

A Constraint Optimization path planning for Dynamic Environment

Nadia I. Khalil¹, Hadeel N. Abdullah², Layth A. Hassnawi³

¹Communication Engineering Department, University of Technology- Iraq, Baghdad, Iraq
Nadia.i.Khalil@uotechnology.edu.iq

²Electrical Engineering Department, University of Technology- Iraq, Baghdad, Iraq

³Research and Development Department, Ministry of Science and Technology, Baghdad, Iraq

Abstract. Optimal path planning is a major challenge for the operation of autonomous unmanned ground vehicles. In dynamic mobile path planning, the goal is to find a collision-free path for a robot from an initial position to a goal position in an environment with obstacles. However, in a dynamic environment, the challenge is bigger than static. In this paper, a modified ABC and an adaptive constraint PSO algorithm are proposed to deal with the dynamic environment; the algorithms were chosen to have a good balance between exploration and exploitation, with a penalty function used to have a collision-free path.

1. Introduction

Mobile robots are widely employed in many fields, including medicine, space exploration, emergency circumstances like fire hazards, etc. The robot should complete the tasks efficiently and effectively without any human interruption. The "Path Planning" concept has been proposed to address such a circumstance. In path planning, the robot must navigate a particular route in a given environment according to criteria such as distance, time, or energy, whether the environment is familiar to the robot. The robot must safely navigate these barriers without colliding with them to identify the best route from its starting point to its destination during navigation, for that path planning involves pre-calculating an obstacle-free route in which the controller guides the vehicle [1, 2].

Path planning aims to find the optimum, collision-free route between two points in a given environment. In general, there are several factors to consider while choosing the shortest, smoothest, least energy-consuming, or time-consuming path for the mobile robot to take to reach the target. Its goal is to find the path based on the shortest distance, path planning can be considered an optimization problem that can be solved without turning to a better solution since it is a multi-objective function. Much research has been introduced to solve the path planning problem [3,4]. There are many path-planning algorithms. However, these algorithms succeed, but every algorithm has a point where it fails in some cases, so algorithms cannot reach an ideal solution individually, especially in complex dynamic environments.

There are four steps to navigate a mobile robot [5]; 1) Perception: Using sensors, the robot generates information about the environment. 2) Localisation: Each iteration reminds the

robot of its position and orientation in the environment. 3) Cognition and Path planning: To reach at the goal point, the robot determines the path. 4) Motion control: The robot tracks its path by controlling the motion.[1, 5] .

Mohamed [6] suggested using an enhanced genetic algorithm (EGA) to determine the best route for a mobile robot. Auxiliary operations were developed to reduce the number of generations needed. The proposed method was better because it found a shorter path with less generations. Wang [7] introduced an adaptive learning-based hybrid PSO technique to prevent the early convergence of PSO on a variety of challenging problems. Abdulkareem [8] proposed using a combination of probabilistic roadmap (PRM), ACO, and third-order B-spline curve to solve path-planning. The outcomes of the suggested method guarantee a viable route in a complex and extremely complex environment. Li [9] developed a better PSO integration strategy based on better details that incorporates the learning factor, cubic spline interpolation, exponential attenuation inertia weight, and uniform distribution; it was improved from uniform initialization strategy to improve the later searchability and prevented the algorithm from falling into the local optimal solutions can control the independent variable as the inertia weight. Kamil [10] suggested a modified version of the artificial Bee Colony Algorithm (ABC), namely the Adaptive Dimension Limit- Artificial Bee Colony Algorithm (ADL-ABC), to determine the optimum global path for the mobile robot algorithm, which had a great potential to solve the proposed problem, the proposed algorithm performed better than the ABC algorithm. Mahmood [11] illustrated a creative method for path planning. The improved approach is created using the glowworm swarm optimization technique and adaptive dimensionality representation. The enhancement required adjusting the number of the current node's neighbors from which one will be chosen as the next current node in which the robot will move. Abhishek [12] proposed two hybrid meta-heuristic algorithms, one based on PSO with HSA and another based on PSO with GA for UAV three-dimension path planning in complicated obstacle-filled environments. The proposed hybrid algorithms gave a better-optimized path. Cheng [13] contributed significantly to the acceleration of convergence and acquisition of the shortest path of the PSO algorithm for robot path planning. Utilizing a new adaptive inertia weight, the GWO, chaos, and the conventional PSO.

1.1 Constraint Optimization

Constraint optimization issues involve a variety of design aspects. Constraints are crucial to engineering design since they are frequently imposed in problem statements and can be challenging to satisfy, making search efforts laborious and ineffective. Any stochastic search algorithm's freshly generated solution might not meet the restrictions of the problem after optimization. The penalty method, which penalizes impractical solutions in order to ensure feasible ones, is now the most widely used method for addressing the limitations of a problem. A bracket operator penalty term can be added to the objective function based on the constraint violation, and a penalized function is created.

Due to their straightforward structure and implementation, evolutionary algorithms (EAs) have been developed to handle constraint optimization and are having increasing interest. Due to the fact that the initial EAs are essentially unconstrained optimization techniques when managing constraint optimization, they are required to have an additional mechanism to deal with constraints. Several constraint-handling strategies have been proposed, one of them is the penalty function which is used in this reseaech.

1.2 The Objective Function

Since the task here is path planning, finding a path that is free of any obstacles is required. First, the robot is considered at the start point with coordinate (x_s, y_s) and the target point with coordinate (x_t, y_t) ; the algorithm is responsible for finding the line joining the two points with obstacle free. The constraint here is the minimization of the total path without any touch of any obstacle. The Euclidean distance is used to calculate the path length, so the minimum path should approach this distance so the path found should be compared with this distance:

$$d_1 = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2} \quad (1)$$

Where, d_1 is the path length between the source and the target.

The path length between the robot and the obstacle is found since it should take care of this distance and not to exceed it when going toward the obstacle.

$$d_2 = \sqrt{(x_r - x_o)^2 + (y_r - y_o)^2} \quad (2)$$

Where, d_2 is the path length between the source and the robot. x_o, y_o are the obstacle coordinates and x_r, y_r are the robot coordinates; this distance should be calculated in every iteration.

For a robot to move from its position point to the target point in the existence of an obstacle in this path, it should pass in an intermediate point, which is away from the obstacle, as shown in Figure (1). The big circle shown in Figure (1) represents an obstacle.

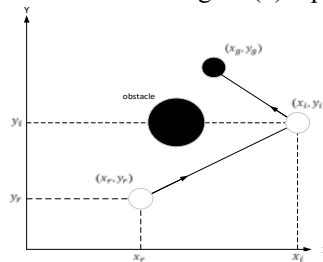


Fig. 1. Movements of the robot

The objective function, which is the path length here, is found by Equation (3):

$$D = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} + \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2} \quad (3)$$

The original objective function deals just with static obstacles. To deal with static and dynamic obstacles, the function should be changed to handle moving obstacles and not collide with the robot. Thus, a penalty function is added to the objective function, and the optimization is changed to the constraint optimization shown in Equation 4:

$$D = \sqrt{(x_r - x_i)^2 + (y_r - y_i)^2} + \sqrt{(x_i - x_g)^2 + (y_i - y_g)^2} + \text{penalty} \quad (4)$$

Usually, the penalty function is taken as a positive value. The penalty function here depends on the obstacle that lies in the path, and the size and shape of the obstacle should be taken into consideration then; choosing the penalty value is chosen so that the penalty value is added to the circumference of the obstacle so that the robot can avoid the obstacle. The robot may come closer to the obstacle but does not collide.

2. The Proposed Algorithms

In order to have an algorithm that deals with dynamic environments, constraint optimization is used with two different algorithms first one is the modified ABC algorithm, and the second is the Adaptive Constraint PSO

2.1 Modified ABC algorithm

The ABC algorithm was created for unconstrained optimization problems. Since ABC is expanded to find the practical ideal answer to an issue, it should address two difficulties, which are the balance between infeasible and feasible solutions and the balance between diversity and convergence.

We modified the initialization and scout phases in the Modified ABC algorithm. Only the first initialization of food sources is entirely random during the startup phase. Other startup phases start with the food source from the previous algorithm run with the highest fitness value as the first new food source. So that the algorithm runs are not entirely independent from one another. Therefore, exploitation of the good sources can be increased, which makes the values closer to the good or efficient solutions. In order to increase the exploitation, the scout bee’s phase was changed. In the scouting phase, the algorithm checks every possible solution. If the solution is not feasible, the food source is replaced with a new randomly produced solution (the random values). A penalty function for constraint optimization is used, enabling the algorithm to deal with moving obstacles. Figure (2) shows the flowchart of the modified ABC algorithm.

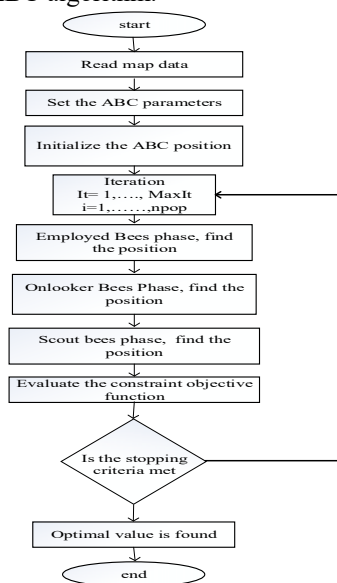


Fig. 2. Flowchart of the modified ABC algorithm.

2.2 Adaptive Constraint PSO

The PSO is a population-based algorithm. In the original PSO, at each iteration, the swarm of particles enters the objective function, and personal and social characteristics are compared to the fitness function to achieve a high convergence rate. Each iteration updates the velocity of each particle. The velocity and position of each particle are updated according to the equations (5) and (6).

$$[v(i, j)]_a^{b+1} = [w * v(i, j) + c_1 * rand() * (partb(i, j) - x(i, j) ... + c_2 * rand() (globb(i, j)x(i, j))]_a^b \tag{5}$$

$$[x(i, j)]_a^{b+1} = [x(i, j)]_a^b + [v(i, j)]_a^{b+1} \tag{6}$$

Adaptive means that the algorithm can accept the change for any parameter in the velocity of Equation (5), such as the inertia weight or the acceleration parameters. Our

modification is based on changing the velocity continuously in each iteration, and the position is changed accordingly since the inertia weight's big value influences the c_2 term. In contrast, the c_1 term is affected by the inertia weight's small value. The PSO algorithm's search capabilities are automatically increased by continuously changing the inertia weight. The inertia controls the velocity and balances the global and local search. It can be varied in many ways, but generally, the inertia value needs to be large at the beginning to enhance the global search and then decrease to enhance the local search. In particular, many ways to change the inertia exist, a linearly decreasing equation as follows :

$$w(t) = wmax - \frac{wmax-wmin}{itmax} * it \tag{7}$$

Where $w(t)$ is the inertia according to the iteration reached, $wmax$ is the maximum value of inertia chosen, $wmin$ is the minimum value of inertia chosen, $itmax$ is the maximum number of iterations, and (it) is the value of iterations reached. The flowchart of the algorithm is shown below.

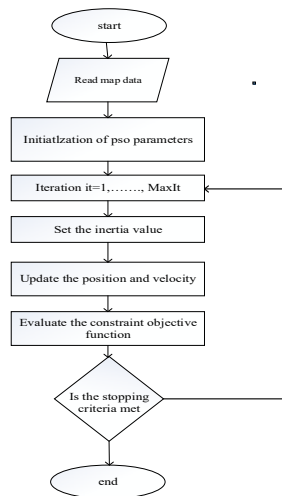


Fig. 3. Flowchart of the adaptive constraint PSO.

The penalty function here is a positive integer; since the objective function is a distance, the penalty function is a distance. This distance is added in both algorithms to make the path safer for the robot so that it does not collide with any obstacle.

3. Results

In order to deal with a constraint problem, the modified ABC is used since the problem here is a constraint one. ABC has a slow range of convergence since ABC's dependency on random values, even when it gets into the iteration, makes it far from the sufficient solutions and needs more time to reach the sufficient one, that the position used inside the iteration is altered in order to converge the algorithm. To make a step of convergence for the algorithm, the scout bees' position is initialized as the position of the last run inside the iteration.

The PSO algorithm's search capabilities are automatically increased for the adaptive constraint PSO by continuously changing the inertia weight. The inertia controls the velocity and balances the global and local search; it can be varied in many ways. In general, the inertia value needs to be large at the beginning to enhance the global search and needs to be decreased to enhance the local search.

The two algorithms were successful in path planning in all cases looking. The results in Table (1) show that the modified ABC has a standard deviation value that can't be dismissed,

which means the algorithm solution is ranging during the iteration in a big space of solutions, even if it was successful but still have a slow convergence compared to PSO.

The cost for the adaptive constraint PSO was less than the modified ABC. The PSO could deal better with the environment; it gave a path length between 1 and 3 segments in all cases.

Table 1. modified ABC and adaptive constraint PSO results.

Cases	Obstacles centres	modified ABC				Adaptive constraint PSO			
		cost	time	Average length	Standard Deviation	cost	Time	Average length	Standard Deviation
Case1	(2,3)	50.709	30.850	230.53	13.634	7.99	169.132	8.1682	0.6241
Case2	(1,2) (3,4)	152.688	30.737	156.62	8.8432	19.58	80.54	19.93	0.8458
Case3	(2,4) (1,1.5) (2,3) (3,2) (4.5,4)	144.053	37.15	192.25	11.109	12.36	50.6	12.337	0.4740

Figure (4) shows the paths taken by the algorithms in all cases; it can be noticed from the figure that the modified ABC took longer paths in all cases.

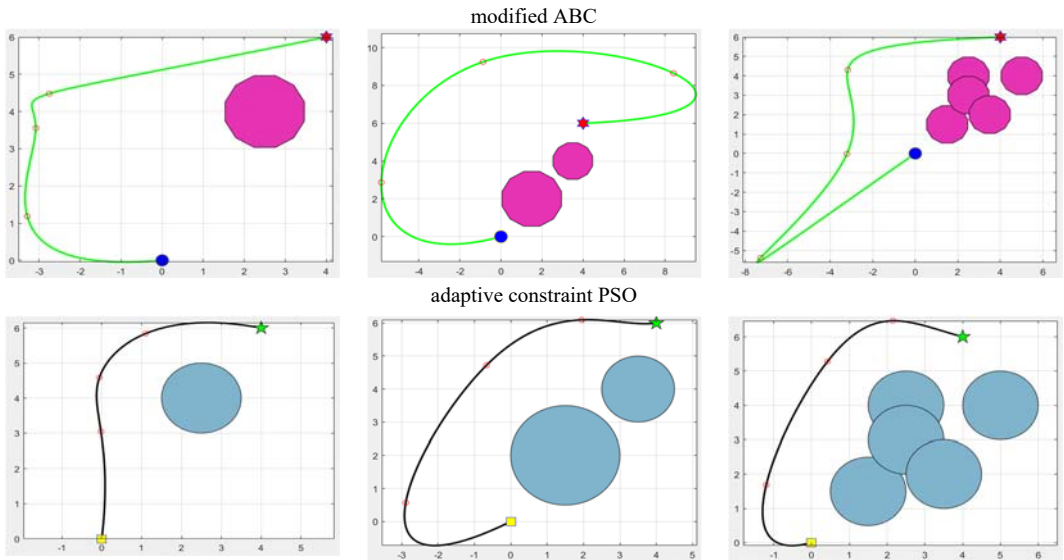


Fig. 4. Path planning for one, two, and five obstacles.

Figure (5) shows the average cost for both algorithms; the first row is for the modified ABC, the second one is for adaptive constraint PSO, the first column is for one obstacle, the second one is for two obstacles, and the third is for five, From results the modified ABC needs a high iteration to reach the optimal path. It still has a slow range of convergence.

modified ABC

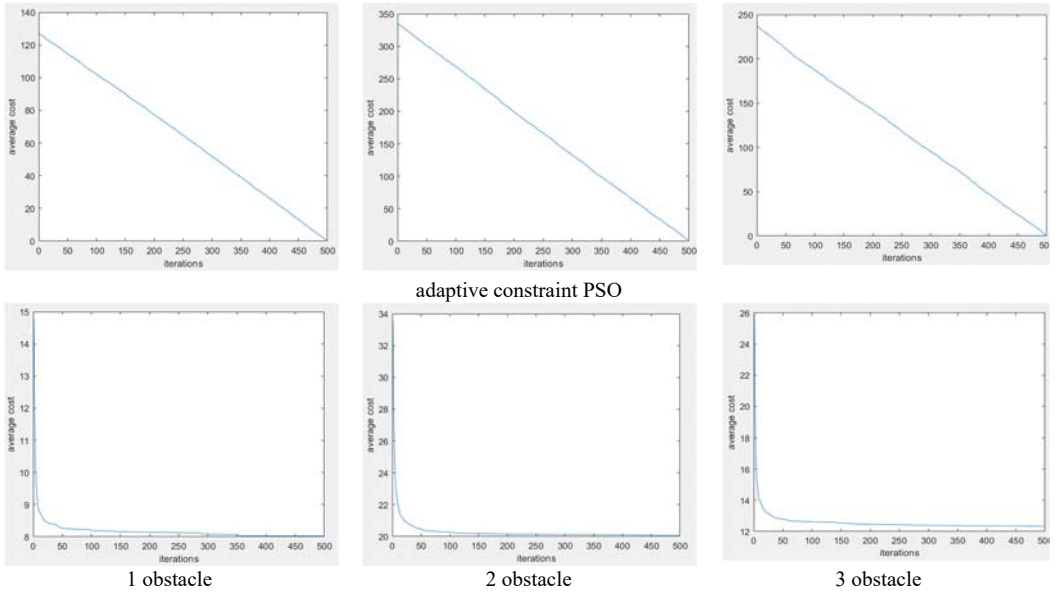


Fig. 5. The average cost.

In order to find the velocity that the algorithm can deal with, the size of the obstacle should be considered. The obstacle radius R_{obs} with the distance moved in a certain time d_{mov} , the ratio of the distance moved to the radius of the obstacle is taken as a reference.

$$\Delta d = \frac{d_{mov}}{R_{obs}} \quad (8)$$

Where Δd is the distance factor, it is the distance moved relative to the radius of the obstacle; this factor is unitless, it indicates the change of obstacle position in a certain time.

Table 2. Results of proposed path planning methods.

Δd	modified ABC	adaptive constraint PSO
≥ 1	fail	fail
0.5	fail	fail
0.25	fail	fail
0.125	success	success
0.1	success	success

The algorithms can succeed in dynamic environments with obstacles with a limited velocity since the two algorithms deal with static cases, but the modification improved the algorithms, and the penalty function gave them the ability to deal with moving obstacles.

4. Conclusions

Path planning in a dynamic environment needs a suitable path planner to determine how to move with no collision; it needs a special optimization to have the shortest possible safe path length in a short time. This paper proposes constraint optimization for path planning in dynamic environments. ABC and PSO are designed to deal with static environments where a modification was done. For ABC, the exploitation of the sound sources was increased to decrease the time needed, but for PSO, an adaptive version was used, and a penalty function was used to make the algorithms able not to collide with obstacles. The results show that the two algorithms succeed in path planning but differ. The path was shorter in PSO, with a time

needed for processing almost convergent to that for ABC, but the number of iterations needed to reach the optimized solution is less for PSO. Finally, both algorithms have a limit for the velocity of obstacles that they can deal with.

1 References

- [1] H. S. Dewang and P. K. Mohanty, “A Robust Path Planning For Mobile Robot Using Smart Particle Swarm Optimization“, *Procedia Computer Science* 133, 290–297, (2018). <https://doi.org/10.1016/j.procs.2018.07.036>
- [2] B.S Shihab., H.N. Abdullah and L.A Hassnawi, “Improved Artificial Bee Colony Algorithm-based Path Planning of Unmanned Aerial Vehicle Using Late Acceptance Hill Climbing“, *International Journal of Intelligent Engineering and Systems*, Vol.15, No.6, (2022). <https://doi.org/10.22266/ijies2022.1231.39>
- [3] N.I. Khalil, H.N. Abdullah, and L.A. Hassnawi, *A Hybrid ABC-PSO Algorithm for Optimal Robotic Path Planner*, In 2023 15th International Conference on Developments in eSystems Engineering (DeSE), (pp 27-32). 09-12 January 2023, Baghdad, Iraq, (2023). <https://doi.org/10.1109/DeSE58274.2023.10100021>
- [4] B. S. Shihab, H.N. Abdullah and L.A. Hassnawi, “Obstacle Avoidance and Path Planning for UAV Using Laguerre Polynomial”, *International Journal of Intelligent Engineering and Systems*, Vol. 15, No. 6, (2022). <https://doi.org/10.22266/ijies2022.1231.58>
- [5] P. Raja, and S. Pugazhenth, “Path planning for a mobile robot in dynamic environments”, *International Journal of the Physical Sciences*, Vol. 6, No. 20, pp. 4721-4731, 23 September, (2011). <https://doi.org/10.5897/IJPS11.573>
- [6] M. J. Mohamed and M. W. Abbas., “Enhanced GA for Mobile Robot Path Planning Based on Links among Distributed Nodes“, *Engineering and Technology Journal*, Vol 31(1A), pp.26-41, (2013). <https://doi.org/10.30684/etj.2013.71193>
- [7] F. Wang, H. Zhang, K. Li, Z. Lin, J. Yang and X. Shen, “A Hybrid Particle Swarm Optimization Algorithm Using Adaptive Learning Strategy”, *Information Sciences* pp.162-177, (2018). <https://doi.org/10.1016/j.ins.2018.01.027>
- [8] M.I. Abdulkareem and F.A. Raheem., “Development of path planning algorithm using probabilistic roadmap based on ant colony optimization”, *Engineering and Technology Journal*, Vol. 38, Part A, No. 03, pp. 343-351, (2020). <https://doi.org/10.30684/etj.v38i3A.389>
- [9] X. Li, D. Wu, J. He, M. Bashir and M. Liping, “An Improved Method of Particle Swarm Optimization for Path Planning of Mobile Robot”, *Journal of Control Science and Engineering*, (2020). <https://doi.org/10.1155/2020/3857894>
- [10] R. T Kamil., M. J. Mohamed and B. K. Oleiwi., “Path planning of mobile robot using improved artificial bee colony algorithm”, *Engineering and Technology Journal*, Vol. 38, No. 9, pp.1384-1395, (2020). <https://doi.org/10.30684/etj.v38i9A.1100>
- [11] Q. R. Mahmood, A. H. Hasan and H. K. Khafaji, “Robot Path Planning Based on Hybrid Adaptive Dimensionality Representation with Glowworm Swarm Optimization”, In 2021 4th International Iraqi Conference on Engineering Technology and Their Applications (IICETA) (pp. 241-246). 21-22 September 2021, Najaf, Iraq. <https://doi.org/10.1109/IICETA51758.2021.9717626>
- [12] B. Abhishek, S. Ranjit, T. Shankar, G. Eappen, P. Sivasankar and A. Rajesh, “Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs”, *SN Applied Sciences* (2020) 2:1805. <https://doi.org/10.1007/s42452-020-03498-0>

- [13] X. Cheng, J. Li, C. Zheng, J. Zhang. and M. Zhao, “An Improved PSO-GWO Algorithm With Chaos and Adaptive Inertial Weight for Robot Path Planning“, *Frontiers in neurorobotics*, 15, p.770361, (2021). <https://doi.org/10.3389/fnbot.2021.770361>
- [14] M. R. Mohan and A. Sundaram, “A PSO Based Hybrid Algorithm with Adaptive Penalty Function Approach for combined Heat and Power Economic Dispatch Problem”, *Archives Des Sciences*, Vol. 66, No. 4, (2013).