Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 6, December 2026, pp. 5360~5370
ISSN: 2302-9285, DOI: 10.11591/eei.v15i6.13344

Particle swarm optimization based multi objective path planning of automatic guided vehicle

Prema C.[1,2], Madhusudhan K. N.[1]
[1]Department of Electronics and Communication Engineering, BMS College of Engineering (BMSCE), affiliated to Visvesvaraya Technological University, Bengaluru, India
[2]Department of Artificial Intelligence and Machine Learning, Sri Krishna Institute of Technology (SKIT), affiliated to Visvesvaraya Technological University, Bengaluru, India

Article Info

Article history:

Received May 13, 2026
Revised Sep 16, 2026
Accepted Oct 4, 2026

Keywords:

Automated guided vehicles
Gravitational search algorithm
Multi-objective optimization
Particle swarm optimization
Path planning
Task scheduling

ABSTRACT

Automated guided vehicles (AGVs) are increasingly used in industrial material-handling systems, making efficient path planning essential for reducing operational time, travel distance, and energy consumption. This study proposes a multi-objective particle swarm optimization (PSO) framework for AGV path planning and scheduling that jointly considers makespan, travel distance, and energy consumption while incorporating obstacle avoidance and trajectory smoothness. The novelty of the proposed framework lies in the integration of task allocation and path optimization within a unified PSO-based multi-objective framework, enabling the trade-offs among competing operational objectives. The framework is implemented in Python and evaluated across different AGV operating scenarios. In the scenario where both PSO and the gravitational search algorithm (GSA) were evaluated, PSO achieved 13.03% lower makespan, 13.59% lower energy consumption, and 15.71% lower travel distance than GSA. The results demonstrate the effectiveness of PSO for multi-objective AGV path planning, particularly in balancing travel distance, completion time, and energy consumption under different warehouse layouts. The proposed work mainly used in warehouses, manufacturing industries, and material movement places where the optimization required to reduce the operational cost. The scenario-based evaluation also highlights the importance of selecting optimization strategies according to the characteristics of the industrial environment.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Prema C.
Department of Electronics and Communication Engineering, BMS College of Engineering (BMSCE)
Affiliated to Visvesvaraya Technological University
Bengaluru, India
Email: premaash7@gmail.com

1. INTRODUCTION

Research on automated guided vehicles (AGVs) has grown rapidly with the increasing use of autonomous material-handling systems in industrial environments. Path planning and scheduling are important operations to optimize the performance of AGVs. In multi-AGV systems, these decisions are interdependent because routing affects travel time, energy consumption, conflicts, and overall scheduling performance.

AGV path planning focused on Dijkstra's and A* algorithms in static environment due to their efficiency. But these algorithms are not suitable for the complicated environment of AGV and proposing the hybrid approaches [1], [2]. Metaheuristic algorithms are proposed when multiple optimization parameters are

considered for AGV routing and scheduling operations [3]–[11]. In dynamic environment, reinforcement learning approaches are suggested for the multi-AGV routing and task assignment [12]–[18].

Studies on AGV scheduling recently is focused on unified approach for path planning and scheduling. To accomplish this by incorporating the various objectives like total completion time, energy consumption, obstacle avoidance, and path smoothness [10], [11], [19]–[24]. This illustrates the demand for multi objective optimization of AGV in highly constrained environment. Recent studies have also shown that combining reinforcement learning with robotic frameworks helps multiple agents coordinate better and avoid trajectory errors in complex environments [25].

Multi objective optimization in AGV is achieved through a particle swarm optimization (PSO) algorithm, which is a population based metaheuristic algorithm used to find the global optimum with constrained objectives. Therefore, this paper focuses on multi objective optimization in AGV for task allocation and scheduling using PSO with different configurations. The multiple objectives considered for the experiments are makespan, travel distance, energy consumption, obstacle avoidance, and path smoothness. These experimental results were compared with the gravitational search algorithm (GSA).

2. PARTICLE SWARM OPTIMIZATION BASED MULTI-OBJECTIVE PATH PLANNING

This paper considers several constraints including battery capacity, maximum payload, grid boundaries, minimum distance between AGVs and additionally each task must be assigned to one AGV only. To validate the throughput elasticity of the PSO algorithm, analyses performed with different warehouse configurations by changing fleet size, number of tasks, number of AGV's, and also changed the PSO parameters like number of particles, inertia weight, acceleration coefficient, and number of iterations.

The optimization framework defines several decision variables for AGV scheduling and path planning. The binary assignment variable $x_{a,t} \in \{0,1\}$ indicates whether task t is assigned to AGV a. A value of 1 denotes assignment, while 0 indicates otherwise. The variable $P_{a,t}$ represents the path planned by the PSO algorithm for AGV a to execute task t. Additionally, each particle in the PSO search space is represented by the vector $\mathrm{x}_i \in \mathbb{R}^{2n_w}$, where n_w denotes the number of waypoints used to construct a candidate route. The PSO algorithm is employed to generate feasible and efficient AGV routes between task locations. Each particle encodes a candidate path from the start position to the destination using waypoint coordinates. The particle dimension is defined in (1):

$$\mathrm{d} = 2 \cdot \mathrm{nwaypoints} \tag{1}$$

The velocity and position of each particle are updated iteratively according to the standard PSO equations is defined as given in (2) and (3), respectively.

$$\mathrm{vi}(\mathrm{t}+1) = \mathrm{w} \cdot \mathrm{vi}(\mathrm{t}) + \mathrm{c1r1}(\mathrm{pi} - \mathrm{xi}(\mathrm{t})) + \mathrm{c2r2}(\mathrm{g} - \mathrm{xi}(\mathrm{t})) \tag{2}$$

$$\mathrm{x_i}(\mathrm{t}+1) = \mathrm{x_i}(\mathrm{t}) + \mathrm{v_i}(\mathrm{t}+1) \tag{3}$$

where w is the inertia weight, c_1 and c_2 are the cognitive and social acceleration coefficients, respectively, and $\mathrm{r}_1, \mathrm{r}_2 \sim \mathrm{U}(0,1)$ are uniformly distributed random variables. In the experiments, the parameter values are fixed as $\mathrm{w} = 0.6$, $\mathrm{c}_1 = 1.8$, and $\mathrm{c}_2 = 1.5$. The quality of each candidate path is evaluated using a composite objective function defined in (4).

$$\mathrm{J_{path}} = \mathrm{L} + \mathrm{P_{obs}} + \mathrm{P_{smooth}} \tag{4}$$

where L represents the total path length defined in (5). Here, p_k and p_{k+1} denote two consecutive points on the path, and m represents the total number of path segments.

$$L = \sum_{k=0}^{m-1} \| p_{k+1} - p_k \| \tag{5}$$

The obstacle penalty term is defined as in (6). $\mathrm{P_{obs}}$ defined in (6) represents the obstacle-avoidance penalty associated with the path. $\mathrm{P_{smooth}}$ represents the path-smoothness penalty, where $f_{\mathrm{obs}}(p)$ represents the obstacle penalty associated with a path point. Here, p represents a point or waypoint on the AGV path and $\mathrm{dist_{obs}}(p)$ represents the minimum distance between point and the nearest obstacle. A penalty of 1000 is assigned when is an invalid point, such as a point located within an obstacle. For a valid point, the penalty is calculated based on its proximity to the nearest obstacle. The penalty increases as the AGV path approaches an obstacle and becomes zero when the point is sufficiently far from the obstacle.

$$P_{\text{obs}} = \sum_{k=0}^{m} f_{\text{obs}}(p_k) \quad (6)$$

$$f_{\text{obs}}(p) = \begin{cases} 1000, & \text{if } p \text{ is invalid} \\ \max(0,2 - \text{dist}_{\text{obs}}(p)) \cdot 50, & \text{otherwise} \end{cases}$$

The smoothness penalty is computed as given in (7), where θ_k denotes the turning angle at waypoint k. This formulation encourages shorter, collision-free, and smoother paths. Smoothness penalties matter in AGV path planning because a geometrically short path is not necessarily a good path for an AGV to physically execute. A path with many sharp turns can increase steering effort, reduce stability, and make the vehicle harder to control.

$$P_{\text{smooth}} = \sum_{k=1}^{m-1} 10 \cdot \theta_k \quad (7)$$

The overall scheduling objective function of the proposed implementation for each candidate AGV-task assignment is defined in (8); the scheduler evaluates the composite scheduling cost:

$$C_{\text{task}} = 0.4 \cdot D + 0.4 \cdot \frac{E}{100} + 0.2 \cdot T \quad (8)$$

where $D = D_{\text{pickup}} + D_{\text{dropoff}}$, $E = E_{\text{pickup}} + E_{\text{dropoff}}$, and $T = t_{\text{current}} + D$.
The experimental framework records three primary performance metrics:

$$\text{Makespan} = \max_a T_a$$

$$\text{Energy} = \sum_a E_a$$

$$\text{Distance} = \sum_a D_a$$

These metrics are used to assess the efficiency of the scheduling strategy across different operational conditions. Several operational constraints are enforced to ensure feasible AGV scheduling and navigation. Each task can be assigned to at most one AGV as defined in (9), where $x_{a,t}$ is a binary decision variable that takes the value 1 if task t is assigned to AGV a, and 0 otherwise; t denotes the total number of AGVs, and NT denotes the total number of tasks. Battery energy constraints is defined as the total energy consumption of an AGV must remain within its battery capacity as defined in (10).

$$\sum_a x_{a,t} \leq 1 \quad \forall t \in \{1, \ldots, NT\} \quad (9)$$

$$E_a^{\text{current}} + E_{a,t} \leq B_a \forall a, t \quad (10)$$

A valid path $P_{a,t}$ must exist for both pickup and drop-off operations is defined as the path feasibility constraint. If no feasible route is found, the path planner returns None, where T_a represents the current completion time of AGV a, $D_{a,t}$ denotes the time required by AGV a to complete task t, and deadline_t represents the maximum allowable completion time for task t. This constraint ensures that task assignments generated by the proposed scheduling framework satisfy the specified delivery deadlines and prevents infeasible schedules resulting from delayed task completion as given in (11):

$$T_a + D_{a,t} \leq \text{deadline}_t \quad (11)$$

The PSO cost function penalizes invalid points and routes passing close to obstacles, thereby enforcing safe navigation. The scheduler performs collision checking during the evaluation of the objective function in (8), which compares AGV positions at integer time steps and flags simultaneous occupancy of the same grid location. The framework evaluates the proposed methodology under multiple AGV-task configurations, including: (1,2), (2,4), (3,6), and (4,8) where each pair represents the number of AGVs and tasks, respectively. Several execution scripts are used for experimentation. The experimental environment uses a 40×40 grid with static obstacles generated through the function. The starting point of the swarm is at the perimeter of the grid and tasks are generated using seed-based randomness to ensure reproducibility. The population size and number of iterations are varied to observe the influence of tuning parameter in PSO.

Initialized with $n_{\text{particles}} = 20, n_{\text{iterations}} = 50$ and for the fast evaluation runs consider the reduced configurations: $n_{\text{particles}} \in \{2,10\}$ and $n_{\text{iterations}} \in \{5,20,30\}$.

Investigating with different configurations ensures the scalability of the algorithm, trade-off between convergence quality and computational time. Pathlength of each AGV is calculated by the number of waypoints the particular AGV passed and each way point represents one grid length. Greedy task assignment approach is employed to assign task to each AGV. This optimization framework minimizes a path cost function by considering distance, obstacle avoidance, and smoothness, while the scheduler optimizes a weighted combination of travel distance, energy consumption, and elapsed time. Operational constraints such as battery limits, path feasibility, deadline compliance, and conflict avoidance ensure realistic AGV operation.

3. RESULTS AND DISCUSSION

A multi-objective solution on AGV path planning is developed considering scheduling efficiency, path quality, scalability, and computational behavior. The summary of results obtained from different scenarios is tabulated in Table 1 for a 40×40 workspace environment without changing the number of particles and iterations.

Table 1. Experimental scenario configurations

Scenario ID	Number of AGVs	Number of tasks	Grid size	PSO particles	Iterations
S1	1	2	40×40	20	50
S2	2	4	40×40	20	50
S3	3	6	40×40	20	50
S4	4	8	40×40	20	50

PSO's ability to generate collision free paths in the considered workspace environment is evident in the experiment thus developed. The components for the path planning objective function are defined in Table 2. Path length is calculated by using Euclidean distance it has to be minimized, while the obstacle penalty is calculated in terms of how close the AGV is there with another AGV which also should be minimized. AGV should be turning smoothly to avoid higher energy usage which is also penalized using smoothness penalty.

Table 2. Path planning objective components

Objective component	Description	Optimization goal
Path length (L)	Total Euclidean distance	Minimize
Obstacle penalty (P_{obs})	Penalty for obstacle proximity/collision	Minimize
Smoothness penalty (P_{smooth})	Turning angle penalty	Minimize
Total cost (J_{path})	Combined path cost	Minimize

Similar to path planning objective parameters scheduling objective parameters is detailed in Table 3. Total AGV travel distance is defined as D and given the weightage of 0.4, while energy consumption (E) and elapsed time (T) is given a weightage of 0.2.

Table 3. Scheduling objective parameters

Parameter	Description	Weight
(D)	Total travel distance	0.4
(E)	Energy consumption	0.4
(T)	Elapsed completion time	0.2

The trend observed in the experiment thus carried out is as given in Table 4. For different number of AGVs, and different number of tasks assigned to those AGVs are as given in Table 4. The path planning obtained by applying PSO multi-objective optimization thus defined is as shown in Figure 1. Average energy used and distance covered for the tasks assigned using the PSO algorithm is given in Figure 2. Average AGV completion time and the tasks assigned is as given in Figure 3.

Table 4. Observed experimental trends

Scenario	Observed behavior
1 AGV–2 tasks	Stable scheduling with minimal conflicts and low path cost
2 AGVs–4 tasks	Improved workload distribution and reduced completion delay
3 AGVs–6 tasks	Increased scheduling complexity with moderate route interactions
4 AGVs–8 tasks	Higher computational load but improved parallel task execution

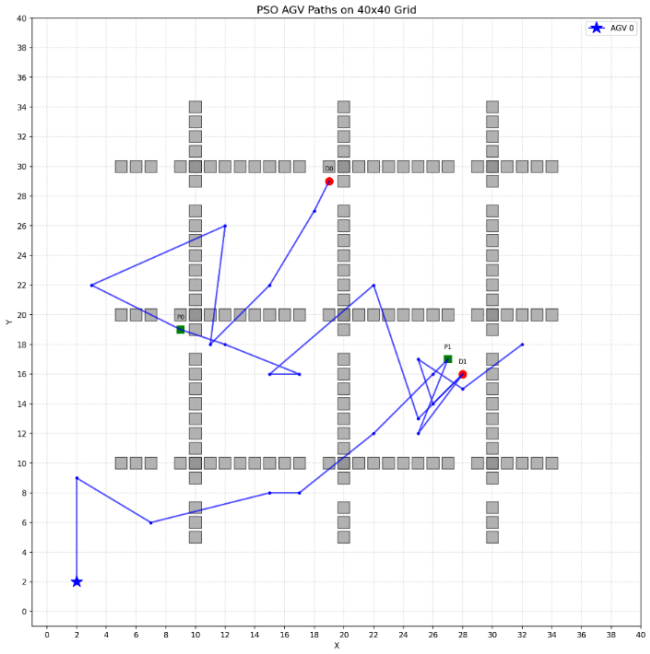


Figure 1. Optimized AGV path with PSO implementation

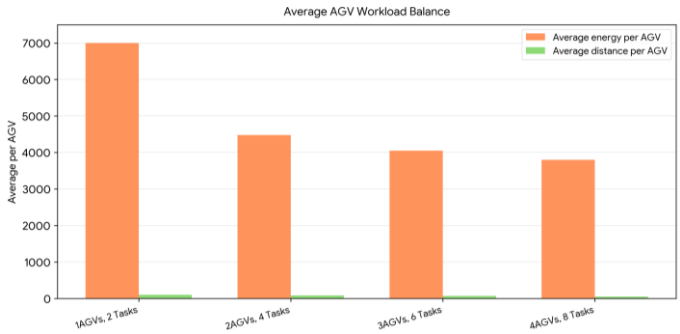


Figure 2. Energy and distance per AGV

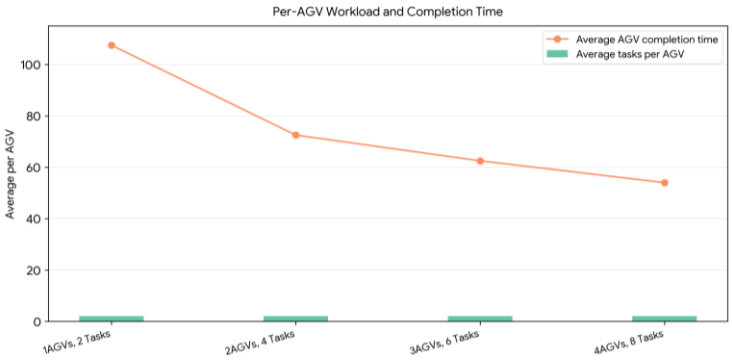


Figure 3. AGV completion time and tasks assigned

Makespan, total energy consumption, total distance travelled, overall average combined energy and cum time per task is shown in the Figure 4 for different number of AGV and tasks scenario. It can be seen that makespan, total energy and total distance are found to be increasing with the number of tasks. Apart from the objective function variables, constraint evaluation is also checked for the implementation with all the constraints working satisfactorily as shown in Table 5.

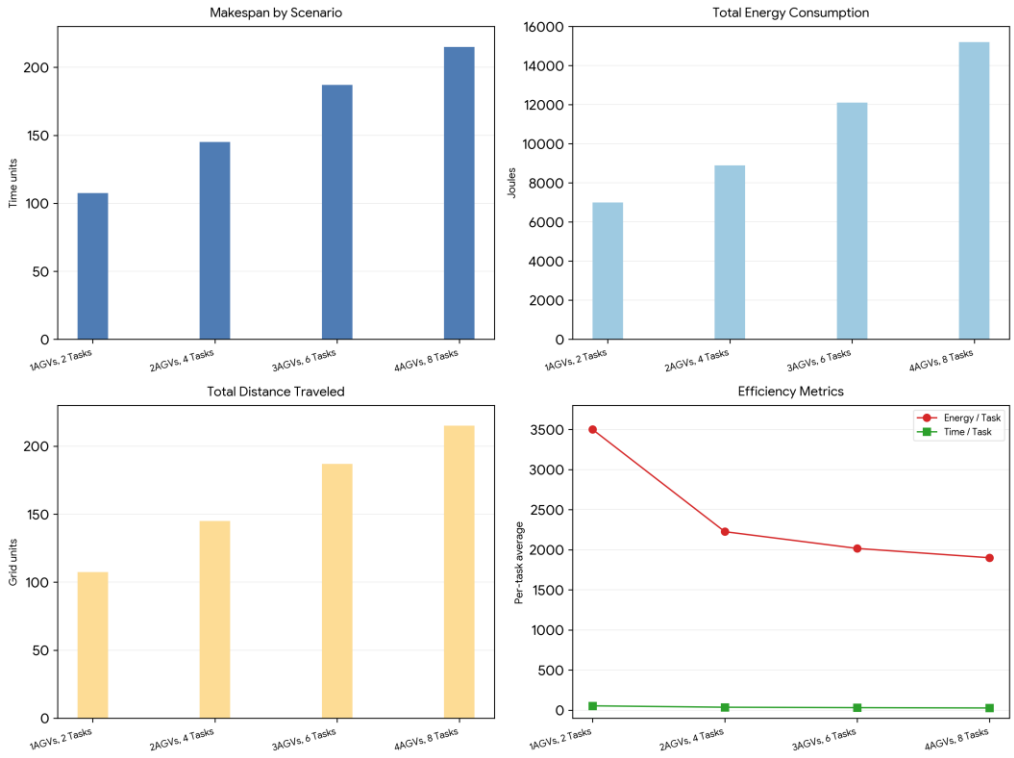


Figure 4. Optimized objective variables

Table 5. Constraint evaluation summary

Constraint	Experimental observation
Single task assignment	Satisfied in all scenarios
Battery capacity	No battery overflow observed
Path feasibility	Invalid paths rejected successfully
Deadline constraints	Maintained when deadlines defined
Obstacle avoidance	Collision-free paths generated
Conflict detection	Same-position occupancy identified correctly

The robustness of any optimization algorithm has to be checked since we need to confirm whether the algorithm works is different number of parameters, thus the observation for different optimization parameters like number of particles and maximum iterations is checked for stable operation of optimization algorithm. The parameter sensitivity is evident in the observation tabulated in Table 6.

Table 6. PSO parameter sensitivity

Experiment type	Particles	Iterations	Observation
Fast validation	2	5	Very fast execution but lower path quality
Reduced swarm	10	20	Moderate convergence and acceptable runtime
Standard configuration	20	50	Best overall optimization performance

The results indicate that larger swarm populations and higher iteration counts improve path quality and scheduling stability but increase computational cost. The multi-objective scheduling using PSO effectively minimized operational costs by balancing distance, energy consumption, and completion time. The results

further show that the framework scales reasonably well as the number of AGVs and tasks increases. PSO-based experimentation showed strong robustness, reproducibility, and scalability for AGV scheduling applications. In order to validate the optimization implementation, the implementation is also validated using the GSA. For different warehouse scenarios as given in Table 7, the comparison of the multi-objective optimization is carried out using both PSO and GSA.

The scenario-specific plots were used to examine changes in optimization behavior rather than to assume that performance in one layout necessarily generalizes to all warehouse configurations. In particular, the cross dock scenario demonstrates that changes in layout can alter the balance between makespan and energy, while the long-aisle scenario highlights the influence of path geometry on distance and completion time. The convergence curves for all the different scenario is as given the Figures 5 to 7.

Table 7. Warehouse scenarios used for PSO and GSA evaluation

Scenario	Warehouse configuration	Main characteristic	Purpose
S1	Default 40×40	Standard grid layout	Baseline evaluation
S2	Crossdock 60×60	Larger cross-dock geometry	Evaluate scalability and layout effects
S3	Long aisles 60×40	Elongated aisle structure	Evaluate topology sensitivity

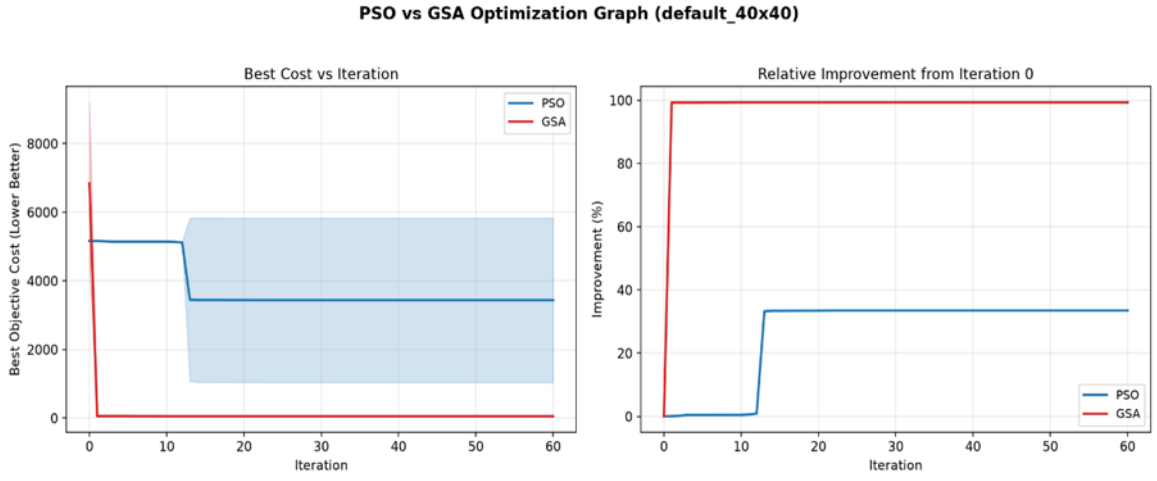


Figure 5. Default 40×40 optimization graph

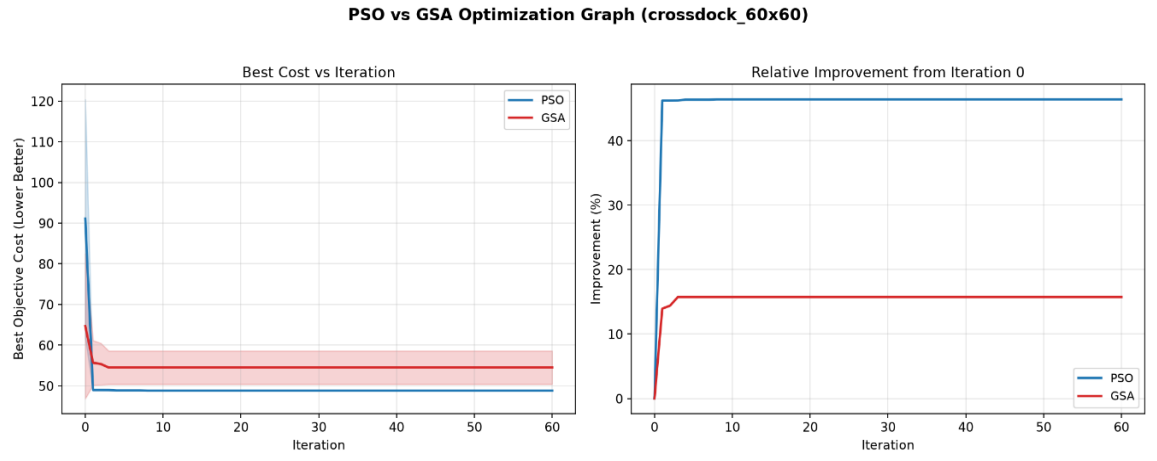


Figure 6. Grid configuration 60×60 optimization graph

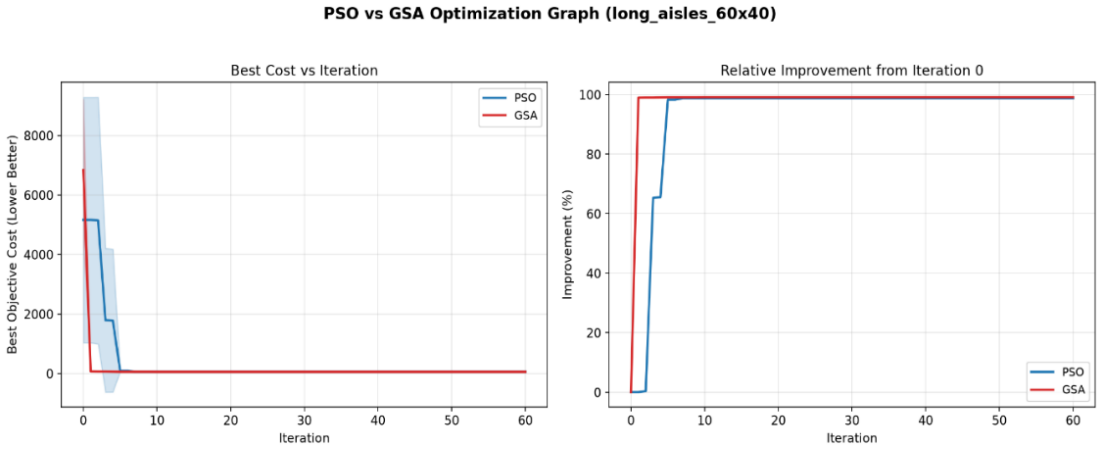


Figure 7. Grid configuration 60×40 optimization graph

The difference is particularly evident in travel distance. PSO generated a route of 256.98 units, whereas GSA generated 304.88 units as can be seen in Table 8, representing approximately a 15.7% lower travel distance for PSO in the common default scenario. PSO also reduced makespan by approximately 13.0% and energy consumption by approximately 13.6% relative to GSA in this scenario (Tables 8 and 9). These comparisons are calculated directly from the reported values and indicate that PSO provided the more effective metaheuristic solution for the tested default environment. Since the PSO algorithm performed better in the comparison in all aspects, the PSO algorithm is further checked for the robustness with all the different scenarios as given in Table 7. Overall, the experimental implementation demonstrates that PSO provides a promising metaheuristic approach for offline multi-objective AGV path planning, particularly when route quality and trade-offs between makespan, energy, and distance are important. Across the reported scenarios, PSO repeatedly achieved favorable makespan and distance values, although this was accompanied by substantially higher computational requirements. Since the obstacles considered are static in nature the offline path planning is considered in the implementation.

The repeated-run (Table 10) experiments demonstrate that PSO consistently outperforms GSA in terms of mean runtime, makespan, energy consumption, and travel distance across all evaluated configurations. For the first configuration, PSO reduces runtime, makespan, energy consumption, and distance by approximately 51.9%, 8.9%, 4.5%, and 5.0%, respectively, compared with GSA. In the second configuration, the corresponding reductions are 62.4%, 16.9%, 23.5%, and 28.8%, while in the third configuration they are 66.3%, 15.6%, 15.6%, and 19.7%. These results indicate that PSO provides better overall optimization performance than GSA for the evaluated AGV planning problems.

Table 8. PSO–GSA comparison in the common scenario

Performance metric	PSO	GSA	Better algorithm
Runtime (s)	228.39	333.89	PSO
Makespan	113.49	130.49	PSO
Energy (J)	20,994.68	24,296.76	PSO
Distance	256.98	304.88	PSO

Table 9. Scenario-wise PSO trend

Scenario	Makespan	Energy (J)	Distance	Runtime (s)
Default	113.49	20,994.68	256.98	228.39
Long aisles	173.46	21,743.38	308.4	469.53
Crossdock	181.14	40,050.69	492.52	611.57

Table 10. PSO and GSA performance results

Algorithm	Repeats	Mean runtime secs	Runtime STD	Mean makespan	Makespan STD	Mean energy	Energy STD	Mean distance	Distance STD
PSO	3	222.02	19.2	111.21	3.754	21688.2	504.3	268.78	11.45
GSA	3	462.37	26.58	122.11	5.9481	22701.4	1129	283.04	15.453
PSO	3	32.776	2.295	106.12	14.947	18210.7	997.9	220.26	18.383
GSA	3	87.201	3.064	127.68	4.9747	23797.3	781.53	309.22	11.907
PSO	5	28.953	2.22	110.55	12.99	20004.5	2454.8	244.57	33.974
GSA	5	86.038	4.228	131.03	7.554	23702.9	616.9	304.57	11.455

4. CONCLUSION

This paper carried out a Python-based multi-objective path planning and scheduling optimization using PSO of AGVs in different warehouse environments. With makespan, energy, distance as the optimization variables it is implemented with constrained task allocation, waypoint-based path optimization, obstacle avoidance, and conflict detection constraints. In order to test the robustness and validate the experimental different warehouse scenario is chosen and also compared with GSA. For the warehouse with 40×40 grid size PSO achieved a makespan of 113.49, energy consumption of 20,994.68 J, and travel distance of 256.98 units, with a computational time of 228.39 s. For warehouse (60×40) grid size, PSO obtained a makespan of 173.46, energy consumption of 21,743.38 J, and distance of 308.40 units, while for (60×60) grid size resulted in a makespan of 181.14, energy consumption of 40,050.69 J, and distance of 492.52 units. These results indicate that PSO can maintain feasible and relatively efficient paths while adapting to different warehouse geometries.

A comparison with the implemented GSA approach in the common default (40×40) scenario further illustrates the effectiveness of PSO. PSO produced a makespan of 113.49, energy consumption of 20,994.68 J, and distance of 256.98 units, compared with 130.49, 24,296.76 J, and 304.88 units, respectively, for GSA. Thus, in this scenario, PSO reduced makespan by approximately 13.0%, energy consumption by 13.6%, and travel distance by 15.7% relative to GSA. However, GSA was evaluated in only one of the three fully reported comparison scenarios; therefore, these results should not be interpreted as sufficient evidence for generalizing the superiority of PSO over GSA across all warehouse configurations.

Overall, the findings demonstrate that the proposed PSO framework provides a flexible and reproducible approach for offline AGV path optimization, particularly when makespan and travel distance are important objectives. However, the relatively high computational time indicates a limitation for direct real-time deployment. Future work should therefore investigate adaptive PSO parameter tuning, hybrid PSO-based optimization, dynamic obstacle handling, comprehensive GSA evaluation across all scenarios, and real-time multi-AGV coordination to improve computational efficiency, scalability, and applicability to dynamic industrial warehouse environments. The limitation of using static obstacles and offline optimization has been explicitly discussed. Future extensions include dynamic obstacle handling, real-time task scheduling, adaptive parameter tuning, and multi-AGV coordination.

ACKNOWLEDGMENTS

The authors express their gratitude to BMS College of Engineering (BMSCE) for providing the necessary facilities and institutional resources during the preparation of this manuscript.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Prema C.	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				
Madhusudhan K. N.		✓		✓		✓			✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] J. Zhang, X. Yang, W. Wang, J. Guan, L. Ding, and V. C. S. Lee, "Automated guided vehicles and autonomous mobile robots for recognition and tracking in civil engineering," *Automation in Construction*, vol. 146, p. 104699, Feb. 2023, doi: 10.1016/j.autcon.2022.104699.

[2] N. Liu, Z. Hu, M. Wei, P. Guo, S. Zhang, and A. Zhang, "Improved A* algorithm incorporating RRT* thought: A path planning algorithm for AGV in digitalised workshops," *Computers & Operations Research*, vol. 177, p. 106993, May 2025, doi: 10.1016/j.cor.2025.106993.

[3] S. Lin, J. Wang, and X. Kong, "Bio-Inspired Reactive Approaches for Automated Guided Vehicle Path Planning: A Review," *Biomimetics*, vol. 11, no. 1, p. 17, Dec. 2025, doi: 10.3390/biomimetics11010017.

[4] S. Chen, Q. Zhang, G.-C. He, and Y.-B. Wu, "Multiple-AGV Path Planning Method for Two Industrial Links," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 36, no. 12, Sep. 2022, doi: 10.1142/S0218001422590297.

[5] X. Sheng, J. Yang, L. You, J. Li, and R. Wang, "GOA-ACO: A goose optimized ant colony algorithm for the automated guided vehicle path planning," *Alexandria Engineering Journal*, vol. 130, pp. 724–737, Oct. 2025, doi: 10.1016/j.aej.2025.09.036.

[6] H. Yu, L. Liang, Y. Zhu, W. Diao, and G. Li, "Energy consumption and trajectory smoothness co-optimization for automated guided vehicle path planning in industrial logistics," *Evolutionary Intelligence*, vol. 19, no. 2, p. 44, Apr. 2026, doi: 10.1007/s12065-026-01155-w.

[7] S. Zhou, Z. Wang, Y. Li, and X. Wen, "A matheuristic method for the automated guided vehicle scheduling problem with flexible charging and job release," *Applied Soft Computing*, vol. 182, p. 113531, Oct. 2025, doi: 10.1016/j.asoc.2025.113531.

[8] H. Chi, H.-Y. Sang, B. Zhang, P. Duan, and W.-Q. Zou, "BDE-Jaya: A binary discrete enhanced Jaya algorithm for multiple automated guided vehicle scheduling problem in matrix manufacturing workshop," *Swarm and Evolutionary Computation*, vol. 89, p. 101651, Aug. 2024, doi: 10.1016/j.swevo.2024.101651.

[9] Z. Xuanying, N. Yida, H. Yizhi, W. Jiongqi, and H. Zhangming, "A two-stage self-learning Jaya algorithm for the unmanned aerial vehicle scheduling problem under conflicting battlefield cargoes features," *Expert Systems with Applications*, vol. 299, p. 130007, Mar. 2026, doi: 10.1016/j.eswa.2025.130007.

[10] X. Han, K. Li, and P. N. R. Kumar, "A branch-and-price algorithm for task allocation and global path planning of multiple AGVs in intelligent warehouses," *Transportation Research Part E: Logistics and Transportation Review*, vol. 206, p. 104587, Feb. 2026, doi: 10.1016/j.tre.2025.104587.

[11] F. Chen, L. Li, and J. Tang, "Multi-agent task assignment and path finding for heterogeneous AGV systems," *Expert Systems with Applications*, vol. 307, p. 130973, Apr. 2026, doi: 10.1016/j.eswa.2025.130973.

[12] Y.-J. Wang, R. Hu, B. Qian, K. Li, W.-B. Zhang, and J.-B. Yang, "Collaborative deep reinforcement learning algorithm for solving multi-AGV dynamic scheduling problem," *Expert Systems with Applications*, vol. 319, p. 131941, Jul. 2026, doi: 10.1016/j.eswa.2026.131941.

[13] S. Al Mahmud, A. M. Ibrahim, and M. R. Khan, "A comprehensive review on multi robot task assignment and path planning: Operational constraints, challenges, and potentials of deep reinforcement learning," *Applied Soft Computing*, vol. 188, p. 114498, Feb. 2026, doi: 10.1016/j.asoc.2025.114498.

[14] X. Chen, S. Liu, J. Zhao, H. Wu, J. Xian, and J. Montewka, "Autonomous port management based AGV path planning and optimization via an ensemble reinforcement learning framework," *Ocean & Coastal Management*, vol. 251, p. 107087, May 2024, doi: 10.1016/j.ocecoaman.2024.107087.

[15] S. Li, L. Fan, and S. Jia, "A hierarchical solution framework for dynamic and conflict-free AGV scheduling in an automated container terminal," *Transportation Research Part C: Emerging Technologies*, vol. 165, p. 104724, Aug. 2024, doi: 10.1016/j.trc.2024.104724.

[16] X. Zhang, Z. Hong, H. Xi, and J. Li, "Optimizing multiple equipment scheduling for U-shaped automated container terminals considering loading and unloading operations," *Computer-Aided Civil and Infrastructure Engineering*, vol. 39, no. 20, pp. 3103–3124, Oct. 2024, doi: 10.1111/mice.13275.

[17] Y. Yang *et al.*, "Integrating multi-equipment scheduling with accurate AGV path planning for U-shaped automated container terminals," *Computers & Industrial Engineering*, vol. 209, p. 111427, Nov. 2025, doi: 10.1016/j.cie.2025.111427.

[18] T. Zuo, H. Liu, S. Yang, W. Wang, Y. Peng, and R. Wang, "A Reinforcement Learning Method for Automated Guided Vehicle Dispatching and Path Planning Considering Charging and Path Conflicts at an Automated Container Terminal," *Journal of Marine Science and Engineering*, vol. 14, no. 1, p. 55, Dec. 2025, doi: 10.3390/jmse14010055.

[19] T. Adamo, G. Ghiani, and E. Guerriero, "Real-time schedule adjustments for conflict-free vehicle routing," *Computers & Operations Research*, vol. 166, p. 106625, Jun. 2024, doi: 10.1016/j.cor.2024.106625.

[20] F. Gehlhoff, N. Jobs, and V. Henkel, "Agent-Based Control of Interaction Areas in Intralogistics: Concept, Implementation and Simulation," *Logistics*, vol. 9, no. 2, p. 52, Apr. 2025, doi: 10.3390/logistics9020052.

[21] M. Wu, J. Gao, L. Li, and Y. Wang, "Control optimisation of automated guided vehicles in container terminal based on Petri network and dynamic path planning," *Computers and Electrical Engineering*, vol. 104, p. 108471, Dec. 2022, doi: 10.1016/j.compeleceng.2022.108471.

[22] Z. Mao, Z. Xie, K. Fang, D. Huang, and Z. Zhao, "In-plant autonomous mobile robot scheduling and routing problem considering battery consumption model," *Transportation Research Part C: Emerging Technologies*, vol. 183, p. 105488, Feb. 2026, doi: 10.1016/j.trc.2025.105488.

[23] N. T. T. Van, N. M. Tien, N. C. Luong, and H. T. K. Duyen, "Energy Consumption Minimization for Autonomous Mobile Robot: A Convex Approximation Approach," *Journal of Robotics and Control (JRC)*, vol. 4, no. 3, pp. 403–412, Jun. 2023, doi: 10.18196/jrc.v4i3.17509.

[24] W. Tan, C. Chen, M. Lu, L. Wu, L. Chen, and X. Wang, "Energy-aware flexible open shop scheduling with multi-load automated guided vehicle by graph reinforcement learning assisted memetic algorithm," *Expert Systems with Applications*, vol. 314, p. 131650, Jun. 2026, doi: 10.1016/j.eswa.2026.131650.

[25] W. Tan, Q. Lv, X. Li, N. Sun, and N. Huang, "Multi-Agent Continual Deep Reinforcement Learning with Artificial Potential Fields and Switching Control for a Cooperative Robotic Task," *Journal of Intelligent & Robotic Systems*, vol. 112, p. 21, Jan. 2026, doi: 10.1007/s10846-026-02360-6.

BIOGRAPHIES OF AUTHORS



Prema C. received her B.E. degree in Computer Science and Engineering from MS University, Tirunelveli, Tamil Nadu, in 2003. She received her master's degree in Computer Science and Engineering from Anna University, Chennai, Tamil Nadu, in 2012. She is currently pursuing the Ph.D. degree in the Department of Electronics and Communication Engineering at B.M.S. College of Engineering, Bangalore, Karnataka, affiliated with VTU. She has published three papers in international journals, and she has also presented papers at one national and four international conferences. She is a member of professional organizations, including ISTE (LM 113295) and IAENG (218101). She worked as an assistant professor in Department of Computer Science and Engineering at different engineering colleges, and she has 17 years of teaching experience. Her current research interests include real time OS, embedded system scheduling, deadlocks, pattern recognition, machine learning, and automated guided vehicle. She can be contacted at email: premaash7@gmail.com or premdevi_c@yahoo.co.in.



Dr. Madhusudhan K. N. is an assistant professor in the Department of Electronics and Communication Engineering at B.M.S. College of Engineering, Bangalore, Karnataka, India. With five international patents, two national patents, five Science Citation Index (SCI)–indexed journal papers, and six Scopus-indexed journal papers to his name. He earned his Doctor of Philosophy (Ph.D.) degree in Image Security from Anna University, Chennai, in July 2021, with a dissertation on "Contemporary Attack Resilient Hybrid Crypto-Stegano-Model for Secure Healthcare Data Communication" in the Faculty of Information and Communication Technology. He completed his post-graduation in digital communication and networking in April 2006 and his graduation in telecommunication engineering in May 2004. He has also presented five papers at national and international technical conferences. He is a member of several professional organizations, including the Institution of Electronics and Telecommunication Engineering (MIETE-M27195), the Indian Society for Technical Education (MISTE-LM 131072), and the International Association of Engineers (MIAENG-270222). He can be contacted at email: madhusudhankn.ece@bmsce.ac.in.