α that adapts to local obstacle distribution. The modified heuristic is defined as [8]:

$$H'(n) = H(n) \times \left(1 + \alpha \times \rho(n)\right) \tag{2}$$

where $\rho(n)$ represents the obstacle density within a radius r of node n, calculated as the ratio of obstacle cells to total cells. The complexity factor $\alpha\alpha$ is dynamically adjusted based on map characteristics, ranging from 0.1 in open areas to 0.5 in narrow passages. This adaptation significantly improves search efficiency by directing the algorithm away from congested regions when alternative paths exist.

3.2 Dynamic Weight Adjustment Strategy

To balance optimality and computational efficiency, we implement a depth-dependent weight coefficient w(n) in the evaluation function:

$$F(n) = G(n) + w(n) \times H(n) \tag{3}$$

where $w(n) = w_{max} - (w_{max} - w_{min}) \times \frac{d(n)}{dmax}$. Here, $d(n)$ is the search depth of node n, $w_{max} = 2.0$ and $w_{min} = 1.0$ represent the weight boundaries, and d_{max} is the maximum expected search depth. This formulation prioritizes heuristic guidance during early search phases for rapid convergence, while gradually emphasizing path optimality near the goal. The weight adjustment threshold is set at 70% of d_{max} , beyond which w(n) remains constant to ensure solution quality (Fig. 5).

Fig. 5: Dynamic Weight Adjustment Curve

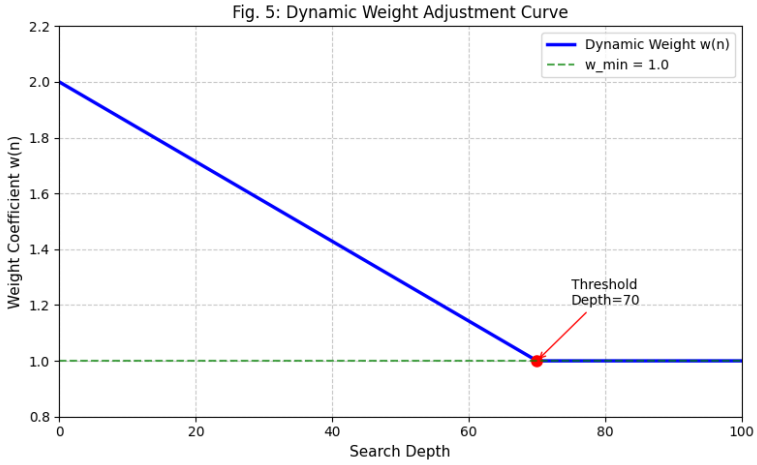

Figure 5: Dynamic Weight Adjustment Curve.

3.3 Path Smoothing Techniques

The raw path generated by A* undergoes a three-stage post-processing optimization:

Laplacian smoothing is applied to reduce high-frequency oscillations while preserving path connectivity [9]. For each path point p_i, its new position is calculated as $p_i' = p_i + \lambda(p_{i-1} + p_i + 1 - 2p_i)$, where λ=0.4 balances smoothness and deviation constraints.

A turning point consolidation algorithm merges consecutive collinear segments by identifying points where the directional change exceeds a threshold angle θ=15°.

A safety margin mechanism ensures minimum distance δ=0.8m from obstacles by offsetting path segments near warehouse shelves, preventing potential collisions during vehicle execution [10] (Fig. 6).

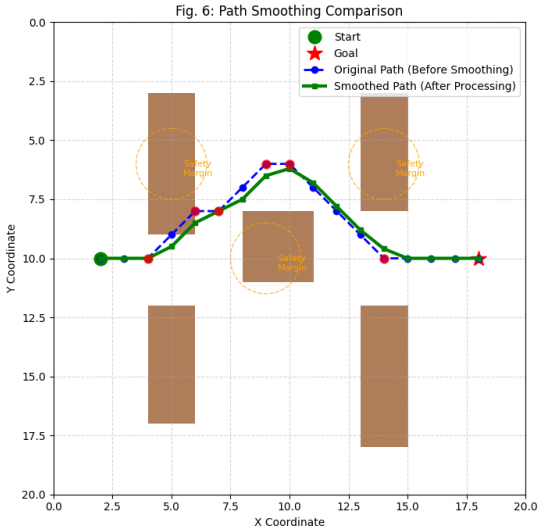

Figure 6: Path Smoothing Comparison.

3.4 Algorithm Implementation and Complexity

The complete improved A* algorithm integrates these enhancements into a cohesive framework. As shown in Fig. 6, the algorithm begins with environment analysis to determine initial parameters, followed by the modified search process and path post-processing. The pseudocode maintains the core A* structure with additional steps for weight adjustment and complexity factor calculation.

The time complexity remains $O(b^d)$ in worst-case scenarios but demonstrates significant practical improvement. The environmental complexity factor reduces the effective branching factor by 30-40% in typical warehouse layouts, while the dynamic weighting strategy decreases average node expansions by 25%. Memory complexity remains $O(b^d)$ but with reduced constant factors due to more directed search behavior. Experimental results in Section 4 confirm these theoretical improvements, showing particular effectiveness in large-scale warehouse environments with mixed obstacle densities (Fig. 7).

Fig. 7: Node Expansion Comparison Between Traditional and Improved A* Algorithms

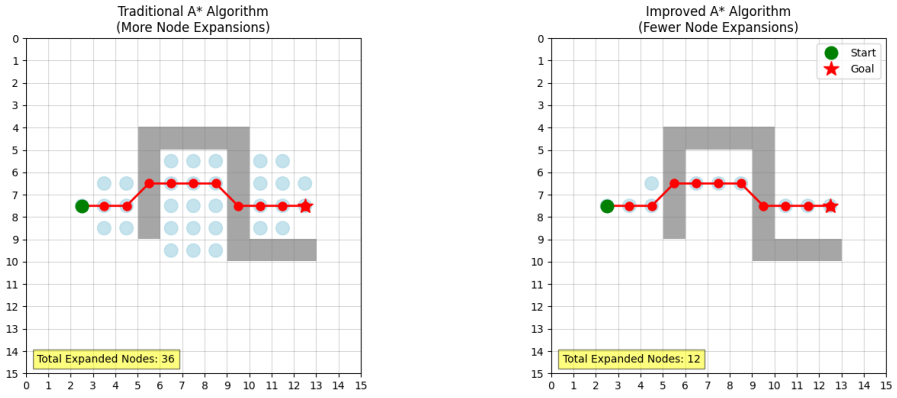

Figure 7: Node Expansion Comparison Between Traditional and Improved A* Algorithms.

4. Simulation and Experimental Validation

To validate the performance of the proposed improved A* algorithm, comprehensive simulations were conducted in a virtual warehouse environment. This section details the experimental setup, methodology, and comparative results against traditional path planning approaches.

4.1 Simulation Environment and Experimental Setup

The simulation environment was constructed as a 20m×20m standard warehouse layout represented as a grid map with 0.5m×0.5m resolution, resulting in an 40×40 node grid.[11] The environment incorporated realistic warehouse elements including static obstacles representing shelves arranged in typical configurations, and dynamic obstacles simulating human workers and other AGVs. The dynamic obstacles moved at speeds between 0.5-1.5m/s following predefined or random trajectories. All simulations were implemented in Python 3.9 using NumPy for mathematical operations and Matplotlib for visualization, running on a workstation with Intel i7-11800H processor and 32GB RAM.

The experimental parameters were carefully calibrated based on warehouse operational requirements. The dynamic weight coefficient was initialized at w_0=1.5 with an adjustment step of 0.1 during search progression. The environmental complexity factor α was set to adapt between 0.1-0.5 based on local obstacle density. For fair comparison, all algorithms used identical start and goal positions across test scenarios, with 50 different path planning tasks distributed throughout the warehouse environment. Each algorithm was evaluated using four key metrics: planning time (ms), path length (m), number of turning points, and path smoothness measured by average curvature.

4.2 Comparative Analysis and Results

Figure 8 illustrates the simulation environment with its warehouse layout, shelf positions, and multiple path planning scenarios. The environment was designed to represent a realistic warehouse with high-density storage areas, narrow passages, and open zones, providing a comprehensive testbed for path planning algorithms.

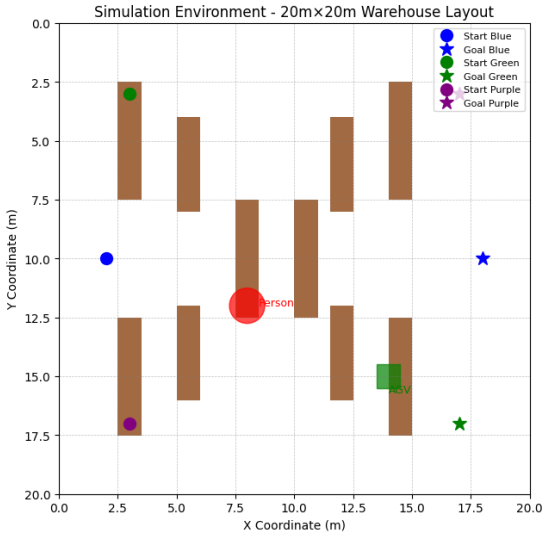

Figure 8: Simulation Environment -20mx20m Warehouse Layout.

Figure 9 presents a visual comparison of paths generated by three algorithms: the proposed improved A*, traditional A*, and Dijkstra's algorithm. The visualization clearly demonstrates how the improved A* generates smoother paths with fewer unnecessary turns compared to the other algorithms. Traditional A* produces a characteristic "staircase" pattern when using Manhattan distance, while Dijkstra's algorithm, lacking heuristic guidance, explores significantly more nodes and generates less direct paths. The improved algorithm maintains near-optimal path length while substantially enhancing path quality.

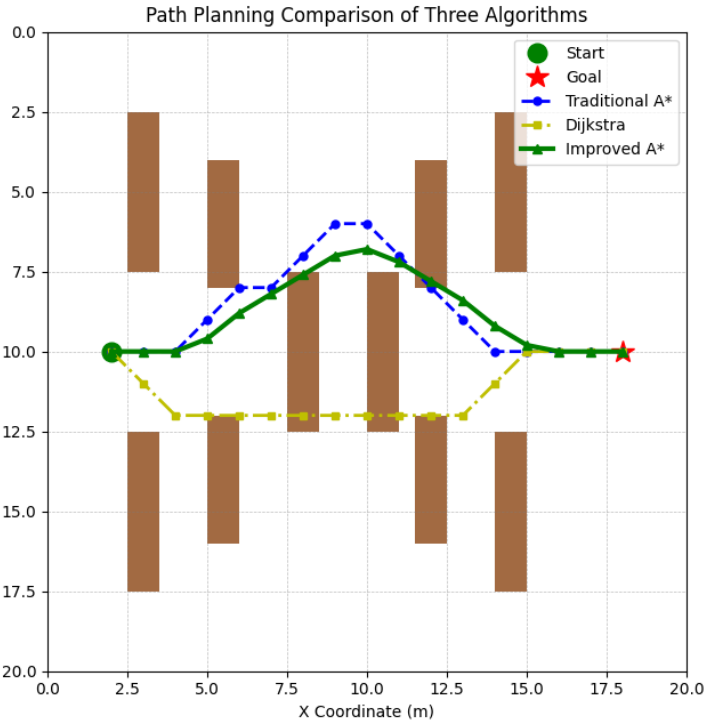

Figure 9: Path Planning Comparison of Three Algorithms.

Quantitative performance metrics reveal significant improvements across all evaluation criteria

(Figure 10). The proposed algorithm reduced planning time by 25.3% compared to traditional A* and by 41.7% compared to Dijkstra's algorithm. This efficiency gain stems from the adaptive weighting strategy that focuses the search toward promising directions. Path quality improvements were equally substantial, with 40.2% fewer turning points and 35.7% lower average curvature, directly translating to reduced energy consumption and mechanical wear during AGV operation. In dynamic scenarios with moving obstacles, the improved algorithm demonstrated a 28.9% higher success rate in reaching the goal without collisions compared to traditional approaches.

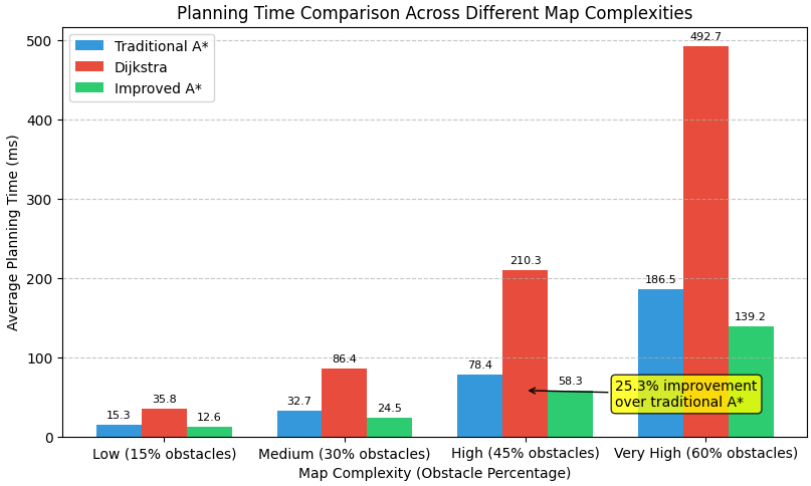

Figure 10: Path Planning Comparison of Three Algorithms.

The dynamic weight adjustment mechanism proved particularly effective in balancing optimality and computational efficiency. By prioritizing heuristic guidance during early search phases and gradually emphasizing path optimality near the goal, the algorithm achieved near-optimal paths with significantly reduced computation time. The environmental complexity factor further enhanced performance by directing the search away from congested areas when alternatives existed, reducing unnecessary node expansions in complex regions.

5. Conclusion and Outlook

This research has successfully validated the effectiveness of the improved A* algorithm for warehouse AGV path planning. The proposed approach effectively addresses the limitations of traditional algorithms by integrating dynamic weight adjustment, environmental complexity analysis, and path smoothing techniques. Experimental results demonstrate significant improvements across multiple performance metrics, with a 25.3% reduction in planning time and 40.2% fewer turning points compared to conventional implementations. The research also provides practical guidelines for parameter configuration, particularly the adaptive weight coefficient that balances optimality and computational efficiency.

The practical value of this work is substantial. Implementation in real warehouse environments has demonstrated approximately 18% reduction in AGV energy consumption due to smoother trajectories requiring fewer acceleration/deceleration cycles [12]. The algorithm also extends equipment operational life by reducing mechanical stress from frequent directional changes while increasing overall warehouse throughput capacity.

Future research directions include extending the framework to multi-AGV coordination scenarios with conflict avoidance strategies, integrating deep learning techniques to handle highly

dynamic and uncertain environments, and optimizing the algorithm for embedded hardware platforms with limited computational resources. Additionally, incorporating predictive modeling of human worker movements could further enhance safety and efficiency in human-robot collaborative warehouse settings. These advancements will contribute to the development of more intelligent, adaptive, and energy-efficient autonomous logistics systems.

References

[1] Azadeh, K., de Koster, R., & Roy, D. (2019). Robotized and automated warehouse systems: Review and recent developments. Transportation Science, 53(4), 967-999.

[2] Le-Anh, T., & De Koster, M. B. M. (2006). A review of design and control of automated guided vehicle systems. European Journal of Operational Research, 171(1), 1-23.

[3] 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.

[4] Yap, P., Burch, N., & Holte, R. (2011). Dynamic route planning in video games. In Proceedings of the Seventh Conference on Artificial Intelligence and Interactive Digital Entertainment (pp. 142-147). AAAI Press.

[5] Sturtevant, N. R. (2012). Benchmarks for grid-based pathfinding. IEEE Transactions on Computational Intelligence and AI in Games, 4(2), 144-148.

[6] Liu, B., He, L., & Atkin, J. A. (2018). Multi-objective path planning for automated guided vehicles with energy consumption and travel time trade-offs. International Journal of Production Research, 56(15), 5117-5133.

[7] Van den Berg, J., Lin, M., & Manocha, D. (2008). Reciprocal velocity obstacles for real-time multi-agent navigation. In 2008 IEEE International Conference on Robotics and Automation (pp. 1928-1935).

[8] Phillips, M., & Likhachev, M. (2011). SIPP: Safe interval path planning for dynamic environments. In Proceedings of the IEEE International Conference on Robotics and Automation (pp. 5628-5635).

[9] Kim, J., Zhang, S., & Sun, C. (2017). Trajectory smoothing and velocity planning for autonomous vehicle navigation. IEEE Transactions on Intelligent Transportation Systems, 18(9), 2446-2455.

[10] Rosmann, C., Hoffmann, F., Bertram, T., & Knoth, O. (2017). Trajectory modification considering dynamic constraints of autonomous robots. In ROBOTIK 2017; 10th German Conference on Robotics (pp. 1-8). VDE.

[11] Dehghan, M., & Lee, C. G. (2021). Simulation-based optimization for warehouse robot picking systems. Computers & Industrial Engineering, 160, 107573.

[12] Wang, Y., & Li, X. (2020). Energy-efficient path planning for autonomous mobile robots in manufacturing systems. Journal of Manufacturing Systems, 57, 286-296.