

Hybrid approach for multi-objective optimization path planning with moving target

Baraa M. Abed, Wesam M. Jasim

Department of Computer Science, College of Computer Science and Information Technology, University of Anbar, Ramadi, Iraq

Article Info

Article history:

Received Jul 11, 2022

Revised Sep 12, 2022

Accepted Sep 28, 2022

Keywords:

Bat algorithm

Dynamic environment

Hybrid approach

Moving target

Multi objective optimization

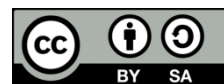
Particle swarm optimization algorithm

Path planning

ABSTRACT

Path planning algorithms are the most significant area in the robotics field. Path planning (PP) can be defined as the process of determining the most appropriate navigation path before a mobile robot moves. Path planning optimization refers to finding the optimal or near-optimal path. Multi-objective optimization (MOO) is concerned with finding the best solution values that satisfy multiple objectives, such as shortness, smoothness, and safety. MOO present the challenge of making decisions while balancing these contradictory issues through compromise (tradeoff). As a result, there is no single solution appropriate for all purposes in MOO, but rather a range of solutions. Several objectives are considered as part of this study, including path security, length, and smoothness, when planning paths for autonomous mobile robots in a dynamic environment with a moving target. Particle swarm optimization (PSO) algorithms are combined with bat algorithms (BA) to make a balance between exploration and exploitation. PSO algorithms used to optimize two important parameters of the bat algorithm. The proposed solution is tested through several simulations based on varying scenarios. The results demonstrate that mobile robots can travel clearly and safely along short paths and smoothly, proving this method's efficiency.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Baraa M. Abed

Department of Computer Science, College of Computer Science and Information Technology

University of Anbar

Ramadi, Iraq

Email: burasoft@gmail.com

1. INTRODUCTION

Mobile robots are finding applications in many different industries these days, including those dealing with entertainment, medical, mining, rescue, education, the military, aerospace, and even agriculture, to mention just a few of them. In the field of robotics, path planning algorithms play the most significant role. A lot of research has been done on this problem in robotics. PP is defined as the process of determining the best feasible path that allows mobile robots to move from a starting point to a target point of the environment before they move, which is considered the main problem in robot navigation [1]. There are three types of environments in the field of autonomous path planning for mobile robots: static environments with fixed obstacles, dynamic environments with movable obstacles, and dynamic environments with movable obstacles and the target point at the same time. The dynamic environment is an unknown environment with multiple portable obstacles moving at the same time with the robot all over the space at random speeds, directions, and speeds. As well as static obstacles (e.g. walls, doors, and furniture), it might also include some moving obstacles [2]. Based on the robot's knowledge of its environment, path planning can be divided into two categories: global (or offline) and local (or online). Mobile robots are capable of planning their global path based on their full knowledge of their surroundings. A robot follows a complete path which is generated by an

algorithm before it begins to move. In order to determine the path of a mobile robot locally, the robot relies on a local sensor to collect information about the environment and construct a new path based on that information [3]. The cell decomposition and roadmap have been used to solve problems related to path planning. There are two main disadvantages to this method: its inefficiency caused by high processing fees and its unreliability caused by local minima risk. A number of heuristic techniques can be used to overcome these limitations, including neuronal pathways, genetics, and algorithms that are inspired by nature [4]. There have been several methods used to solve the path planning problem, including safety and single-objective optimization, such as particle swarm optimization (PSO) [5], cuckoo search (CS) [6], artificial immune systems (AIS) [7], whale optimization algorithm (WOA), bat algorithm (BA) [8], and bacterial foraging optimization (BFO) [9]. In addition, neural networks and fuzzy logic have been implemented. In either static or dynamic settings, these methods aim to optimize for many criteria at once, such as the shortest and smoothest path. We present an algorithm for solving the mobile robot path planning problem in this article. Our research aims to develop a short, safe, and smooth path collision avoidance algorithm for mobile robot path planning in dynamic environments. The remainder of this paper is organized as follows: In section 2, related work is discussed. PP is discussed in section 3; it includes all aspects associated with it, including path encoding and PP objectives. Sections 4 and 5 describe two optimization algorithms, PSO and BA. In section 6, the proposed method is discussed along with the experimental environment, PSO-BA parameter settings, and scenarios. The results and conclusions are explained in sections 7 and 8.

2. RELATED WORK

A number of studies have been conducted in the past few years to address the PP problem. This study reviewed previous research that aimed at solving multiobjective optimization problems. Multi-objective grey wolf optimizer (MOGWO) [10] is introduced as an optimization algorithm for solving problems with multiple objectives. A fixed-size external archive is integrated into the GWO to store and retrieve the pareto optimal solutions, and then the social hierarchy is defined, and the hunting behaviour of gray wolves is simulated. As well, [11] presents a new method based on region of sight that is intended to simplify and speed up the process of path planning. As a result of this method, robot movement is minimized while meeting a variety of requirements, including safety, smoothness, and the shortest path length possible. Based on the natural behaviour of frogs, the shuffled frog-leaping algorithm (MOSFLA) was developed in [12]. Several factors are taken into account when selecting a path, including its safety, length, and smoothness. The results of the non-dominated sorting genetic algorithm II are compared with the results of the genetic algorithm. There was also a proposal in [13], for a multi-objective firefly algorithm (MO-FA). As a result of the method, three distinct goals are intended to be achieved: path safety, length, and smoothness. A test path based on eight realistic scenarios is also calculated for the MO-FA. Using a genetic algorithm to find the length of a path, the smoothness of the path, and the security of the path, an alternative algorithm can be employed [14].

A novel multi-objective optimization method based on the WOA is also presented for planning mobile robot paths [15]. During robot path planning, a robot's path should be minimized in terms of distance and smoothness. WOA determines the fitness of a solution based on the position of the target and obstacles in an environment. Authors achieved three objectives regarding [16] collision risk degree path length and smoothness. To improve the search capabilities of a PSO, this mechanism selects the most appropriate search strategies dynamically at various stages of the optimization process. Using multi-objective memetic algorithms (MOMA) [17], described a path planning algorithm for wheeled robots that optimizes several objectives at once, including path length, smoothness, and safety. Using conventional multi-objective genetic algorithms in combination with non-dominant elitist sorting and decomposition strategies, two MOMA are used to optimize path length while simultaneously reducing smoothness. MOMA implements new path encoding schemes, path refinements, and specific evolutionary operators to improve the search capability of the algorithms. On the basis of a Mars Rover scenario, the author minimized path difficulty, danger, elevation, and length from a starting point to a destination in [18]. Using multi-objective fuzzy optimization strategies and Voronoi diagrams to coordinate weight conflicts of sub-objective functions [19], developed a new approach to improve the cost performance index. The Pareto front was calculated using the intelligent water drops (IWD) algorithm, a multi-objective elitist approach based on coefficients of variation that have been proposed in [20]. The method is called CV-based MO-IWD, and it is aimed at optimizing both the length of the path as well as the safety of the path.

Hybridization is also used to make meta-heuristic algorithms for planning paths work better. Hybridization is the process of putting together two or more meta-heuristic algorithms to make a more powerful algorithm. A new hybrid approach based on enhanced genetic algorithms by modified search A* algorithm and fuzzy logic was proposed in [21]. Path optimization was carried out using PSO-modified frequency bat (PSO-MFB) algorithms [22]. The hybrid grey wolf optimizer-PSO algorithm was used in [23], to optimize a path. Geetha *et al.* [24] proposed a path planning algorithm based on ant colony optimization and genetic algorithms.

Planned paths have to meet several objectives, including length, smoothness, and safety. A hybrid multi-objective based on the bare-bones PSO with differential evolution was illustrated in [25]. In spite of this, most of these studies focus primarily on solving single or multiple objectives without accounting for changes in the location of the target. It is common for the location of the target to change over time. An example would be the flow target of unmanned aerial vehicles.

Based on the previous literature review, our present study contributes the following; i) PSO-BA is a hybrid path planning algorithm that combines PSO and bat algorithms in order to solve multi-objective optimization path planning problems with moving targets in dynamic environments; ii) We are aware of no other work that has addressed the PP problem algorithm in multi-objective robot path planning with moving targets in dynamic environments; iii) In order to produce accurate and efficient paths, the proposed PSO-BA takes into account a number of distinct objectives. These objectives include path safety, path length, and smoothness of the path.

3. PATH PLANNING

Path planning problem involves determining a path allowing a robot to move in a given environment from one point to another. In the following subsections, for the purpose of solving the PP problem, here, we'll talk about path encoding and how to manage objectives. In this study, the robot and the environment are considered as two-dimensional entities.

3.1. Path encoding

The paths in this work are handled as sorted lists of grid coordinates. Grid coordinates are pairs of (x, y) . In a list, the first element corresponds to the path's starting point, and the last element corresponds to the path's target point. The list can contain as many intermediary coordinates as necessary, which is variable. In this representation of the paths, the segments constitute the paths; therefore, two consecutive coordinates in the list correspond to a segment. An encoding of a path is shown in Figure 1.

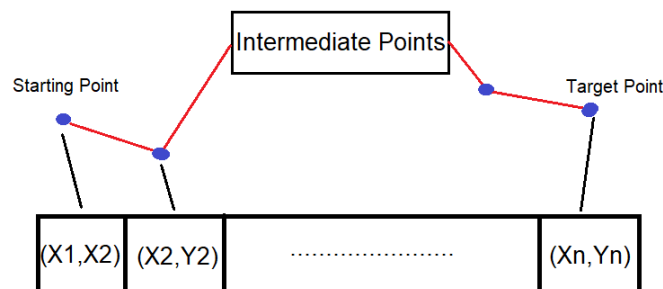


Figure 1. Path encoding

3.2. Path planning objectives

By addressing three different objectives, this study aims to ensure efficiency and accuracy. The objectives of these projects include the following: i) Path safety, ii) Path length, and iii) Path smoothness. In order to minimize all three objectives, the PSO-BA calculates the last two objectives in such a way that they are minimized together. These objectives are explained in the following subsections.

3.2.1. Path length

When transferring a robot from one location to another, the shortest path is usually the best option. The path length objective is designed to achieve the shortest path possible as much as possible. From a mathematical perspective, it represents the sum of the lengths of the segments of a path. The length of the segment is calculated by applying Euclidean distance between points. The Euclidean distance between two points, $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$, can be calculated in (1).

$$D(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (1)$$

The shortest path length (SPL) for a certain workout is found by adding up all of the accumulated distances between the exercise's origination point ($p(1)$) and its destination ($p(N)$).

$$SPL = \sum_{i=1}^{N-1} d(p_i(t), p_i(t + 1)) \tag{2}$$

According to the path planning algorithm, the index i represents the shortest solution.

3.2.2. Path smoothness

In most cases, the angle difference between the current position and the current goal is measured to determine how smooth the path is. If possible, the angle difference should be reduced as much as possible. At this angle, two lines intersect, allowing the robot's location to be determined at the end of the process. Energy consumption is directly related to this objective. Using (3), we can calculate the smoothness of a path. A graphical explanation of path smoothness is shown in Figure 2.

$$f_2(x, y) = \sum_{t=1}^{N-1} |\theta_{(p_i(t), p_i(t+1))} - \theta_{(p_i(t), p(N))}| \tag{3}$$

Where

$$\theta_{(p_i(t), p_i(t+1))} = \tan^{-1} \frac{y_{p_i(t+1)} - y_{p_i(t)}}{x_{p_i(t+1)} - x_{p_i(t)}} \quad \theta_{(p_i(t), p(N))} = \tan^{-1} \frac{y_{p(N)} - y_{p_i(t)}}{x_{p(N)} - x_{p_i(t)}}$$

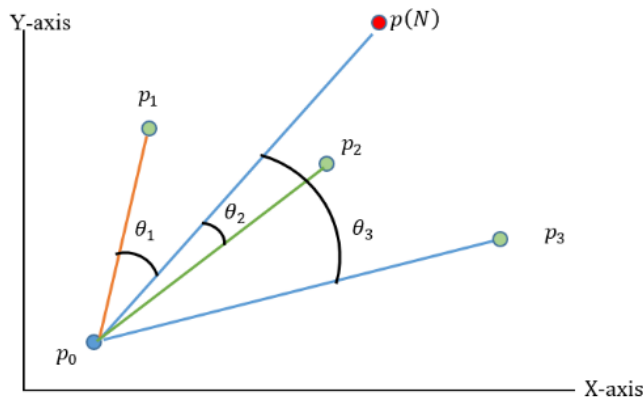


Figure 2. Path smoothness example

The fitness value is determined by combining the two objectives, weight being one of them,

$$f = (x, y) = w_1 f_1(x, y) + w_2 f_2(x, y) \tag{4}$$

Both values w_1 and w_2 represent relative importance weights for each objective. It is essential that their respective values meet the following requirements:

$$w_1 + w_2 = 1 \tag{5}$$

here is a definition of the overall fitness function.

$$fitness = \frac{1}{f(x,y)+e} \tag{6}$$

Typically, the e factor precludes division by zero (e = 0.001, for example). Each iteration can pick an optimal solution by balancing the two performance objectives indicated in (1), (2), and (3). Among the four contending locations depicted in Figure 3, p2 is optimal for the t iterations, and p3 is optimal for the (t + 1) iterations. However, for the second iteration (t + 2), the lengths for p1 and p4 are shorter, but the angles are bigger thus, p2 finds a compromise between the two requirements. Continue this approach until a GP is identified.

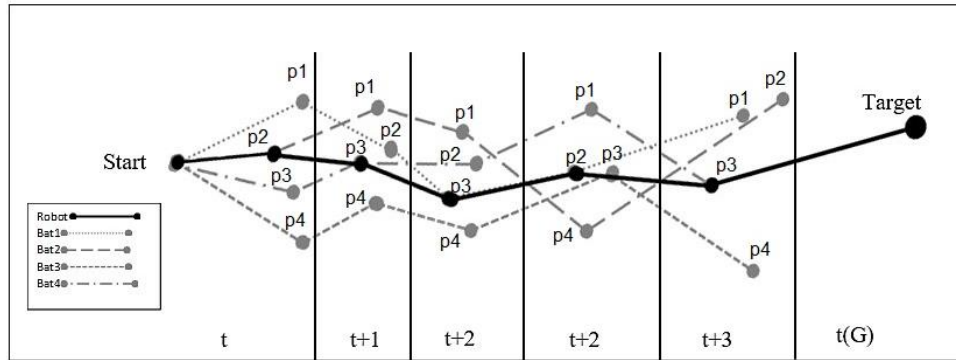


Figure 3. Path planning using point selection

4. PSO ALGORITHM

Eberhard and Kennedy [26] created a PSO algorithm in 1995 as a technique of simulating the behavior of social animals. When flying, birds follow the individual who is closest to the food, not their leaders. Through excellent communication among population members, the flock of birds obtains the required solution. In addition, PSO algorithms employ particles as possible solution representations, with each particle having its own position and velocity. Position refers to a particle's location, whereas velocity refers to its motion in relation to its location. The PSO builds the solution in two steps [27] based on random data (position and velocity). The velocity of each particle is updated according to (7).

$$v_i(t + 1) = v_i(t) + c_1 r_1 (cpbest_i - x_i) + c_2 r_2 (gbest - x_i) \quad (7)$$

A particle velocity is represented by $v_i(t)$, a particle position is represented by $x_i(t)$, an individual's best position is represented by $pbest$, a group's best position is represented by $gbest$, c_1 and c_2 represent the constant and r represents the random number $[0,1]$. Particle positions are updated by (8).

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (8)$$

5. BAT ALGORITHM

In the year 2010, Yang devised an algorithm that was based on biological principles and was given the name BA [28]. Microbats are able to detect their prey through the use of echolocation. Bats rely heavily on echolocation in order to navigate their environment. By turning sound pulses into acoustic signals, a bat is able to identify impediments while it is in flight. Their ability to identify impediments relies on the difference in timing between their two ears. A bat is able to estimate the speed, size, and form of prey and obstacles based on the volume of the response as well as the delay time between responses. Bats have the ability to modulate their sonar by emitting sound pulses of a high frequency. The bat is able to gather more specific information about its environment in a more expedient manner as a result of this [29].

5.1. Artificial bat movement

The bat's position is updated as follows:

$$f_i = f_{min} + (f_{max} - f_{min}) * \beta_i \quad (9)$$

$$v_i(t + 1) = v_i(t) + (x_i(t) - x^*)f_i \quad (10)$$

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \quad (11)$$

where x^* is the current best location (solution) in the world as determined by comparing all of the solutions across all of the individual bats in existence and β_i is a uniformly distributed random vector $[0, 1]$. Once all the bats have made their choices, the random walk concept is applied to generate a new strategy.

$$x_{new} = x_{old} + \epsilon A(t) \quad (12)$$

The variable ϵ here is a random number between -1 and 1, and it symbolizes the level of randomness as well as the path it takes. The average level of loudness for all of the bats at step t in an iteration is denoted by the value $A(t)$. After each iteration, it is important to do an update on the level of loudness as well as the pulse rate. According to these equations, when a bat locates its prey, the loudness of its call often reduces, but its pulse rate increases.

$$A_i(t + 1) = \alpha A_i(t) \tag{13}$$

$$r_i(t + 1) = r_i(0)[1 - \exp(-\gamma t)] \tag{14}$$

Where $0 < \alpha < 1$ and $\gamma > 0$.

6. PROPOSED METHOD

The proposed algorithm for mobile robot path planning is discussed in this subsection. A two-swarm optimization algorithm hybrid is used. The first is PSO and the second is BA. The following subsection provide details for each algorithm

6.1. Hybrid PSO-BA proposed

The best features of two or more optimization algorithms are combined to create a hybridized optimization algorithm. Because BA has limited exploration capabilities, convergence to the global optimum point is quite unlikely [29]. All dimensions of solutions are affected by the bat-specific factors A and r in BA. To achieve a balance between exploration and exploitation, a hybridization of PSO and BA is proposed. Alpha (α) and gamma (γ), are the variables that control these parameters, and their optimal value enhances loudness A_i and pulse rate variation r_i . Thus, Exploration and exploitation can be balanced with BA. The PSO algorithm adapts the appropriate range (α, γ). A pseudo code for PSO-BA is shown in Algorithm 1, while the overall procedure is shown in Figure 4. PSO particle size repressed in two-dimensional vector $x(\alpha, \gamma)$. Readers can also find PSO-MFA in Ajeil *et al.* [22]. We implemented PSO-BA with different algorithm and different environment.

Algorithm 1: PSO-BA algorithm pseudo-code

```

1-Initialise PSO and BA parameters:
   PSO_parameters: population size of particles NP, r1, r2, c1, c2
   BA_parameters: population size of bats NB, frequency (fi), pulse rate(ri) and loudness (Ai)
2-Generate random solutions of PSO s1= [α1, γ1], s1 = [α2, γ2 ]...., sn = [αn, γn ]
3-For i =1 to NP
4-Use BA algorithm from (9)-(14)
5-Pick the bat with the best fitness according to (6)
6-If the stopping criteria is not satisfied, then
7-Update position and velocity using (7)-(8) and go to step 3
8-Otherwise, Calculate the resulting output
End if
    
```

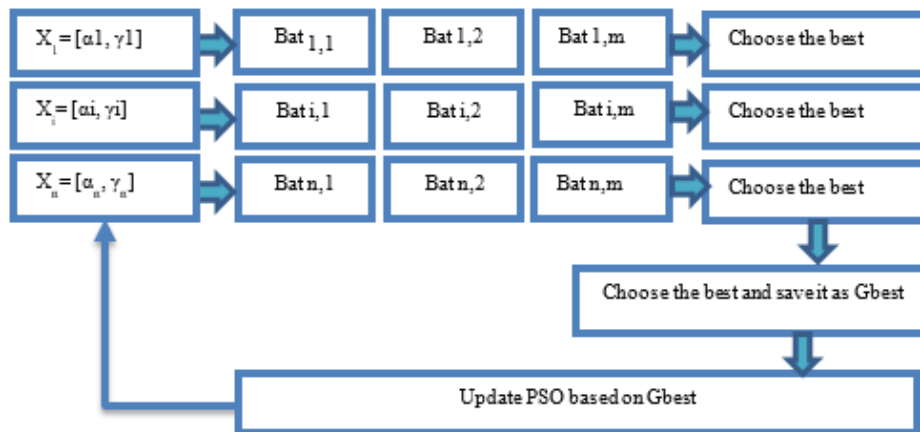


Figure 4. Proposed hybrid PSO-BA optimization algorithm

6.2. Experimental environment

An environment consists of several dynamic obstacles, which have different shapes, as well as six static obstacles, which have different dimensions and forms. A dynamic obstacle moves linearly in this environment. Obstacles that are dynamic are colored red, while obstacles that are static are colored black. Two robots are present in the environment. The first robot is represented by a blue point, while its target is represented by a purple point. The second robot is represented by a red point, and its target is the first robot. While the first robot is attempting to reach its goal, the second robot reaches the point at which the first robot will meet. The experimental environment is shown in Figure 5.

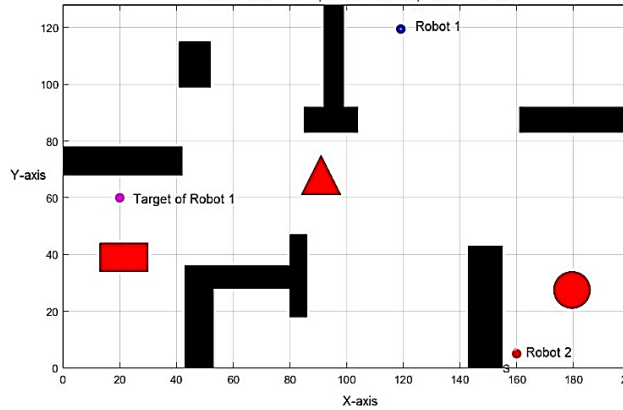


Figure 5. Experimental environment

6.3. PSO-BA parameter setting

Numerous experiments have been conducted in order to adjust the PSO-BA parameters. As a first step, several runs are run to determine which parameters are the most influential. The most influential parameter will be set up first, and so on. In each experiment, we conducted 31 independent runs, which was sufficient to ensure statistical significance. A set of tested values is selected for each parameter. This process is repeated for all parameters and algorithms. In Table 1, each parameter of each algorithm is listed.

Table 1. Parameter setting for hybrid PSO-BA

Parameter	Tested values	Selected Value
Population size PSO	{50,100,150,200,250 and 300}	200
Population size BA	{50,100,150,200,250 and 300}	250
C1	2,3,4,5,6,7,8,20,12	2
C2	2,3,4,5,6,7,8,20,12	2
fmin	0,1,2,3,4,5,6,7	1
fmax	10,11,12,13,14,15	14
generation	{100,120,150,200,300 and 600}	150

6.4. The scenarios

The PSO-BA algorithm is put through its paces with the use of a scenario in order to gauge its efficacy. Scenarios are used to define the environment, and the starting and ending points of the path is calculated. We employ eight sets of points that are dispersed in the maps of the surroundings in a way that corresponds to the path's extremes. This is why we employ eight hypothetical situations in our analysis. We've made sure that the pathways' termini are all pointed in the general directions indicated by the maps' coordinate systems. In Table 2, you can see where exactly each piece of the setting was placed.

Table 2. The scenarios

Characteristics	Scenario A	Scenario B	Scenario C	Scenario D
Starting Point of Robot no.1	(60,120)	(120,100)	(100,20)	(20, 20)
Target of Robot no.1	(20,60)	(40,60)	(60,110)	(20,110)
Starting Point of Robot no.2	(120,80)	(170,60)	(170,110)	(170,40)
	Scenario E	Scenario F	Scenario G	Scenario H
Starting Point of Robot no.1	(120,20)	(50,80)	(130,80)	(20,110)
Target of Robot no.1	(120,120)	(130,80)	(70,120)	(170,110)
Starting Point of Robot no.2	(20,100)	(100,30)	(100,10)	(10,10)

7. SIMULATION RESULTS

As a result of applying PSO-BA to a dynamic environment with a moving target, we present and analyze the results obtained. Based on the scenarios in Table 2, two algorithms (PSO-BA and BA) were run. Simulations are carried out using MATLAB R2020a. On a computer with a 2.80 GHz Core i7 processor and 16 GB of RAM, the MATLAB code is run. Based on the reference points used in this study, Table 3 shows the simulation results using BA and PSO-BA algorithms. Graphical solution for the scenario A as shown in Figure 6. Figure 6(a) illustrate graphical result of BA approach for scenario A and Figure 6(b) illustrate graphical result of PSO-BA approach for scenario A. Graphical solution for the scenario F as shown in Figure 7. Figure 7(a) illustrates graphical result of BA approach for scenario F and Figure 7(b) illustrate graphical result of PSO-BA approach for scenario F. PSO-BA always produces better results than BA for all scenarios, as can be seen in Table 3.

Table 3. Simulation result using BA and PSO-BA algorithm

Run no.	BA		PSO-BA	
	Path length	Path smoothness	Path length	Path smoothness
Scenario A	87	885	87	821
Scenario B	115	30	59	20
Scenario C	104	43	104	4
Scenario D	152	1871	151	1876
Scenario E	119	37	119	37
Scenario F	153	207	119	37
Scenario G	67	21	65	3
Scenario H	108	1835	107	1981

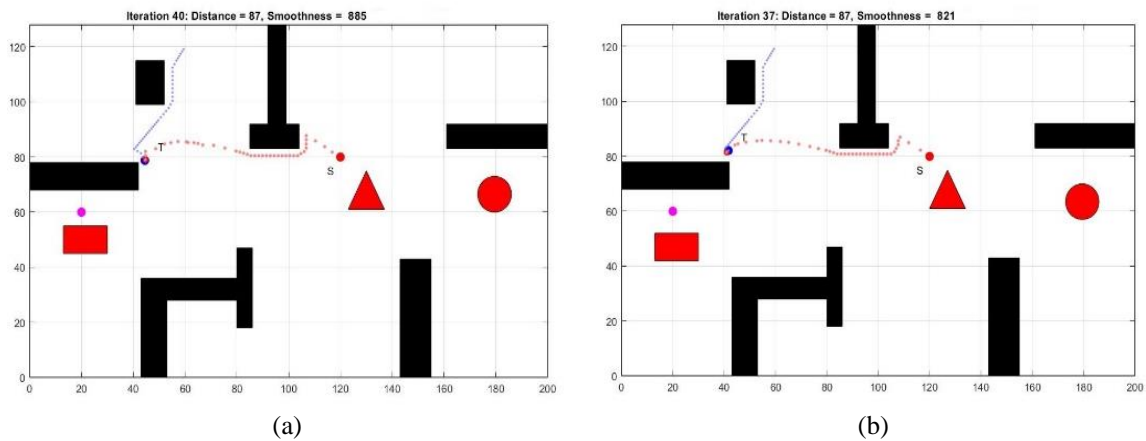


Figure 6. Graphical solution for the scenario A, (a) BA and (b) PSO-BA

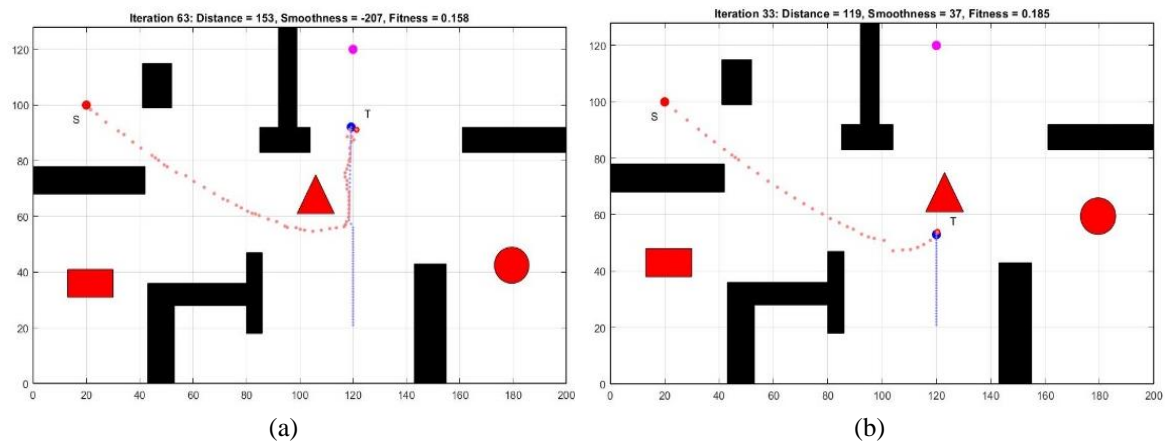


Figure 7. Graphical solution for the scenario F, (a) BA and (b) PSO-BA

8. CONCLUSION

A hybrid PSO-BA optimization algorithm was proposed for path planning for mobile robots in this study. PP is tackling one of the most challenging and significant problems in mobile robotics. BA generates points, while PSO optimizes two important parameters for bat to achieve balance between exploration and exploitation. This problem has three objectives: length of the path, safety of the path, and smoothness of the path. Various scenarios are used to test the algorithm's performance in dynamic environments with moving targets. The proposed algorithm has been demonstrated to be effective in avoiding both static and dynamic obstacles based on simulation results. PSO-BA always produces better results than BA.




REFERENCES

- [1] B. Shih, H. Chang, and C. Chen, "Path planning for autonomous robots – a comprehensive analysis by a greedy algorithm," *Journal of Vibration And Control*, vol. 20, no. 10, pp. 1603-1603, 2014, doi: 10.1177/1077546311429841.
- [2] S. Hosseinejad and C. Daddkhah, "Mobile robot path planning in dynamic environment based on cuckoo optimization algorithm," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, pp. 1–13, 2019, doi: 10.1177/1729881419839575.
- [3] N. M. Nakrani and M. M. Joshi, "An adaptive motion planning algorithm for obstacle avoidance in autonomous vehicle parking," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 3, pp. 687–697, 2021, doi: 10.11591/IJAI.V10.I3.PP687-697.
- [4] T. T. Mac, C. Copot, D. T. Tran, and R. De Keyser, "A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization," *Appl. Soft Comput. J.*, vol. 59, pp. 68–76, 2017, doi: 10.1016/j.asoc.2017.05.012.
- [5] H. S. Dewang, P. K. Mohanty, and S. Kundu, "A robust path planning for mobile robot using smart particle swarm optimization," *Procedia Comput. Sci.*, vol. 133, pp. 290–297, 2018, doi: 10.1016/j.procs.2018.07.036.
- [6] W. Wang, M. Cao, S. Ma, C. Ren, X. Zhu, and H. Lu, "Multi-robot odor source search based on Cuckoo search algorithm in ventilated indoor environment," *Proc. World Congr. Intell. Control Autom.*, vol. 2016-Sept, pp. 1496–1501, 2016, doi: 10.1109/WCICA.2016.7578817.
- [7] P. K. Das, S. K. Pradhan, S. N. Patro, and B. K. Balabantaray, "Artificial immune system based path planning of mobile robot," *Stud. Comput. Intell.*, vol. 395, pp. 195–207, 2012, doi: 10.1007/978-3-642-25507-6_17.
- [8] G. Wang, L. Guo, H. Duan, L. Liu, and H. Wang, "A bat algorithm with mutation for UCAV path planning," *Sci. World J.*, pp. 1-15, 2012, doi: 10.1100/2012/418946.
- [9] M. A. Hossain and I. Ferdous, "Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique," *Rob. Auton. Syst.*, vol. 64, pp. 137–141, 2015, doi: 10.1016/j.robot.2014.07.002.
- [10] S. Mirjalili, S. Saremi, and S. Mohammad, "Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization," *Expert Systems with Applications*, vol. 47, pp. 2015–2017, 2016.
- [11] H. Hliwa and B. Atieh, "Multi objective path planning in static environment using region of sight," *Proc. 2nd 2020 Int. Youth Conf. Radio Electron. Electr. Power Eng. REEPE 2020*, 2020, pp. 1–5, doi: 10.1109/REEPE49198.2020.9059199.
- [12] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "MOSFLA-MRPP: multi-objective shuffled frog-leaping algorithm applied to mobile robot path planning," *Eng. Appl. Artif. Intell.*, vol. 44, pp. 123–136, 2015, doi: 10.1016/j.engappai.2015.05.011.
- [13] A. Hidalgo-Paniagua, M. A. Vega-Rodríguez, J. Ferruz, and N. Pavón, "Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach," *Soft Comput.*, vol. 21, no. 4, pp. 949–964, 2017, doi: 10.1007/s00500-015-1825-z.
- [14] J. Hu and Q. Zhu, "Multi-objective mobile robot path planning based on improved genetic algorithm," *2010 Int. Conf. Intell. Comput. Technol. Autom. ICICTA 2010*, vol. 2, 2010, pp. 752–756, doi: 10.1109/ICICTA.2010.300.
- [15] T. K. Dao, T. S. Pan, and J. S. Pan, "A multi-objective optimal mobile robot path planning based on whale optimization algorithm," *Int. Conf. Signal Process. Proceedings, ICSP*, vol. 0, 2016, pp. 337–342, doi: 10.1109/ICSP.2016.7877851.
- [16] G. Li and W. Chou, "Path planning for mobile robot using self-adaptive learning particle swarm optimization," *Science China Information Sciences*, vol. 61, no. 5, pp. 1–18, 2018, doi: 10.1007/s11432-016-9115-2.
- [17] Z. Zhu, J. Xiao, J. Q. Li, F. Wang, and Q. Zhang, "Global path planning of wheeled robots using multi-objective memetic algorithms," *Integr. Comput. Aided. Eng.*, vol. 22, no. 4, pp. 387–404, 2015, doi: 10.3233/ICA-150498.
- [18] K. Jeddisaravi, R. J. Alitappeh, and F. G. Guimari, "Multi-objective mobile robot path planning based on A* search," In *2016 6th International Conference on Computer and Knowledge Engineering (ICCKE)*, 2016, pp. 7-12. IEEE, doi: 10.1109/ICCKE.2016.7802107.
- [19] Y. Wang, T. Wei, and X. Qu, "Study of multi-objective fuzzy optimization for path planning," *Chinese J. Aeronaut.*, vol. 25, no. 1, pp. 51–56, 2012, doi: 10.1016/S1000-9361(11)60361-0.
- [20] S. Salmanpour, H. Monfared, and H. Omranpour, "Solving robot path planning problem by using a new elitist multi-objective IWD algorithm based on coefficient of variation," *Soft Computing*, vol. 21, no. 11, pp. 3063–3079, 2017, doi: 10.1007/s00500-015-1991-z.
- [21] B. K. Oleiwi, H. Roth, and B. I. Kazem, "Modified genetic algorithm based on A* algorithm of multi objective optimization for path planning," *J. Autom. Control Eng.*, vol. 2, no. 4, pp. 357–362, 2014, doi: 10.12720/joace.2.4.357-362.
- [22] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," *Appl. Soft Comput. J.*, vol. 89, pp. 1–27, 2020, doi: 10.1016/j.asoc.2020.106076.
- [23] F. Gul, W. Rahiman, S. S. N. Alhady, A. Ali, I. Mir, and A. Jalil, "Meta-heuristic approach for solving multi-objective path planning for autonomous guided robot using PSO-GWO optimization algorithm with evolutionary programming," *J. Ambient Intell. Humaniz. Comput.*, vol. 12, no. 7, pp. 7873-7890, 2020, doi: 10.1007/s12652-020-02514-w.
- [24] S. Geetha, G. M. Chitra, and V. Jayalakshmi, "Multi objective mobile robot path planning based on hybrid algorithm," in *2011 3rd International Conference on Electronics Computer Technology*, Apr. 2011, vol. 6, pp. 251–255, doi: 10.1109/ICECTECH.2011.5942092.
- [25] J. H. Zhang, Y. Zhang, and Y. Zhou, "Path planning of mobile robot based on hybrid multi-objective bare bones particle swarm optimization with differential evolution," *IEEE Access*, vol. 6, pp. 44542–44555, 2018, doi: 10.1109/ACCESS.2018.2864188.
- [26] R. Eberhart and J. Kennedy, "New optimizer using particle swarm theory," *Proc. Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995, doi: 10.1109/mhs.1995.494215.
- [27] A. Koubaa, H. Bennaceur, I. Chaari, S. Trigui, and A. Ammar, *Robot Path Planning and Cooperation*, vol. XXII, no. 2. 2018.




- [28] X. S. Yang and A. H. Gandomi, "Bat algorithm: A novel approach for global engineering optimization," *Eng. Comput. (Swansea, Wales)*, vol. 29, no. 5, pp. 464–483, 2012, doi: 10.1108/02644401211235834.
- [29] S. Yilmaz, E. U. Kucuksille, and Y. Cengiz, "Modified bat algorithm," *Elektron. ir Elektrotechnika*, vol. 20, no. 2, pp. 71–78, 2014, doi: 10.5755/j01.eee.20.2.4762.

BIOGRAPHIES OF AUTHORS



Baraa M. Abed    is Ph.D. Student in the field of Robotics and Machine learning. He has received B.Sc. and M.Sc. degrees in Computer Science from University of Anbar. His research interests are autonomous systems, robotics, machine learning, classification and optimization techniques. He has published research papers at national and international journals, conference proceedings. He can be contacted at email: burasoft@gmail.com.



Wesam M. Jasim    received the B.Sc. and M.Sc. degrees in control automation engineering from University of Technology, Baghdad, Iraq, and Ph.D. degree in computing and Electronics from University of Essex, Essex, UK. Currently, he is an Assistant Professor with the College of Computer Science and Information Technology, University of Anbar. His current research interests include robotics, multiagent systems, cooperative control, Robust control, linear and nonlinear control, deep learning. He has published research papers at national and international journals, conference proceedings as well as chapters of book. He can be contacted at email: co.wesam.jasim@uoanbar.edu.iq.