Journal of Computational Science 67 (2023) 101938

Contents lists available at ScienceDirect

Journal of Computational Science

journal homepage: www.elsevier.com/locate/jocs

An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse

Shiwei Lin [a,*], Ang Liu [a], Jianguo Wang [a], Xiaoying Kong [b]

[a] Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia
[b] School of IT and Engineering, Melbourne Institute of Technology, Sydney Campus, Sydney, NSW, Australia

ARTICLE INFO

Keywords:
Optimization
Meta-heuristic algorithm
Mobile robot path planning

ABSTRACT

Mobile robots play crucial roles in industry and commerce, and automatic guided vehicles (AGV) are one of the primary parts of smart manufactory and intelligent logistics. Path planning is the core task for the AGV system, and it generates the path from origin to destination. The motivation of the study is to improve the scalability, flexibility, adaptability, and performance of the robot path planning systems. We propose the hybrid PSO-SA algorithm for the optimization of AGV path planning. Compared with other heuristic algorithms by benchmark functions, including HS, FA, ABC and GA, the proposed algorithm shows excellent performance in dealing with optimization problems. It reduces the possibility of getting trapped in one local optimum and enhances the efficiency to get the best global solution with faster convergence and less time consumption. It is evaluated with multiple cost functions and path planning with simulations and experiments. The objective of the proposed algorithm is to minimize the path length and produce a smooth path without collision. The proposed PSO-SA algorithm is compared with PSO in the path planning application, and the mean runtime and iteration times are usually significantly lower than PSO.

1. Introduction

Mobile robots have various applications, such as entertainment, cleaning, surveillance, and object delivery [1]. The requirements of the AGV navigation system in flexible manufacturing systems include versatility, flexibility, no human interaction, scalability, performance, adaptability, and robustness [2,3]. Path planning, real-time monitoring, task scheduling, and traffic coordination are the primary consideration for AGV operations [4]. The requirements of AGV path planning for enterprises are feasibility and practicability, which consists of no line or rail navigation, low latency, and cost, remotely controllable, and precise positioning [3,5].

The path planning approaches can be categorized into geometric, grid-based, reward-based, random incremental, and Next Best View [6]. Robot path planning algorithms can also be classified as evolutionary and non-evolutionary algorithms [7]. The common used robot path planning algorithms consist of the artificial potential field [8], the random search ant colony algorithm, the genetic algorithm [9], grid maps [1], the search algorithms [10], particle swarms [11], and reinforcement learning and neural network [12].

Multi-criteria decision-making is proposed in [10] for crowd-based path planning in the unknown environment, using the full consistency method and implementing the D* Lite algorithm. D* algorithm extends the A* algorithm, an incremental graph search algorithm, and D* Lite is algorithmically simpler and different from D* for a partially known environment [10]. EA* is used in path planning, then implementing assignment techniques to inform the robots and fault-detection algorithms to handle robots that fail path planning [13]. Symbiotic navigation for multiple robots is proposed in [1], which enables a knowledge-sharing mechanism with the D* algorithm, and for minimum communication, the map is represented by nodes. A linear temporal logic formula and a weighted transition system for high-level mission specification are presented for automatic path planning for multi-robot [14].

Additionally, machine learning is a technology for implementing human learning abilities, and reinforced learning can make a sequence of decisions for robots to achieve goals in complex and uncertain environments. Model-free approaches are implemented in robot motion planning problems, requiring much training data [15]. Convolution Neural Network is combined with Deep q learning for strengthening the learning algorithm to analyse the situations and information in the images with reward function [12]. In reinforcement learning, the Q-learning algorithm can establish interactive relationships and build a dynamic environment without prior knowledge of the environment.

* Corresponding author.
E-mail address: Shiwei.Lin-1@student.uts.edu.au (S. Lin).

https://doi.org/10.1016/j.jocs.2022.101938
Received 6 May 2022; Received in revised form 20 October 2022; Accepted 27 December 2022
Available online 30 December 2022
1877-7503/

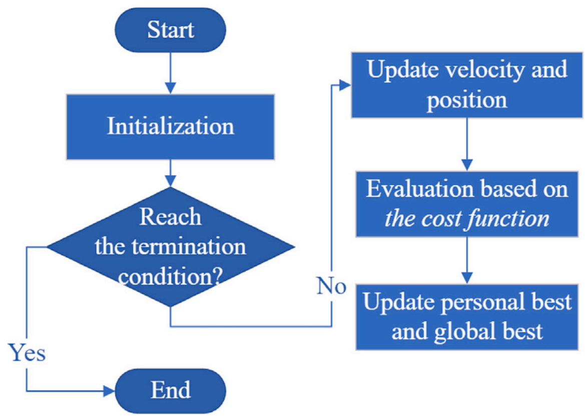


Fig. 1. The flowchart of PSO algorithm.

An empirical playback mechanism is combined with a Q-learning algorithm in Deep Q-network for multi-robot path planning. It solves the problems of excessive randomness, and slow convergence [16]. Reinforcement learning can be integrated with meta-learning to enhance the generalization ability, and transfer learning is involved in the testing process [15].

The complete coverage and path planning (CCPP) techniques can be classified as online and offline approaches. The online method generates the information from the environment, while the offline algorithms are designed for the known environment. Advanced techniques such as cellular decomposition, grid-based, and topological coverage are introduced [17]. [18] proposes a collaborative CCPP approach for unknown environments with maximizing incremental coverage. The Voronoi diagram generates various points as the cores to separate a flat space into multiple areas for robot path planning. The Voronoi diagram is extended to multi-robot path planning and sets a path-priority order for each robot, using the Dijkstra algorithm to generate all navigation points [19].

Evolutionary algorithms are one of the well-researched path planning approaches. They are inspired by the laws of biological survival and natural phenomena, originating from physics and mathematics [7]. Artificial intelligent algorithms convert the path search problem to functional optimization by realizing the path planning of robots with self-renewal and self-learning abilities [7]. Multi-robot path planning requires addressing the validity issues of non-holonomic constants, dynamic changes in the robotic plan, and memory and execution time complexity [9]. Meta-heuristic and heuristic algorithms achieve efficient local and global search by balancing intensification, and diversification [20]. A jumping mechanism particle swarm optimization (PSO) is proposed with a safety gap obstacle avoidance algorithm. This approach uses a fitness function to measure the convergence and then control the update of velocity [7]. The fuzzy inference system and the artificial potential field (APF) are implemented for collision avoidance strategy [8]. A genetic algorithm is modified to handle partitioning and routing sensor-based coverage path planning [21].

The optimization of the evolutionary algorithms has a significant chance of being struck with local optima and getting slow when the dimensionality rises, but the cooperative evolutionary algorithm divides the problem into more minor issues with smaller dimensionality [9]. Evolutionary operators and the improved version of PSO are combined to compute the optimal path for multi-robot [22]. Modified genetic algorithms and improved cooperatively coevolving PSO are introduced in a cooperative path planning approach to address multi-robot persistent coverage [17]. Co-evolutionary grammar-based genetic programming is developed with a maze-like map in [9], and a master evolutionary algorithm achieves overall path optimality. A coevolution-based PSO is presented with evolutionary game theory for a self-adaptive approach, which improves optimization efficiency and guarantees convergence, addressing the stagnation issues and adjusting local and global search abilities [11].

In a continuous environment, the APF is to generate feasible paths based on a time-efficient deterministic scheme and use an enhanced genetic algorithm for modifying the positions for multi-objective multi-robot path planning [23]. The proposed hybridization of kidney-inspired and sine–cosine algorithm chooses subsequent optimal positions for robots, avoiding collisions with other robots and dynamic obstacles [24]. The genetic algorithm is introduced to optimize the goal points, and the boundary node method and path enhancement method are combined to get an optimal collision-free path [25]. An improved PSO is integrated with the gravitational search algorithm inspired by nature, and the proposed co-evolutionary algorithms maintain the balance between exploitation and exploration [26]. A hybridization of improved PSO and differentially perturbed velocity algorithm is proposed in [20], and it aims to minimize the maximum path length and arrival time.

From the literature, path planning algorithms consist of grid-based, reward-based, geometric-based, and evolutionary-based approaches. The grid-based and geometric-based method usually implements graph search or evolutionary algorithms while it wastes the available zones for path planning. The reward-based approach makes a sequence of robot decisions, but it requires enormous computation space, time, and reliable train data. Evolutionary algorithms have robust and straightforward implementations, but they would likely be trapped in local optima or require huge computation space and time for optimization. The co-evolutionary approach can overcome the shortness of each evolutionary algorithm.

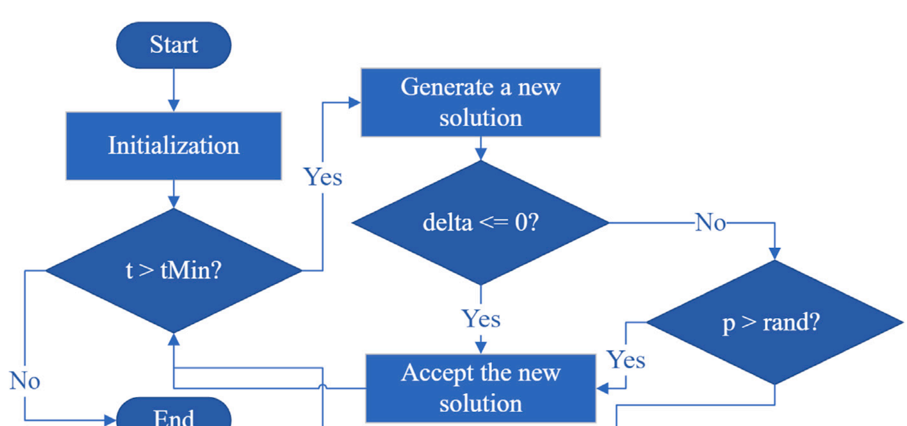


Fig. 2. The flowchart of SA algorithm.

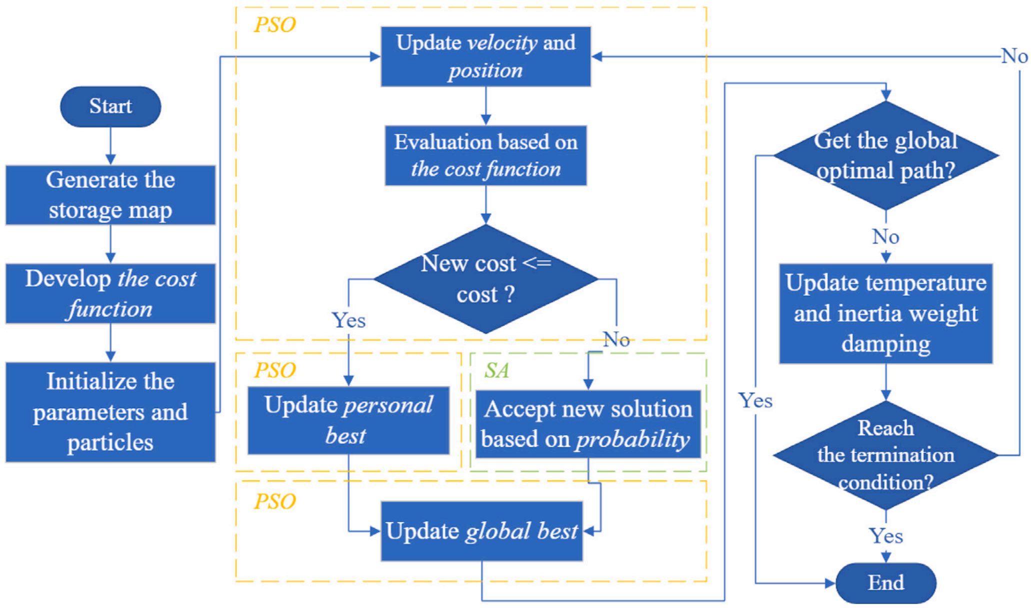


Fig. 3. The flowchart of path planning by Hybrid PSO-SA.

Table 1
Test functions.

Function	Type	Name	Test function
$f_1(x)$	Valley-Shaped	Rosenbrock	$f_1(x) = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
$f_2(x)$	Many Local Minima	Ackley	$f_2(x) = -a\ exp(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}) - exp(\sqrt{\frac{1}{d}\sum_{i=1}^{d} cos(cx_i)}) + a + exp(1)$
$f_3(x)$	Many Local Minima	Levy	$f_3(x) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i - 1)^2[1 + 10sin^2(\pi\omega_i + 1)] + (w_d - 1)^2[1 + sin^2(2\pi\omega_d)], where\ w_i = 1 + \frac{x_i - 1}{4}, for\ all\ i = 1, \dots, d$
$f_4(x)$	Bowl-Shaped	Sphere	$f_4(x) = \sum_{i=1}^{d} x_i^2$
$f_5(x)$	Bowl-Shaped	Sum squares	$f_5(x) = \sum_{i=1}^{d} ix_i^2$
$f_6(x)$	Plate-Shaped	Zakharov	$f_6(x) = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i^2)^2 + (\sum_{i=1}^{d} 0.5ix_i^2)^4$
$f_7(x)$	Steep Ridges/Drops	Michalewicz	$f_7(x) = -\sum_{i=1}^{d} sin(x_i)sin^{2m}(\frac{ix_i^2}{\pi})$

This proposes a new hybrid meta-heuristic algorithm for AGV path planning to explore the optimal global solution with enhanced search abilities. The paper aims to provide an approach for robot path planning and ensure flexibility, scalability, adaptability, performance, and practicability for AGV path planning. It is aimed to reduce the computation space and runtime for generating the path with faster convergence and not getting trapped in the local optima. It also uses coordinators to produce more available zones during path planning. Compared with other well-known evolutionary algorithms, it shows excellent performance in optimization. The approach provides faster convergence and high flexibility in the challenging static environment with the developed cost function for path planning. The proposed approach requires less computation time and iterations to get the optimal global solution in the simulation and experiment. The paper's main contributions include

- proposing a novel hybrid heuristic algorithm that can obtain globally optimal solutions with great performance and compared with other heuristic algorithms
- applying the proposed hybrid PSO-SA algorithm for the path planning application, which has faster convergence and less runtime
- developing cost functions that consider path length, smoothness, and collision avoidance for AGV path planning to provide flexibility, scalability, adaptability, performance, and practicability

The paper provides a robot path planning approach based on the hybrid intelligence algorithm. The report is organized as follows. Section 2 explains the hybrid PSO-SA algorithm. The simulation and experiment results are provided in Section 3 and concluded in Section 4.

2. Hybrid PSO-SA

2.1. Preliminary knowledge

2.1.1. Particle swarm optimization (PSO)

The social behaviour inspires particle swarm optimization (PSO), which is the population-based stochastic optimization approach. As a meta-heuristic optimization approach, it can gain global, or near-global optimum solutions [27]. For PSO, each particle is treated as a potential solution, exploring an optimum within the searching space. PSO has various applications, such as optimization problems [28–31], robot path planning and navigation [20,32–37], and network applications [38].

PSO is a robust optimization algorithm with fast convergence, but it may get trapped in a local optimal in multi-modal problems. The hybrid meta-heuristic approach is introduced to overcome the trapping issue; combining it with another algorithm can enhance searching and exploring abilities to obtain the global optimum solutions. The velocity

Table 2
Mean iteration times and fitness value.

Function		PSO-SA	PSO	SA	HS	FA	ABC	GA
$f_1(x)$	Iterations	**1.00**	**1.00**	200.00	184.75	197.55	131.95	199.10
	Value	**0.00**	**0.00**	−0.21	1.98	0.40	0.77	6.97
$f_2(x)$	Iterations	181.20	187.60	193.20	**179.35**	183.25	196.10	199.30
	Value	0.18	**0.00**	0.05	0.01	**0.00**	**0.00**	**0.00**
$f_3(x)$	Iterations	60.75	**48.60**	199.90	181.35	193.50	196.00	199.40
	Value	0.02	**0.00**	0.83	**0.00**	**0.00**	**0.00**	**0.00**
$f_4(x)$	Iterations	**173.25**	188.50	199.85	185.05	186.95	197.60	199.45
	Value	**0.00**	**0.00**	0.03	**0.00**	**0.00**	**0.00**	**0.00**
$f_5(x)$	Iterations	**180.10**	188.00	199.80	186.20	189.70	197.15	199.45
	Value	**0.00**	**0.00**	0.06	**0.00**	**0.00**	**0.00**	**0.00**
$f_6(x)$	Iterations	**178.15**	187.20	199.80	193.85	189.95	182.15	198.85
	Value	**0.00**	**0.00**	0.05	0.02	**0.00**	**0.00**	0.14
$f_7(x)$	Iterations	1.05	**1.00**	1.30	185.60	192.70	114.70	190.70
	Value	**−0.89**	−0.99	2.19	−4.88	−4.45	−3.47	−9.24

and position for particles of PSO are updated in each iteration as (1) and (2). PSO algorithm is demonstrated in Algorithm 1. The flowchart of PSO algorithm is shown in Fig. 1.

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (pbest_i^t - x_i^t) + c_2 r_2 (gbest^t - x_i^t) \tag{1}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{2}$$

where v_i^{t+1} denotes the velocity for ith particle in the $t+1$ timeslot, and x_i^{t+1} is the position. Consider ω as the weighting factor, r_1 and r_2 are random numbers, and c_1 and c_2 are cognitive parameter and social parameter. $pbest_i^t$ is the particle's best position, and $gbest$ is the global best position for all particles.

Algorithm 1: PSO algorithm

Initialization
for $iteration = 1 : iteration_{max}$ **do**
  for $particle = 1 : particle_{max}$ **do**
    Update velocity and velocity bounds
    Update position
    Evaluation by the cost function
    if $pcost_i^t < pbest_i^t$ **then**
      Update Personal Best
      if $pbest_i^t < gbest^t$ **then**
        Update Global Best
      end
    end
  end
  Update Global Best Cost
end

2.1.2. Simulated Annealing (SA)

Simulated Annealing (SA) algorithm is proposed to reduce the possibility of getting trapped in one local minimum and accepting other solutions. SA is inspired by the annealing process of crystals, which can reach the minimum energy, and if the temperature reduces slower, the energy state will reach lower [13]. SA algorithm is shown in Algorithm 2, and the flowchart of the SA algorithm is indicated in Fig. 2. The probability of accepting a new solution is as (3).

$$\rho = \begin{cases} 1, & newcost \le cost \\ e^{\frac{newcost - cost}{t}} & newcost > cost \end{cases} \tag{3}$$

where $newcost$ is the cost of the new state, and t is the temperature in the SA.

Algorithm 2: SA algorithm

Initialization
while $t > t_{min}$ **do**
  for $particle = 1 : particle_{max}$ **do**
    Generate a new solution
    $delta \leftarrow cost_{new} - cost$
    if $delta <= 0$ **then**
      $particle \leftarrow particle_{new}$
    else
      $p \leftarrow exp(-delta/kt)$
      if $p > rand$ **then**
        $particle \leftarrow particle_{new}$
      end
    end
  end
  $t \leftarrow t * alpha$
end

2.2. Hybrid PSO-SA

2.2.1. Description

Hybrid PSO-SA is proposed for optimization problems, and the application of path planning is developed. The algorithm of the proposed hybrid PSO-SA is in Algorithm 3. The cost function is designed to evaluate the results; initialization is involved. The generated particles have the initial status, then updating velocities and positions for particles based on (1) and (2). The cost is estimated by the cost function for each particle and compared with its personal best. Then the swarm gets the best solutions from the particles as the global best. Gaining the statuses of particles and updating the local-oriented and global-oriented best values are inspired by the PSO algorithm.

PSO only accepts the lower-cost solution, which is likely to get trapped on local optima. SA usually reaches the maximum iteration times to get the solution. However, the proposed PSO-SA may take the new solution even with a higher cost, and it overcomes the shortness of each algorithm. SA algorithm inspires accepting the new solution, while in the proposed algorithm, it updates the local best-oriented value rather than accepting the new solution. The PSO-SA will calculate the probability based on (4) and (5). The new solution is accepted if the probability is larger than a random number between 0 and 1. It provides the possibility to get rid of one local optimum to find the global optimization result.

$$\delta = \frac{pcost_i^t - pbest_i^t}{pbest_i^t} \tag{4}$$

$$\rho = e^{-\frac{\delta}{kt}} \tag{5}$$

where $pcost_i^t$ is the current cost of particle i in the t timeslot, and $pbest_i^t$ is the personal best value. δ in (5) is calculated by (4).

2.2.2. Path planning

The proposed PSO-SA is aimed at path planning, and path planning is formulated as the optimization problem. For the mobile robots' navigation, path planning is the crucial part. We develop the cost function that evaluates path length, collision, and path smoothness.

The path length is one of the primary considerations during path planning in our scenario. The position of one particle is (x, y), and the next position is (x_{k+1}, y_{k+1}). The path consists of n particles, and the cost function of path length is as (6).

$$f_{length} = \sum_{k=1}^{n} \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2} \tag{6}$$

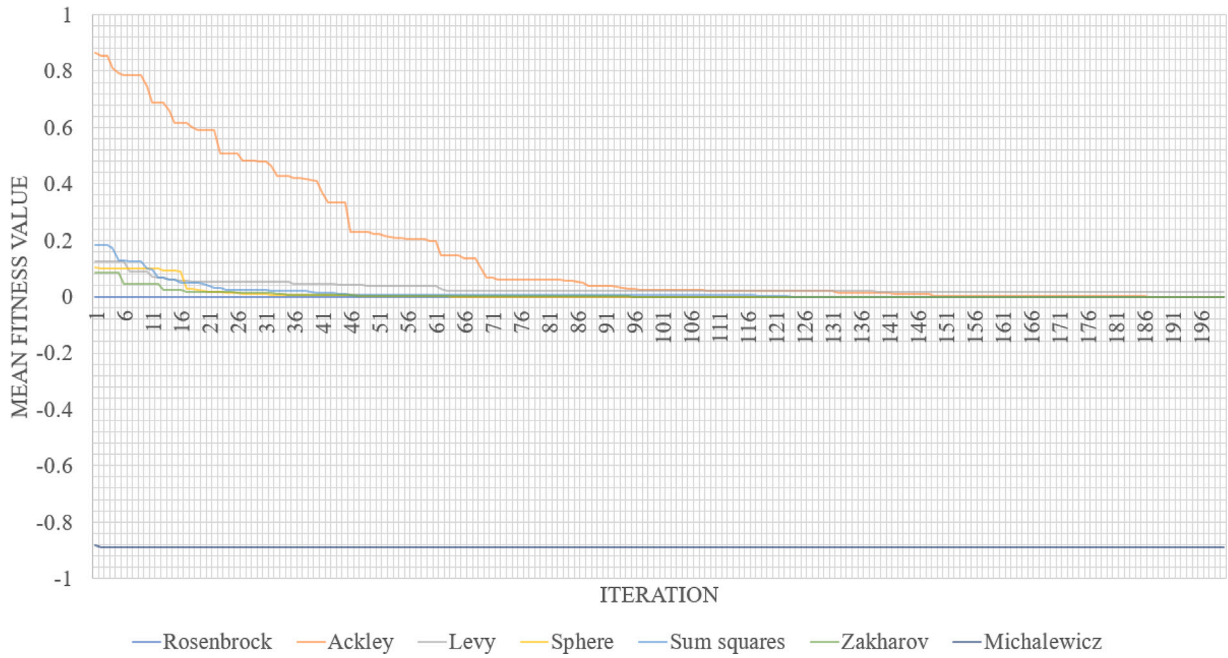


Fig. 4. Convergence curve for PSO-SA.

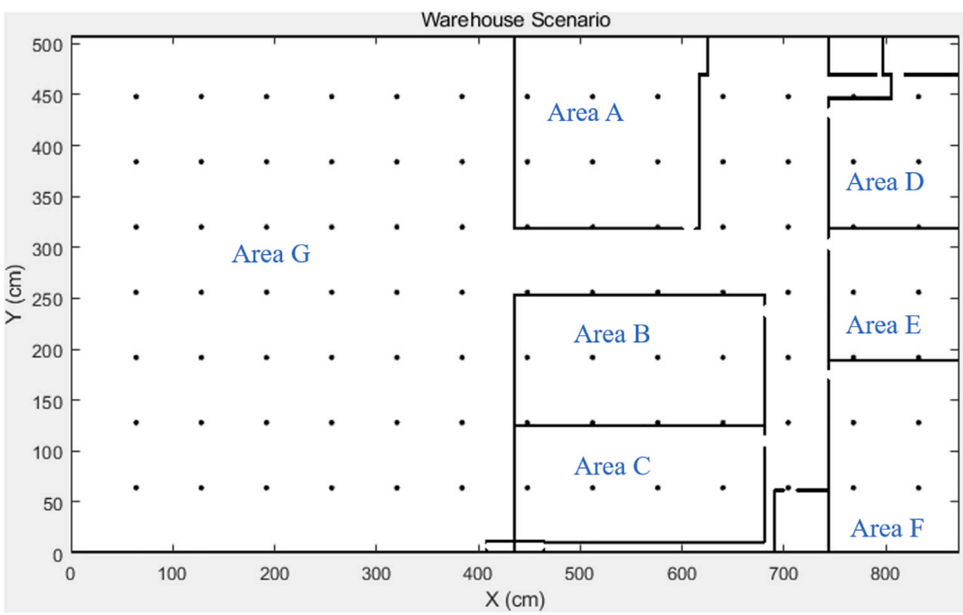


Fig. 5. The scenario of a warehouse.

where n is the number of particles, k is the current particle, and $k+1$ is the following particle.

For robots' operation, a smoother path is easier to follow and consumes less energy. The smoothness of the path is calculated by the slopes from particles. The cost function of smoothness is (7) and (8).

$$s_k = \begin{cases} 0, & if\, x_{k+1} - x_k = 0 \\ \left| \frac{y_{k+1}-y_k}{x_{k+1}-x_k} \right| & else \end{cases} \tag{7}$$

$$f_{smoothness} = \sum_{k=1}^{n} s_k \tag{8}$$

where s_k is the smoothness of the kth particle.

The cost function of collision is evaluated by (9) and (11). The particle k has the collision cost represented by c_k.

$$c_k = \sum_{c=1}^{j} r_c - d_c,\ if\ d_c < r_c \tag{9}$$

$$f_{collision} = \sum_{k=1}^{n} c_k \tag{10}$$

where r_c is the influence radius of the obstacle, and d_c is the distance from the particle to the centre of the obstacle for calculating c_k. j is the number of obstacles. The cost function calculates every collided obstacle with the current particle and then sums it up as c_k.

The cost function considers each factor as (9).

$$f_{cost} = w_1 \cdot f_{length} + w_2 \cdot f_{smoothness} + w_3 \cdot f_{collision} \tag{11}$$

Algorithm 3: Hybrid PSO-SA algorithm

Initialization
for $iteration = 1 : iteration_{max}$ **do**
  for $particle = 1 : particle_{max}$ **do**
    Update velocity and velocity bounds
    Update position
    Evaluation by the cost function
    if $pcost_i^t < pbest_i^t$ **then**
      Update Personal Best
      if $pbest_i^t < gbest^t$ **then**
        Update Global Best
      end
    else
      $delta \leftarrow (pcost_i^t - pbest_i^t)/pbest_i^t$
      $p \leftarrow exp(-delta/kt)$
      if $p > rand$ **then**
        $pbest_i^t.cost \leftarrow pcost_i^t.cost$
      end
    end
  end
  Update Global Best Cost
  Update inertia weight
  $t \leftarrow t * alpha$
end

where w_1, w_2, w_3 are the weight factors for each cost function, and the sum of them is 1.

The scenario is AGV path planning in the warehouse. The flowchart of PSO-SA path planning is shown in the Fig. 3. Using a warehouse map to generate the path, then developing the cost function and initialization for the parameters and the swarm, updating the particles' states, and evaluating the path based on the cost function. The initialization of parameters includes setting the maximum iterations, initial temperature, population size, inertia weight and damping ratio, temperature reduction rate, learning coefficients, and bounds. The complexity of the application determines the weights and parameters, depending on the considered factors.

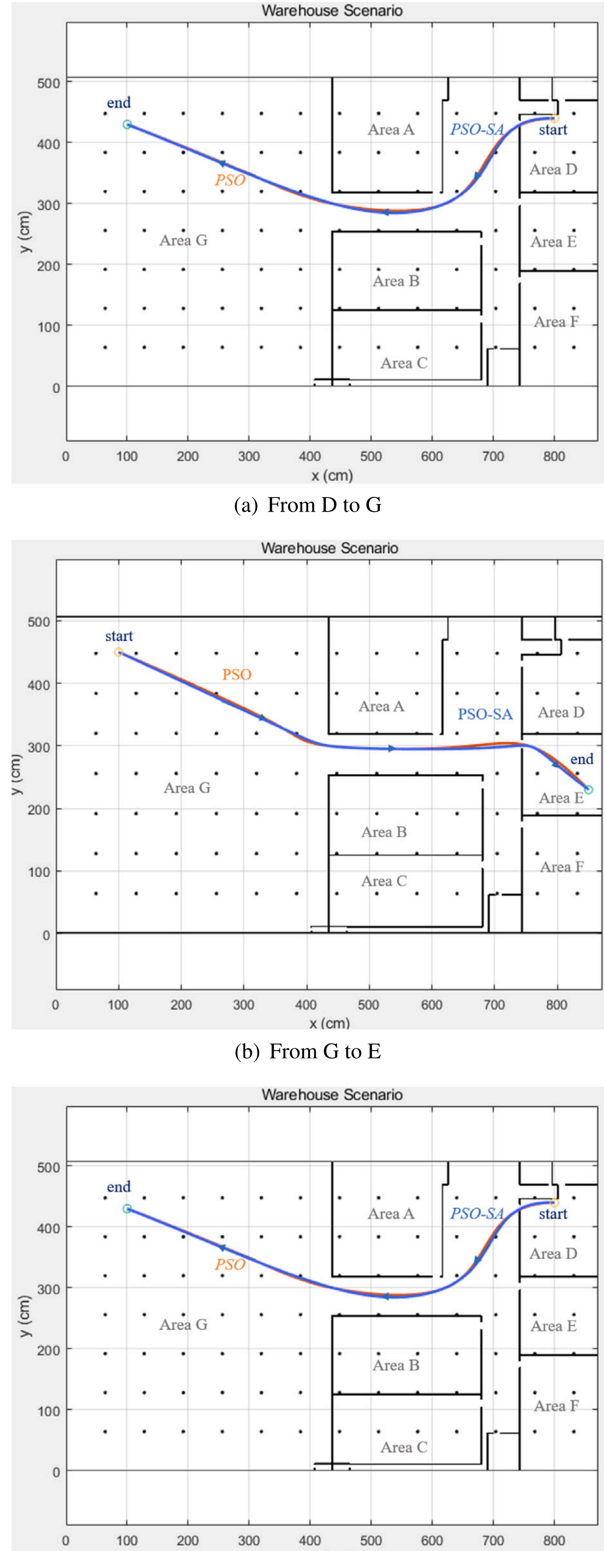
If the new cost is more minor, the personal best of the current particle is updated; otherwise, the probability is involved. If the cost is less than the global best, the cost becomes the global best solution. The optimal global path is determined if the global best cost occurs ten times continuously. The iterations can be terminated in advance once the globally optimal path is determined.

3. Experiment

3.1. Simulation with test functions

The proposed PSO-SA is validated through tests with MATLAB and compared with various optimization algorithms, including PSO, SA, Harmony Search (HS) [39], Firefly Algorithm (FA) [40], Artificial Bee Colony (ABC) [41], and Genetic Algorithm (GA) [42]. The benchmark test functions are the standard optimization problems to test the performance of meta-heuristic algorithms. The test functions are listed in Table 1, and the characters of each function are listed. The graph of valley-shaped, bowl-shaped, plate-shaped, and Steep Ridges/Drops functions are like the valley, bowl, plate, and steep ridges/drops, respectively. The many local minima functions have many local optima. The number of iterations is 200, and each algorithm runs 20 times.

Table 2 compares the mean iteration times for each test function when the algorithms get the best global solution and list the final fitness value for the algorithms. The best mean value for each test function is highlighted. The results show that PSO-SA uses the least mean iteration times to get the best answer in most test functions. For Rosenbrock $f_1(x)$ and Michalewicz function $f_7(x)$, PSO-SA and PSO can easily find the optimal solution.



(a) From D to G

(b) From G to E

(c) From G to F

Fig. 6. The paths for PSO and PSO-SA.

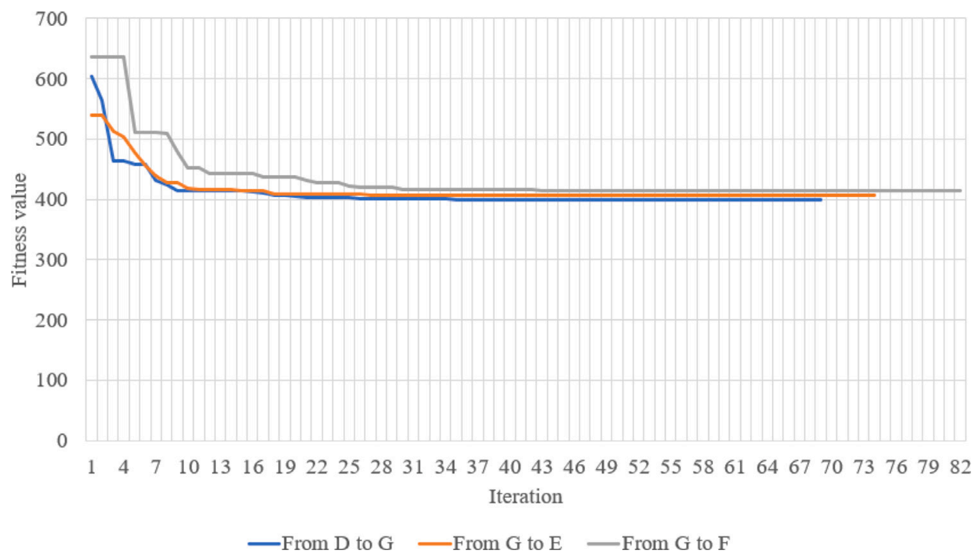


Fig. 7. The convergence curve of PSO-SA path planning.

Table 3
Runtime of each algorithm.

Runtime (ms)		PSO-SA	PSO	SA	HS	FA	ABC	GA
$f_1(x)$	Mean	**3.25**	4.29	4.88	115.15	933.90	641.93	500.12
	Std. dev	**3.05**	3.52	1.67	21.43	28.60	78.29	104.81
$f_2(x)$	Mean	**4.31**	7.61	14.24	116.18	944.02	660.24	465.68
	Std. dev	**2.65**	2.79	12.75	18.03	64.44	39.11	56.01
$f_3(x)$	Mean	**3.99**	6.41	30.08	149.02	1183.89	853.02	603.02
	Std. dev	**2.01**	3.04	4.57	24.94	47.26	45.79	49.97
$f_4(x)$	Mean	4.18	**4.16**	8.43	125.35	925.05	675.87	479.10
	Std. dev	2.59	**2.55**	5.79	21.81	45.72	43.05	51.24
$f_5(x)$	Mean	**4.03**	4.17	6.86	119.92	947.17	652.37	471.05
	Std. dev	2.72	**2.57**	5.79	17.87	50.20	51.32	38.50
$f_6(x)$	Mean	4.57	**4.12**	10.34	115.60	901.46	638.35	478.45
	Std. dev	0.59	**0.35**	1.30	14.09	41.23	21.30	20.71
$f_7(x)$	Mean	**7.08**	11.79	12.33	127.69	981.54	686.64	528.40
	Std. dev	**2.94**	3.42	3.32	22.96	35.61	52.84	52.18

Table 3 compares the run time for the algorithms, with the minimum one highlighted. It can draw that the proposed PSO-SA usually has less mean run time for most functions, except the functions $f_4(x)$ and $f_6(x)$, where PSO has a bit less run time than PSO-SA. The algorithm HS, FA, ABC and GA are very slow compared to PSO, SA and PSO-SA. As mobile robot path planning in real-time, the algorithm's speed is critical for practical application.

Fig. 4 presents the mean fitness value's convergence curve for each test function in 200 iterations. PSO-SA can get the global optimal solution in most cases. The runtime and iteration numbers are also considered. It can conclude that the proposed PSO-SA has an excellent performance in most optimization problems with faster convergence and less consumed time.

3.2. Path planning

3.2.1. Simulation

The environment is generated from an existing warehouse, as shown in Fig. 5, and walls and pillars are exhibited. The AGVs depart from Area G to different storage rooms for operation and return to Area G from storage rooms after completing the tasks. Black dots indicate the pillars, and the walls are annotated by black.

The maximum number of iterations is 150, and the population size is 150. The population size should be within $[100, 200]$ for the path planning problem. For simulation, we tested the population size for the path from Area G to E as the size is 100, 150, and 200. When the population size is 150, PSO-SA has the best performance, with the least runtime for each iteration and fewer iteration times.

The simulation generates the paths from Area G to Area E and Area F and the path from Area D to Area G by PSO and PSO-SA. The start and target position, the best costs, the mean runtime for each iteration, and the iteration times for the paths are shown in Table 4. The cost function (6)–(11) is defined to evaluate the path length, collision, and smoothness. Because the path length and collision are the primary considerations, w_1 is set as 0.5, w_2 is set as 0.4 and w_3 is set as 0.1.

Table 4 shows that PSO-SA performs better in path planning than PSO. Paths for PSO and PSO-SA are shown in Fig. 6, and the convergence curve of PSO-SA is shown in Fig. 7. The areas, the source and target positions, and the directions for the paths are marked. It can draw that the proposed algorithm can get the optimal way with an outstanding performance by Table 4 and Fig. 6. The algorithm terminates when it gets the best cost ten times and records the convergence curve. For the paths, the proposed PSO-SA has less runtime than 30.50%, 51.68%, and 34.43%, respectively. It also achieves path planning with fewer iteration times and better costs than PSO in most situations.

3.2.2. Experiment

The test scenario is shown in Fig. 8. The AGVs are operating at Area A for moving goods to and from the gate of other areas. The three-dimensional storage system is assumed to be utilized in Area B to Area F. The AGVs depart from Area A and return to Area A for parking once they finish the tasks.

The experiment employed a robot carried a Decawave Ultra-Wide Band (UWB) sensor and Raspberry Pi, as shown in Fig. 9. UWB is the positioning sensor that can provide centimetre-level accuracy for the indoor environment. The robot follows the defined path for collecting position data, and the arrow of each path indicates the direction. The robot departs from Area A to the storage room for inbound delivery and moves to Area A for outbound delivery.

There are some obstacles in the simulation to create a more challenging path planning environment. Table 5 lists the start and target positions and the best cost for each path. The maximum number of iterations is set as 150. Fig. 10 exhibits the path and the experiment data in the map, and the simulated path is blue, while the UWB positions are marked yellow. The start position is highlighted in orange, and the target is highlighted in blue. The UWB positions have measurement bias at the centimetre level, and the correction of the bias would be the future work.

Table 4
Performance measurements of the simulated paths.

Path	Start position	End position	Algorithm	Mean runtime (s)	Iteration times	Best Cost
From D to G	(800,440)	(100,430)	PSO-SA	**1.0915**	**69**	400.1210
			PSO	1.5704	129	**398.9587**
From G to E	(100,450)	(850,230)	PSO-SA	**1.0301**	**74**	**406.4598**
			PSO	2.1319	99	407.4978
From G to F	(80,430)	(800,150)	PSO-SA	**1.0922**	**82**	**415.2469**
			PSO	1.6658	97	415.4940

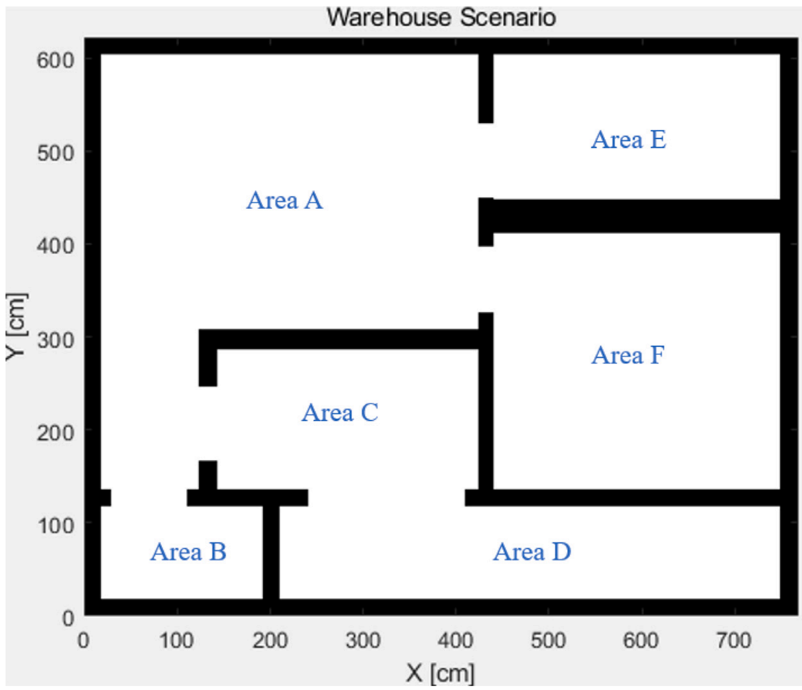


Fig. 8. The scenario of the test side.

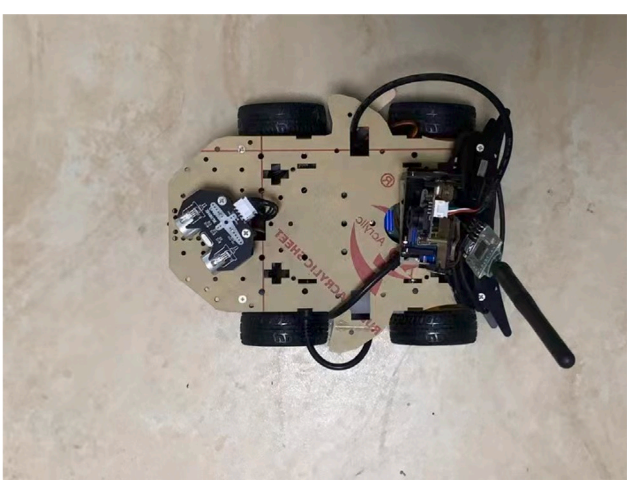


Fig. 9. Robot with UWB.

Table 5
Simulated paths.

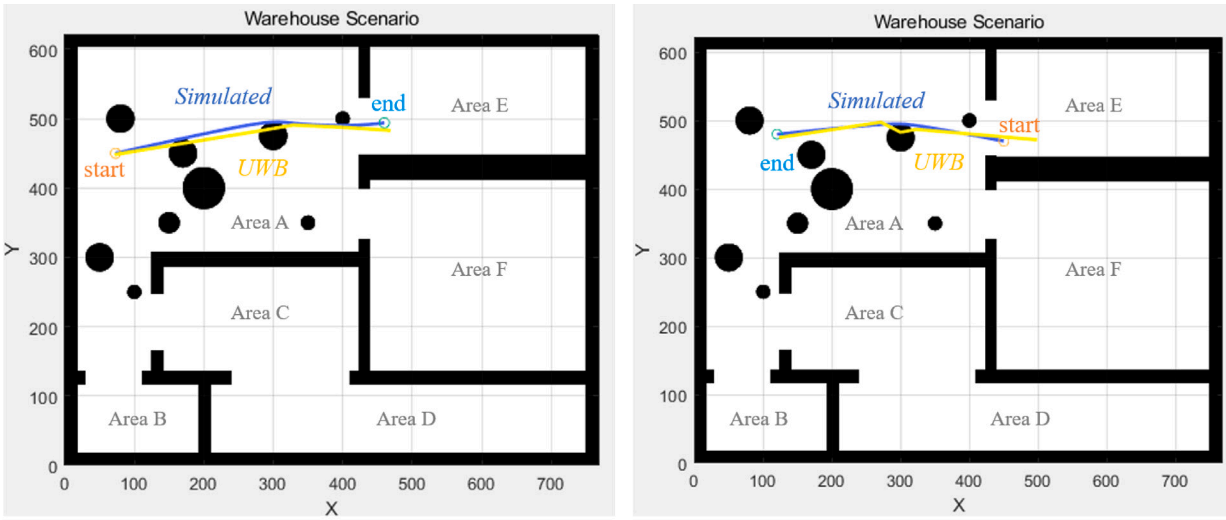
Path	Start	Target	Best cost
From A to E	(73,450)	(460,494)	196.9564
From A to F	(100,400)	(450,350)	179.3558
From F to A	(450,380)	(100,400)	178.5797
From E to A	(450,470)	(120,480)	167.2975

The proposed PSO-SA evaluates the cost for the path in each iteration and reduces the global best to get the optimal path. If the exact best cost of the optimal path occurs ten times continuously, the algorithm will treat the solution as the best solution and then terminate the iterations. In this simulation, the maximum iteration number is around 70, and the minimum iteration number is about 60.

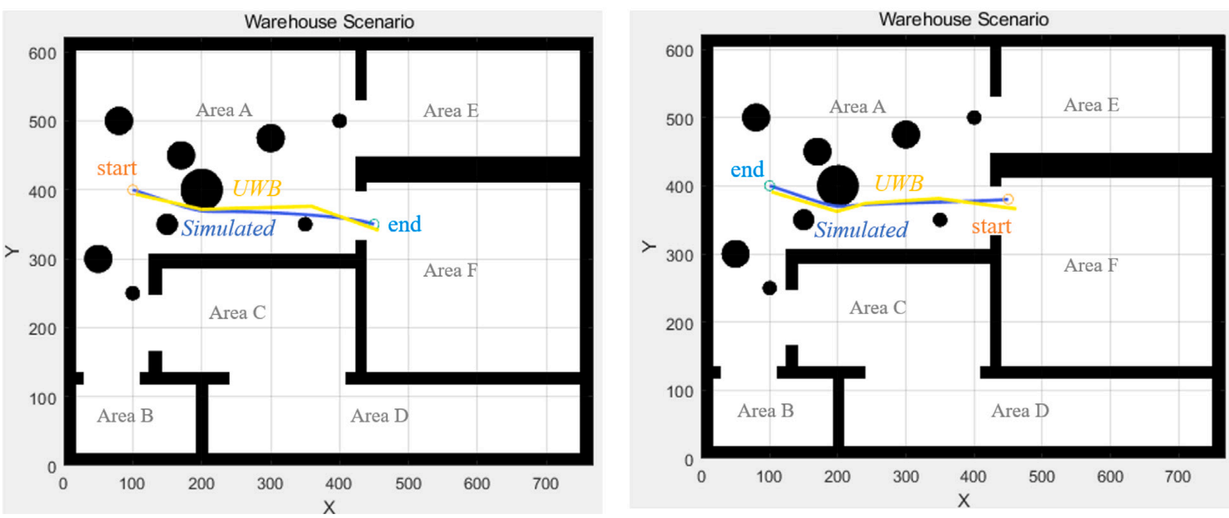
4. Conclusion

The hybrid PSO-SA algorithm improves the PSO algorithm by jumping out of the local optimum to get the best global solution with inspiration from the SA algorithm. It is compared with well-known evolutionary algorithms, including HS, GA, FA, and ABC, by benchmark functions. The proposed PSO-SA algorithm has less 3%, 35%, 40%, 42%, 36%, 44% mean iterations times than PSO, SA, HS, FA, ABC, and GA algorithms respectively. For the runtime of each algorithm, the proposed PSO-SA algorithm has less 26%, 64%, 96%, 100%, 99% and 99% mean values than PSO, SA, HS, FA, ABC, and GA algorithms respectively. The result indicates that it has faster convergence, high accuracy, and less runtime and iterations to get the best solution.

The evolutionary-based approach is the primary path planning approach, providing flexibility and scalability. The proposed hybrid PSO-SA approach is used for AGV path planning. Collision avoidance, smoothness, and path length are considered the cost function for the optimal path. The approach has been compared with the PSO for path planning. The model is validated through the storage scenario



(a) From A to E and from E to A



(b) From A to F and from F to A

Fig. 10. The paths for UWB.

with simulation and experiment, and it can obtain the best path with improved convergence performance. The proposed algorithm can be adapted to different environments with the developed cost function, and it can be applied to more robots easily with robustness. The approach will be modified to adapt to the dynamic environment with multiple robots and moving obstacles as the future work.

CRediT authorship contribution statement

Shiwei Lin: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Project administration, Funding acquisition. **Ang Liu:** Writing – review & editing. **Jianguo Wang:** Writing – review & editing, Supervision, Funding acquisition. **Xiaoying Kong:** Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] A. Ravankar, A.A. Ravankar, Y. Kobayashi, T. Emaru, Symbiotic navigation in multi-robot systems with remote obstacle knowledge sharing, Sensors (Basel) 17 (7) (2017) http://dx.doi.org/10.3390/s17071581, URL https://www.ncbi.nlm.nih.gov/pubmed/28678193. Ravankar, Abhijeet Ravankar, Ankit A Kobayashi, Yukinori Emaru, Takanori eng Switzerland Sensors (Basel). 2017 Jul 5;17(7). pii: s17071581. doi: 10.3390/s17071581.

[2] R. Kumar, A. Haleem, S.K. Garg, R.K. Singh, Automated guided vehicle configurations in flexible manufacturing systems: a comparative study, Int. J. Ind. Syst. Eng. 21 (2) (2015) 207–226, http://dx.doi.org/10.1504/IJISE.2015.071510.

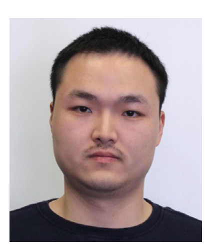
[3] S.G. Kumbhar, R.B. Thombare, A.B. Salunkhe, Automated guided vehicles for small manufacturing enterprises: A review, SAE Int. J. Mater. Manuf. 11 (3) (2018) 253–258, http://dx.doi.org/10.4271/05-11-03-0024.

[4] Y. Yang, Y. Quan, Y. He, Research on multi-AGV management system of autonomous navigation AGVs for manufacturing environment, J. Phys. Conf. Ser. 1910 (1) (2021) 12025, http://dx.doi.org/10.1088/1742-6596/1910/1/012025.

[5] H.-W. Cheong, H. Lee, Requirements of AGV (automated guided vehicle) for SMEs (small and medium-sized enterprises), Procedia Comput. Sci. 139 (2018) 91–94, http://dx.doi.org/10.1016/j.procs.2018.10.222.

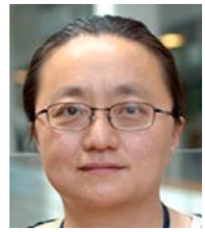
[6] R. Almadhoun, T. Taha, L. Seneviratne, Y. Zweiri, A survey on multi-robot coverage path planning for model reconstruction and mapping, SN Appl. Sci. 1 (8) (2019) http://dx.doi.org/10.1007/s42452-019-0872-y.
[7] S. Tian, Y. Li, Y. Kang, J. Xia, Multi-robot path planning in wireless sensor networks based on jump mechanism PSO and safety gap obstacle avoidance, Future Gener. Comput. Syst. 118 (2021) 37–47, http://dx.doi.org/10.1016/j.future.2020.12.012.
[8] T. Zhao, H. Li, S. Dian, Multi-robot path planning based on improved artificial potential field and fuzzy inference system, J. Intell. Fuzzy Systems 39 (5) (2020) 7621–7637, http://dx.doi.org/10.3233/jifs-200869.
[9] R. Kala, Multi-robot path planning using co-evolutionary genetic programming, Expert Syst. Appl. 39 (3) (2012) 3817–3831, http://dx.doi.org/10.1016/j.eswa.2011.09.090.
[10] Zagradjanin, Pamucar, Jovanovic, Cloud-based multi-robot path planning in complex and crowded environment with multi-criteria decision making using full consistency method, Symmetry 11 (10) (2019) http://dx.doi.org/10.3390/sym11101241.
[11] B. Tang, K. Xiang, M. Pang, Z. Zhanxia, Multi-robot path planning using an improved self-adaptive particle swarm optimization, Int. J. Adv. Robot. Syst. 17 (5) (2020) http://dx.doi.org/10.1177/1729881420936154.
[12] H. Bae, G. Kim, J. Kim, D. Qian, S. Lee, Multi-robot path planning method using reinforcement learning, Appl. Sci. 9 (15) (2019) http://dx.doi.org/10.3390/app9153057.
[13] A.K. Singh, Fault-detection on multi-robot path planning, Int. J. Adv. Res. Comput. Sci. 8 (8) (2017) 539–543, http://dx.doi.org/10.26483/ijarcs.v8i8.4832.
[14] A. Ulusoy, S.L. Smith, X.C. Ding, C. Belta, D. Rus, Optimality and robustness in multi-robot path planning with temporal logic constraints, Int. J. Robot. Res. 32 (8) (2013) 889–911, http://dx.doi.org/10.1177/0278364913487931.
[15] S. Wen, Z. Wen, D. Zhang, H. Zhang, T. Wang, A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning, Appl. Soft Comput. 110 (2021) http://dx.doi.org/10.1016/j.asoc.2021.107605.
[16] Y. Yang, L. Juntao, P. Lingling, Multi-robot path planning based on a deep reinforcement learning DQN algorithm, CAAI Trans. Intell. Technol. 5 (3) (2020) 177–183, http://dx.doi.org/10.1049/trit.2020.0024.
[17] G. Sun, R. Zhou, B. Di, Z. Dong, Y. Wang, A novel cooperative path planning for multi-robot persistent coverage with obstacles and coverage period constraints, Sensors (Basel) 19 (9) (2019) http://dx.doi.org/10.3390/s19091994, URL https://www.ncbi.nlm.nih.gov/pubmed/31035410. Sun, Guibin Zhou, Rui Di, Bin Dong, Zhuoning Wang, Yingxun eng 61773031/National Natural Science Foundation of China 61573042/National Natural Science Foundation of China Switzerland Sensors (Basel). 2019 Apr 28;19(9). pii: s19091994. doi: 10.3390/s19091994.
[18] H.Y. Lin, Y.C. Huang, Collaborative complete coverage and path planning for multi-robot exploration, Sensors (Basel) 21 (11) (2021) http://dx.doi.org/10.3390/s21113709, URL https://www.ncbi.nlm.nih.gov/pubmed/34073565. Lin, Huei-Yung Huang, Yi-Chun eng MOST 106-2221-E-194-004/Ministry of Science and Technology, Taiwan Switzerland Sensors (Basel). 2021 May 26;21(11). pii: s21113709. doi: 10.3390/s21113709.
[19] S.-K. Huang, W.-J. Wang, C.-H. Sun, A path planning strategy for multi-robot moving with path-priority order based on a generalized voronoi diagram, Appl. Sci. 11 (20) (2021) http://dx.doi.org/10.3390/app11209650.
[20] P.K. Das, H.S. Behera, S. Das, H.K. Tripathy, B.K. Panigrahi, S.K. Pradhan, A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment, Neurocomputing 207 (2016) 735–753, http://dx.doi.org/10.1016/j.neucom.2016.05.057.
[21] M. Kapanoglu, M. Alikalfa, M. Ozkan, A. Yazıcı, O. Parlaktuna, A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time, J. Intell. Manuf. 23 (4) (2010) 1035–1045, http://dx.doi.org/10.1007/s10845-010-0404-5.
[22] P.K. Das, P.K. Jena, Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators, Appl. Soft Comput. 92 (2020) http://dx.doi.org/10.1016/j.asoc.2020.106312.
[23] M. Nazarahari, E. Khanmirza, S. Doostie, Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm, Expert Syst. Appl. 115 (2019) 106–120, http://dx.doi.org/10.1016/j.eswa.2018.08.008.
[24] P.K. Das, Hybridization of kidney-inspired and Sine–Cosine algorithm for multi-robot path planning, Arab. J. Sci. Eng. 45 (4) (2019) 2883–2900, http://dx.doi.org/10.1007/s13369-019-04193-y.
[25] R.A. Saeed, D. Reforgiato Recupero, P. Remagnino, The boundary node method for multi-robot multi-goal path planning problems, Expert Syst. 38 (6) (2021) http://dx.doi.org/10.1111/exsy.12691.
[26] P.K. Das, H.S. Behera, B.K. Panigrahi, A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning, Swarm Evol. Comput. 28 (2016) 14–28, http://dx.doi.org/10.1016/j.swevo.2015.10.011.
[27] R. Eberhart, J. Kennedy, A New Optimizer Using Particle Swarm Theory, IEEE, pp. 39–43, http://dx.doi.org/10.1109/MHS.1995.494215.
[28] H. Garg, A hybrid PSO-GA algorithm for constrained optimization problems, Appl. Math. Comput. 274 (2016) 292–305, http://dx.doi.org/10.1016/j.amc.2015.11.001.
[29] A.R. Jordehi, Enhanced leader PSO (ELPSO): A new PSO variant for solving global optimisation problems, Appl. Soft Comput. 26 (2015) 401–417, http://dx.doi.org/10.1016/j.asoc.2014.10.026.
[30] P. Liu, J. Liu, Multi-leader PSO (MLPSO): A new PSO variant for solving global optimization problems, Appl. Soft Comput. 61 (2017) 256–263, http://dx.doi.org/10.1016/j.asoc.2017.08.022.
[31] Z. Meng, J.-S. Pan, H. Xu, Quasi-affine transformation evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization, Knowl.-Based Syst. 109 (2016) 104–121, http://dx.doi.org/10.1016/j.knosys.2016.06.029.
[32] N. Geng, D.W. Gong, Y. Zhang, PSO-based robot path planning for multisurvivor rescue in limited survival time, Math. Probl. Eng. 2014 (2014) 1–10, http://dx.doi.org/10.1155/2014/187370.
[33] E. Krell, A. Sheta, A.P.R. Balasubramanian, S.A. King, Collision-free autonomous robot navigation in unknown environments utilizing PSO for path planning, J. Artif. Intell. Soft Comput. Res. 9 (4) (2019) 267–282, http://dx.doi.org/10.2478/jaiscr-2019-0008.
[34] S. Sahu, B.B. Choudhury, PSO based path planning of a six-axis industrial robot, Adv. Intell. Syst. Comput. (2020) 213–220, http://dx.doi.org/10.1007/978-981-13-8676-3_19, book section Chapter 19.
[35] K. Sameshima, K. Nakano, T. Funato, S. Hosokawa, Strrt-based path planning with PSO-tuned parameters for RoboCup soccer, Artif. Life Robot. 19 (4) (2014) 388–393, http://dx.doi.org/10.1007/s10015-014-0177-6.
[36] B. Song, Z. Wang, L. Zou, An improved PSO algorithm for smooth path planning of mobile robots using continuous high-degree Bezier curve, Appl. Soft Comput. 100 (2021) http://dx.doi.org/10.1016/j.asoc.2020.106960.
[37] S. Tian, Y. Li, J. Li, G. Liu, A. Maseleno, X. Yuan, V.E. Balas, Robot global path planning using PSO algorithm based on the interaction mechanism between leaders and individuals, J. Intell. Fuzzy Systems 39 (4) (2020) 4925–4933, http://dx.doi.org/10.3233/jifs-179978.
[38] S. Kathpal, R. Vohra, J. Singh, R.S. Sawhney, Hybrid PSO – SA algorithm for achieving partitioning optimization in various network applications, Procedia Eng. 38 (2012) 1728–1734, http://dx.doi.org/10.1016/j.proeng.2012.06.210.
[39] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Engrg. 194 (36) (2005) 3902–3933, http://dx.doi.org/10.1016/j.cma.2004.09.007.
[40] X.-S. Yang, Firefly Algorithm, Lévy Flights and Global Optimization, Springer London, London, 2009, pp. 209–218, http://dx.doi.org/10.1007/978-1-84882-983-1_15.
[41] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Global Optim. 39 (3) (2007) 459–471, http://dx.doi.org/10.1007/s10898-007-9149-x.
[42] D.E. Goldberg, J.H. Holland, Genetic algorithms and machine learning, Mach. Learn. 3 (2) (1988) 95–99, http://dx.doi.org/10.1023/A:1022602019183.

Ms Shiwei Lin obtained her B.Sc. degree in Science of Information Technology from University of Technology Sydney (UTS) in 2019. She is a Ph.D. candidate in School of Civil and Environmental Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney (UTS). Her research focuses on mobile robots' path planning and navigation algorithms, optimization framework, multi-robot systems, robotics and intelligent transportation, and sensor fusion.



Ang Liu is a Ph.D. student in the School of Civil and Environmental Engineering, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. His current research focuses on the indoor position system, UWB position and sensors fusion.

Dr Jianguo Wang was appointed as a Chancellor's Postdoctoral Research Fellow with the ARC Centre of Excellence for Autonomous Systems (CAS) in UTS early 2009. Currently, he is a lecturer at the School of Civil and Environmental Engineering. He received his B.Sc. in Physics from Nanjing University, M.Sc. in Electrical and Information Engineering from The University of Sydney (USyd), and PhD in Geomatics from The University of New South Wales (UNSW). Dr Wang was awarded China Scholarship Council research fellowship in 1997 and came to Australia as a visiting scholar in USyd. From 1999 to 2008, he studied and worked at USyd and UNSW as a postgraduate student and research associate. He also served as a research fellow, senior engineer and project manager overseas and in Australia. With more than 20 years of broad experience in academic and industry, he is continuously developing his multi-discipline researches in sensor fusion for surveying, navigation and perception; robotics and intelligent systems; environment friendly transportation and housing; GNSS, IMU, Vision and Laser sensors modelling and data fusion etc.



Dr Xiaoying Kong has broad interests in control engineering and software engineering. Her recent research work has been in GPS, inertial navigation systems, data fusion, robotics, sensor networks, Internet monitoring systems, agile software development methodologies, web technologies, web design methodologies, web architecture framework, analytical models of software development, and value-based software engineering. She has many years work experience in aeronautical industry, semiconductor industry, and software industry.

Update

Journal of Computational Science

Volume 76, Issue , March 2024, Page

DOI: https://doi.org/10.1016/j.jocs.2023.102183

Journal of Computational Science 76 (2024) 102183

Contents lists available at ScienceDirect

Journal of Computational Science

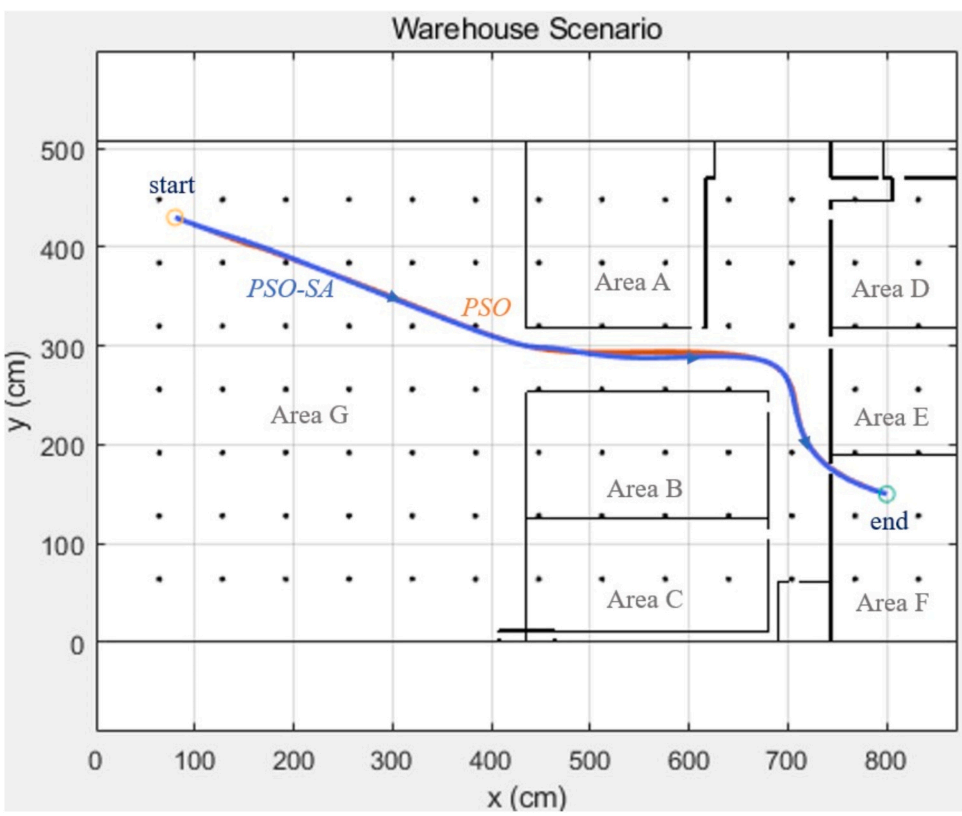
journal homepage: www.elsevier.com/locate/jocs

Corrigendum to "An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse" J. Comput. Sci. 67 (2023) 101938

S. Lin [a,*], A. Liu [a], J. Wang [a], X. Kong [b]

[a] Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW, Australia
[b] School of IT and Engineering, Melbourne Institute of Technology, Sydney Campus, Sydney, NSW, Australia

The authors regret Fig. 6(c) was not the correct version that we intended to include. The correct version of the figure is:



The authors would like to apologise for any inconvenience caused.

DOI of original article: https://doi.org/10.1016/j.jocs.2022.101938.
* Corresponding author.
E-mail address: Shiwei.Lin-1@student.uts.edu.au (S. Lin).

https://doi.org/10.1016/j.jocs.2023.102183

Available online 24 November 2023