Autonomous Mobile Robot Navigation Based on PSO Algorithm with Inertia Weight Variants for Optimal Path Planning

Tahseen Fadhel Abaas1,2 and Alaa Hassan Shabeeb1,3

1 Department of Production Engineering and Metallurgy, University of Technology, Baghdad-Iraq
2 Email: 70047@uotechnology.edu.iq
3 Email: alaa.h.shabeeb@uotechnology.edu.iq

Abstract: Motion planning is an important domain since its performance can significantly affect the utilization of robots. This paper addresses our work to developing a path planner for wheeled a mobile robot using a swarm intelligence technique for optimal path planning within a short computational time to get better path planning results. Through this technique, we developed particle swarm optimization (PSO) for generating fast and optimal path planning. Inertia weight technique is used for performance comparison of PSO Algorithms to get optimal path planning within a complex environment, PSO with a time-varying mechanism for the inertia weight values (TV-IWPSO), to analyze the performance proposed approach on the PSO algorithm performance. Finally, the comparison has been done in between TV-IWPSO with both particle swarm optimization with constant inertia weight (B-PSO), and standard particle swarm optimization (S-PSO), in two different maps to perform analysis for algorithms through various environments. The simulation results, which carried out using Matlab 2018a showed that the PSO algorithm with inertia-weight strategy made good results for generating optimal path planning and efficiently than (S-PSO) and (B-PSO) in terms of path distance, execution-time

Keywords: Metaheuristic algorithm (PSO), Path Planning, Obstacle Avoidance, Mobile Robot.

1. Introduction

In the past few decades, as Artificial Intelligence (AI) technology evolves rapidly, the intelligent wheeled mobile robot (WMR), and their application is extended in many industrial fields involving factory inspection, artificial intelligence, intelligent production, and others. This growth in applications of WMR is related to the need for safety for human life, where in the last years that witnessing the Robotic Engineering is adoption in different industrial and commercial environments. For example, in an industrial building is using autonomous mobile robots to move materials and other manufacturing applications, in hospitals are used to track patient health and move materials, while the military and defense sector uses professional service robots that are deployed for combat scenarios, therefore. Techniques for autonomous navigation of wheeled mobile robot motion problem (WMR-
MP) has become the most important task in the robotics field, which involves determining the optimal path or near-optimal path free of collisions between the initial position and the target destination with specific conditions of constraint by using intelligent equipment attached with WMR to convert high-task requirements to low-requirements for humans [1,2].

The motion planning problem of WMR is divided into route planning (path planning) and preparation of the trajectory (trajectory planning). The Path planning deals with generation an optimal collision-free path from the beginning to the endpoint with taking into account the spatial features of the obstacles and the kinematic limitations of the WMR while trajectory planning deals with the dynamic constraints of the WMR, obstacles Moving or obstacles not previously identified which are time-dependent [3]. Generally, WMR-MP can be classified into two different evaluation categories. one is based on WMR environment path planning, it is can be a static or dynamic environment, in the static environment, the details about the environment and the location of all obstacles do not alter over time while in the dynamic environment the target location and the obstacle can change partially or entirely unknown. The other is focused on the perception of WMR about the environmental facts which Splitting the WMR-MP into global and local path planning. Global path planning means WMR can obtain complete global environmental model details, including all obstacle information, destination information, but because of the slowness caused by their complexity, this technique is suitable only in the case of a static area where obstacles not moving. On the other hand, local route planning can only get information about local areas. The biggest difference between local and global strategies lies in the preparation of path planning [4]. The whole process of route planning can be considered as a series of continuous local route planning processes, thus should the combining local path planning and global path planning thus, the combination of local path planning and global path planning for ensuring that the developed hybrid system makes use of its advantages and avoids the weakness of each method. According to the above, the basic autonomous WMR navigation can be divided into three tasks: environment modeling in the form map, Path Planning, and Control system.

Thus, the path planning problem of the navigation path is classified as an optimization problem, so, the designing a fast and efficient procedure for navigation is an essential step in Mobile robots path planning, WMR has multiple paths for reaching the target point in any task, But reckoning of the best route depends on the adjusted criteria of some guidelines Such as the shortest distance, consumed time and energy. Many optimization algorithms have been used to control the path planning problem and collision avoidance [5,6]. These algorithms are divided into two approaches including the classical approaches such as (Cell decomposition methods, road-map based approaches), Metaheuristics approaches-based methods such as (fuzzy logic control, neural networks, genetic algorithm, swarm intelligence algorithm). In general, the Classical approaches not suitable for real-time mobile path planning due to high computation cost and failure to respond to the uncertainty percent in the environments less able to solve problems in complex environments compares with metaheuristic techniques which easy in implementation with a great ability to handle the uncertainty present in the environment [7,8]. Thus, select the appropriate algorithm of the path planning process is a vital issue to ensure that MR navigation optimally in the environment. This paper proposed a methodology based on a particle swarm optimization (PSO) algorithm to find optimal navigation for a WMR. PSO is an optimization tool and based on swarm intelligent, easily in the implement, effective, and fast, it's applied in many optimization problems [9]. Each optimization algorithm mentioned above needs to tackle the exploration and exploitation of the workspace. To achieve results, an optimization algorithm needs to set a suitable ratio between exploration and exploitation. In this paper, a modified PSO (M-PSO) is proposed to Equilibria the exploration and exploitation trade-off in the standard PSO algorithm. various approaches are formed to tune the parameters of the PSO algorithm for varying exploration and exploitation combinations during iterations. The exploration ability and exploitation ability have a related impact on the searching performance of an algorithm. For the PSO, exploration is referred to as a particle stay away from the original search path to some point and looking a new path, which indicates the particle’s ability to exploit regions unknown. Exploitation is referred to that a particle that continues to looking more closely on the original trajectory to a certain extent, that will ensure the particle makes a detailed search to the area that has been explored. But how to make the algorithm get a workable balance between exploration and exploitation is a Problem worth studying.
In this paper Inertia weight technique is proposed to achieve a proper compromise between exploration and exploitation to improve the performance of standard PSO. An optimization model is designed to find minimal path length with taken into consideration path safety. A modified PSO algorithm proposed for generating optimal path planning and efficiently than Standard particle swarm optimization (S-PSO). The contributions of our study are as follows:

(1) The study of the inertia weight effect on performance for the PSO algorithm to find the optimal path through various environments and measure its effect on the efficiency of the PSO algorithm.
(2) The compare the effective strategy proposed of WMR movement path planning in an environment with a fixed position of the obstacles and target points to reduce the total time for the path-finding.
(3) Prove the effectiveness of our methods by conducting simulations with various scenarios.

2. The Objectives
The objectives considered in this research for generation optimal navigation path through find the best and feasible path to WMR and study the effect inertia weight parameter on performance for the PSO algorithm to find the optimal path through various environments. Before discussing our algorithm, there are some assumptions about the model we use in this paper.

2.1 The Map Building Description
The WMR navigation in the 2D environment requires existence a map for located start and destination points, it's used to design WMR optimal movement, WMR represented in this map as point by a set of Cartesian coordinate positions (x,y) in the map, that contains many static obstacles. Each obstacle is surrounded in disk with a radius equal to the value of the WMR radius (R-WMR) depending on the studied WMR shape, to assure the safety of the robot while trying movement in the environment, as shown in Figure 1. The WMR has no prior knowledge of environmental parameters such as locations, shapes, and sizes of the obstacles, and with existence equipping overhead sensor (camera) for on-line detection to the workspace with Compass modules for located the WMR, and build the map in the same time.

![Figure 1. Extend obstacle size corresponding to WMR.](image)

3. Overview of Particle Swarm Optimizer Algorithm (PSO)
PSO algorithm is classified as part of the meta-heuristic optimization algorithms, it's based on swarm-intelligence, put forward by Kennedy and Eberhart in 1995 [10]. Swarm intelligence is based on strategies of the systems that its individuals tend to show an intelligent group behavior to reach to intellectual level the higher than the smart of any one particle in the swarm.

3.1 Standard Particle Swarm Optimization (S-PSO)
PSO is initialized with a random particulate population using an even distribution. PSO is an optimization method depends on swarm intelligence, each individual in swarm called a particle, each particle in a swarm is the representing potential solution, a particle moving iteratively through the problem space in search of the best-fitted solution, at initialized, the system starts with a population of
the random solutions, also, everyone in a population has random position and velocity that guide the particle through its navigation space [11]. Each particle in PSO follows a track in research space and updating constantly a velocity vector and position vector for each particle. also, after each iteration, each particle moves toward of the two “best” values are the best fitness ($P_{\text{Best}}$) has achieved for each particle and another “best” value ($G_{\text{Best}}$) is the best fitness achieved over the whole swarm. This cycle continues, until either the desired destination is found or the largest number of iterations is reached.

Let us a D-dimensional search space (d) with the initial population (swarm) of size N, each individual (particle) position in a swarm is given by a 2xn-matrix, it is denoted $L_i$.

$$L_i = \begin{bmatrix} x_{i1} & x_{i2} & ... & x_{in} \\ y_{i1} & y_{i2} & ... & y_{in} \end{bmatrix}^T$$

(1)

The previous-velocity of this particle is denoted by another 2xn-matrix is denotes $V_i$.

$$V_i = \begin{bmatrix} v_{x1} & v_{x2} & ... & v_{xn} \\ v_{y1} & v_{y2} & ... & v_{yn} \end{bmatrix}^T$$

(2)

The previous best-visited position (PBest) after each generation has achieved for each particle is denoted as $P_i$.

$$P_i = \begin{bmatrix} p_{x1} & p_{x2} & ... & p{x_{in}} \\ p_{y1} & p_{y2} & ... & p{y_{in}} \end{bmatrix}^T$$

(3)

The best-visited position (GBest) after each iteration cross the whole swarm is denoted as $P_g$.

$$P_g = \begin{bmatrix} p_{x1} & p_{x2} & ... & p{g_{xn}} \\ p_{y1} & p_{y2} & ... & p{g_{yn}} \end{bmatrix}^T$$

(4)

Each particle is updating its velocity and position by following two-equation.

\[ V_{i(t+1)} = V_{i(t)} + C_1 r_1 (P_{i(t)} - S_{i(t)}) + C_2 r_2 (P_g - S_{i(t)}) \]  
\[ L_{i(t+1)} = L_{i(t)} + V_{i(t+1)} \]

(5)

(6)

Also, each particle in swarm randomly initialized for velocity and position within a uniform range from [xmin, xmax] and [vmin, vmax] respectively.

\[ x_i = x_{\text{min}} + \rho_1 (x_{\text{max}} - x_{\text{min}}) \]  
\[ v_i = v_{\text{min}} + \rho_2 (v_{\text{max}} - v_{\text{min}}) \]

(7)

(8)

Where $\omega$ is inertia weight; the parameters $c_1$ and $c_2$ are balance factors between the individual knowledge and the collective knowledge when particle flying towards the target; $r_1$ and $r_2$: $i=1:N$; N is the number-size of the particle; $t=1:t$-max; t is the number of iterations; $r_1$, $r_2$, $\rho_1$ and $\rho_2$ represent random-numbers from 0 to 1.

3.2. Modified PSO based on its Parameters.

The PSO strategy based on tuning its parameters considers an important key to reaching toward the optimum accurate and solution, in this part, we taken the Inertia weight parameter in two approaches are constant value and time-varying to study effect inertia weight parameter on performance for the PSO algorithm.

3.2.1. PSO with constant Inertia Weight value (B-PSO).

In the literature, many variations of the PSO algorithm was developed. [12], that showed that PSO searches for range-wide applications effectively but lacks to balance between exploration and exploitation at the end of the optimization processing in some cases. The researchers proposed using inertia weight ($\omega$), as a workable solution to enhance the velocity updating of the particle. In this study, modified PSO by constant inertia weight ($\omega$) strategy, is known as basic particle swarm
optimization (B-PSO), according to that, updating the velocity equation is rewritten by the following equation.

\[
V_{t+1}^{i_d} = \omega V_t^{i_d} + C_1 r_1 (p_t - S_t^{i_d}) + C_2 r_2 (p_g - S_t^{i_d})
\]  

(9)

3.2.2. PSO with Time-Varying of the Inertia Weight value (TV-IWPSO).

The inertia-weight \( \omega \) is considered the most impact parameter in PSO algorithm behavior. The inertia weight is used to control the effect of the bygone velocity of velocities on the Present velocity. we used time-varying as a mechanism to linearly decreasing inertia weight value as an approach to balance between the global search (exploration) and local search (exploitation) of the swarm. where the small value of \( \omega \) helps the individuals of the swarm in local search, while the high value of \( \omega \) help individuals of the swarm to explore the problem space more easily and improve global search, for balance between global and local search exploration. In the last years a lot of the researchers studied TV-IW approach as a significant improvement mechanism for the performance of PSO. This approach can enhance the exploration in initial search iterations and increase the exploitation during the final search iterations of the search [13]. Equation (10), shows how the \( \omega \) value is updated.

\[
\omega = (\omega_{\text{max}} - \omega_{\text{min}}) \frac{\text{iter}}{\text{Iter}_{\text{max}}} + \omega_{\text{max}}
\]  

(10)

Where the \( \omega_{\text{max}} \) and the \( \omega_{\text{min}} \) are the max and min value of the inertia weight; Itermax is the max value of iterations; iter is the present iteration.

4. Evaluation Function of PSO

| Table 1: General Pseudo_code for Proposed Techniques (PSO). |
|-------------------------------------------------------------|
| 1- Input: set parameters to initialize (swarm size, Itermax , c1, c2, wmin, Wmax, Cmax, Cmin, w). |
| 2- Select the type of PSO strategy |
| 3- Output: G_{Best} and its points (x,y). |
| for each particle I (i=1 to N) do |
| Position and velocity initialized by random |
| end for |
| iter \( \leftarrow \) 0 |
| while (iter < iter_{max}) do |
| for each particle i (i=1 to N) |
| Fitness Function Assessment \( f_{(final)} \) for each particle. |
| Update position \( V_{t+1}^{i_d} \) and velocity \( L_{t+1}^{i_d} \) based on Equs (5) & (6). |
| Fitness Assessment to update P_{Best} & G_{Best}. |
| End for |
| iter \( \leftarrow \) iter+1 |
| end while |


The Evaluation Function is of the most important of the algorithm, it should be precisely examined. PSO algorithm is based on the behavior of swarm to find the food (target) source with minimum time and path length with consideration of the obstacle’s avoidance. So, the evaluation Function in this research based on two estimation functions to optimize the objective function, as described below.

4.1 Minimization of the Length path between Initial and Target Point

Euclidean distance is the criteria to find minimal path length (minimum optimization) is defined as the distance between the WMR and the goal position in each iteration. The performance evaluation considered in this research is positions that are given the minimum value of objective function must be selected for the next move of a WMR, hence for tracking. The minimal path length from the initial point \((R_x, R_y)\) to target point \((T_X, T_Y)\) via way-points in the 2D environment given by the following formula.

\[
F_1(i) = \sqrt{(R_x - T_X)^2 + (R_y - T_Y)^2} \quad ............. (11)
\]

where:

\(F_1(i)\): distance function between WMR and target.
\(i\): current iteration
\((R_x, R_y)\): coordinates of the start position.
\((T_X, T_Y)\): coordinates of the target position.
\((R_x, R_y)\): coordinates of the WMR at the current position.

4.2 Optimum Safe Distance from Obstacles.

A second part to completely optimize the objective function is optimum avoiding of the collision during path planning, the WMR should have the safety distance from the obstacles and it should be equal or larger than the radius of WMR \((R_{WMR})\). Therefore, we used penalty value, if the maximum distance between WMR and obstacle is equal or less than safety distance \((\epsilon)\), the penalty value will be added to the main function \((F_1)\) and path become infeasible because the optimization problem is minimization, otherwise penalty value is equal to zero, the distance between WMR and obstacle can be calculated by the following formula.

\[
Dis(i) = \sqrt{(R_x - O_{xn})^2 + (R_y - O_{yn})^2} \quad ............. (12)
\]

Where:

\(Dis(i)\): Distance function between WMR and obstacle.
\((O_{xn}, O_{yn})\): Coordinates of the obstacle, \(n\) is the number of an obstacle.

\[
F_2(i) = \begin{cases} 
1, & \text{if } Dis \leq \epsilon \\
0, & \text{otherwise} 
\end{cases}
\]

Hence, the total fitness function is

\[
Z_i = F_1(i) + F_2(i) \quad ............. (13)
\]

Where \(\epsilon\) is a minimum distance between intermediate path points and obstacle. Based on Equ (13), the local best position \((P_{best})\) and global best position \((G_{best})\), and the new position for WMR is determined. This process repeats for interpolating positions (intermediate points) until the WMR is reaching the target point.

5. Performance Study and Simulation Analysis

5.1 Setting and Environment

To getting simulation results for the performance of developed PSO algorithms with taking into consideration the effect interior weight and acceleration coefficients on performance for the algorithm to find the optimal path through two environments with taking different numbers of obstacles, the
proposed methods in this study are coded in MatLab R2018a and tested on Intel(R) core i7, 2.2 GHz CPU, 8.00 GB RAM system The Parameters for Simulation that used in the PSO Each environment is shown in table 2.

Table 2. Parameters for Simulation that used in the PSO

| Method   | Parameters | Number of iterations | Number of particles |
|----------|------------|----------------------|---------------------|
| S-PSO    | $c_1$, $c_2$=2 | 100                 | 80                  |
| B-PSO    | $c_1$, $c_2$=2, w=0.7 |                 |                    |
| TV-IWPSO | $c_1$, $c_2$=2, $w_{\text{max}}$=0.9, $w_{\text{min}}$=0.4 | | |

We created two maps with dimension (10x10) units during simulation setting, each the map is evaluated by proposed methods, The WMR is circular in form and red in color and its move from the starting point (0,0) to the goal point (10,10) with several obstacles for each map, and location of each obstacle is given in table 3 for each map.

Table 3. Obstacle location in each map

| Environment | No of obstacles | Obstacle options in (unit) |
|-------------|----------------|---------------------------|
|             |                | (x,y, radius)              |
| Map 1       | 5              | (2,4,0.5), (2,8,0.75), (8,6,0.5), (8,2,0.75), (8,6,0.5) (5,5,1) |
| Map 2       | 9              | (2,6,0.5), (3,7,0.5), (8,2,0.75) (5,8,0.5), (5,6,0.5) (5,4,2,0.5), (4,1,0.5), (1,2,0.37), (7,4,0.75), |

5.2. Simulation Results

In this part, we compared and an evaluated of results of our developed PSO strategy in a simulation environment and modeling each map and its fitness function base on the WMR sensory information, the first environment is map1 with five obstacles, the second environment is map2 with nine obstacles, the result of simulation proved that proposed algorithms are effective in finding the optimal path in all of two maps as shown in figures (2, 3). The results evaluated through compares the optimal path length and the algorithm execution time in each map as summarized in tables (4,5), after analysis, inertia weight has impact factor on performance for the PSO algorithm in both maps, results indicated that inertia weight is more effective than S-PSO and B-PSO on results for the traveled path length and the algorithm execution time.

Table 4. Comparison of the length of the Traveled Path and time of the PSO algorithms in map one.

| Environment | PSO method | Optimal path length (in pixels) | PSO execution time (in seconds) | Number of best generations |
|-------------|------------|---------------------------------|---------------------------------|---------------------------|
| Map1        | S-PSO      | 14.408                          | 1.6054                          | 64                        |
|             | B-PSO      | 14.3872                         | 1.5719                          | 73                        |
|             | TV-IWPSO   | 14.3792                         | 1.5354                          | 83                        |
Table 5. Comparison of the length of the Traveled Path and time of the PSO algorithms in map two.

| Environment | PSO method | Optimal path length (in pixels) | PSO execution time (in seconds) | Number of best generations |
|-------------|------------|---------------------------------|---------------------------------|-----------------------------|
| Map2        | S-PSO      | 14.2179                         | 1.6654                          | 66                          |
|             | B-PSO      | 14.3994                         | 1.6661                          | 55                          |
|             | TV-IWPSO   | **14.1796**                     | **1.6606**                      | 88                          |

(a): Best path found by S-PSO  
(b): Variation of optimal path length with iterations by S-PSO  
(c): Best path found by B-PSO  
(d): Variation of optimal path length with iterations by B-PSO
Figure 2: comparison of path length optimization of developed methods based on PSO in Map1
Figure 3: comparison of path length optimization of developed methods based on PSO in Map2

6. Discussion and Conclusion

This paper presents path planning and obstacle avoidance to the wheeled mobile robot using PSO algorithm, we studied the performance of the PSO algorithm depending on variant parameters strategy for fine-tuning its parameters, we selected strategy based on TV-IW approach for enhancing the balance between the exploration and exploitation of PSO algorithms to get better path planning results for reducing the total time for the pathfinding. Compared results of the performance analysis of our developed PSO algorithms showed that our strategy can successfully track the optimal path with avoiding the collision. Additionally, the analysis of the results showed that both inertia weight parameter affects the performance of the PSO algorithm to find the optimal path, results indicate showed that inertia weight parameter performance enhances to get better path planning results.

In future works, we would like to study our work through a comparison with another algorithm.

References

[1] Patle, B. K., Pandey, A., Parhi, D. R. K., & Jagadeesh, A, 2019. A review: On path planning strategies for navigation of mobile robot Defence Technology, vol. 15, pp. 582–585.
[2] Bayat, F., Najafinia, S. and Aliyari, M., 2018. Mobile robots path planning: Electrostatic potential field approach. Expert Systems with Applications, 100, pp.68-78.
[3] Mac, T.T., Copot, C., Tran, D.T. and De Keyser, R., 2017. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. Applied Soft Computing, 59, pp.68-76.
[4] Abbadi, A. and Přenosil, V., 2015. Safe path planning using cell decomposition approximation. Distance Learning, Simulation and Communication, 8, pp.1-6.
[5] Hoang, V.D., Hernández, D.C., Hariyono, J. and Jo, K.H., 2014, June. Global path planning for unmanned ground vehicle based on road map images. In 2014 7th International Conference on Human System Interactions (HSI).pp. 82-87. IEEE.
[6] Lamini, C., Benhlima, S. and Elbekri, A., 2018. Genetic algorithm based approach for autonomous mobile robot path planning. Procedia Computer Science, 127, pp.180-184.
[7] Akka, K. and Khaber, F., 2018. Mobile robot path planning using an improved ant colony optimization. International Journal of Advanced Robotic Systems, 15(3), pp1-7.

[8] Adamu, P.I., Jegede, J.T., Okagbue, H.I. and Oguntunde, P.E., 2018. Shortest Path Planning Algorithm—A Particle Swarm Optimization (PSO) Approach.

[9] Wang, D., Tan, D. and Liu, L., 2018. Particle swarm optimization algorithm: an overview. Soft Computing, 22(2), pp.387-408.

[10] Wang, D., Tan, D. and Liu, L., 2018. Particle swarm optimization algorithm: an overview. Soft Computing, 22(2), pp.387-408.

[11] Lei, X., Wang, F. and Tan, Y., 2019. Swarm intelligent optimization algorithms and its application in mobile robot path planning. In Rapid Automation: Concepts, Methodologies, Tools, and Applications (pp. 609-648). IGI Global.

[12] Li, M., Chen, H., Wang, X., Zhong, N. and Lu, S., 2019. An improved particle swarm optimization algorithm with adaptive inertia weights. International Journal of Information Technology & Decision Making, 18(03), pp.833-866.

[13] Zhang, G., Li, C., Gao, M. and Sheng, L., 2019. July. Global Smooth Path Planning for Mobile Robots Using a Novel Adaptive Particle Swarm Optimization. In 2019 Chinese Control Conference (CCC), pp. 2124-2129. IEEE.