Vol. 10(35), Jan. 2020, PP. 4483-4492

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

A Path Planning Technique for Autonomous Mobile Robot

Zaharuddeen Haruna, Umar Musa, Mohammed B. Mu'azu and Abubakar Umar

Department of Computer Engineering, Ahmadu Bello University, Zaria, Nigeria

***Phone Number:** +2347036895532*

***Corresponding Author's E-mail:** hzaharuddeen@abu.edu.ng*

Abstract

This research aimed at development of a dynamic path planning technique for autonomous mobile robot using a modified bat algorithm. Autonomous mobile robots are programmable and mechanical device with the ability of moving from one location (called the source location) to another position (known as the target location) in an environment containing obstacles without human intervention. Thus, for a mobile robot to be autonomous, it has to be intelligent enough in perceiving the environment so as to acquire information in the environment and make decision based on it. Therefore, path planning becomes essential for the autonomous mobile robot to reach its target location. To achieve this, an objective function was modelled in form of distance function using the coordinates of the source and target locations. A path planning technique was then developed using modified bat algorithm that optimized the objective function to generate an optimal collision free motion path for the autonomous mobile robot. The performance of the developed algorithm was determined by implementing in an unknown static environment under different complexities of obstacles. The simulation result obtained showed that the path planning algorithm was effective for the control of autonomous mobile robot as it generated an optimal path without colliding with obstacles in different environment under different complexities as compared to results obtained using bat algorithm and ant colony optimization algorithm.

Keywords: *path planning, autonomous mobile robot, modified bat algorithm, obstacles, objective function*

1. Introduction

Autonomous mobile robots (AMR) are programmable and mechanical device with the ability of moving from one location to another in an environment containing obstacles without human intervention. They use sensors to perceive the environment in order to acquire different information in the environment. These information are in the form of obstacles location, robot start location and target destination [1]. AMR has been applied in different areas ranging from surveillance, transportation, medicine, industries, military and hazardous environment. For an AMR to be able to execute different type of task in these areas of applications, it has to be intelligent in making decisions with respect to its movement. Path planning is defined as computation of an optimal collision free motion path for AMR to reach its target destination from the source location in a static environment (the environment is populated by only static obstacles) or dynamic environment (the environment is populated by static and moving obstacles). The optimal path is always determined based on some optimality criteria as shortest distance, minimum time, minimum cost and minimum energy where shortest distance and minimum time is the most commonly used criteria [1]. Path planning problem comprises of obtaining a sequence of moves for repositioning an AMR from an initial source position to a given target destination in a static or dynamic environment, and the AMR must avoid obstacles in the environment. This problem

Article History:
Received Date: Apr. 14, 2019
Accepted Date: Jun. 11, 2019
Available Online: Jan. 01, 2020

can be solved by the combination of global (offline) and local (online) path planning [2]. Global path planning of AMR is performed in an environment where whole information about static obstacles and route of moving obstacles are available prior to the movement of the AMR. As such, the AMR has to only compute the optimal collision free path once at the start and then track the planned path to reach the target destination. However, if whole information of the environment is not known prior to the movement of the AMR, information about the environment are acquires through sensors as the AMR moves towards the target destination in the environment. This is called local path planning [3].

In recent years, researchers have implemented different techniques to address path planning problem for AMR. Nature inspired metaheuristic algorithms haven used for path planning of AMR. These algorithms are inspired by mimicking the features of natural or artificial systems [4]. For example, genetic algorithm was developed by mimicking the Darwinian law of survival of the fittest [5], particle swarm optimization was developed by mimicking the behavior of fish schooling or social behavior of bird flocking [6], ant colony optimization was developed by mimicking the foraging behavior of ant colonies [7], Cat Swarm Algorithm was developed by mimicking the observing behavior of cats as they hunt their prey [8], biogeography-based optimization was developed by mimicking the migration behavior of island species [9], Cuckoo Search Algorithm was developed by mimicking the obligate brood parasitism behavior of cuckoo species [10] and Bat Algorithm, BA was developed by mimicking the echolocation characteristics of bats [11]. Figure 1 presents the path planning algorithms used in literature.

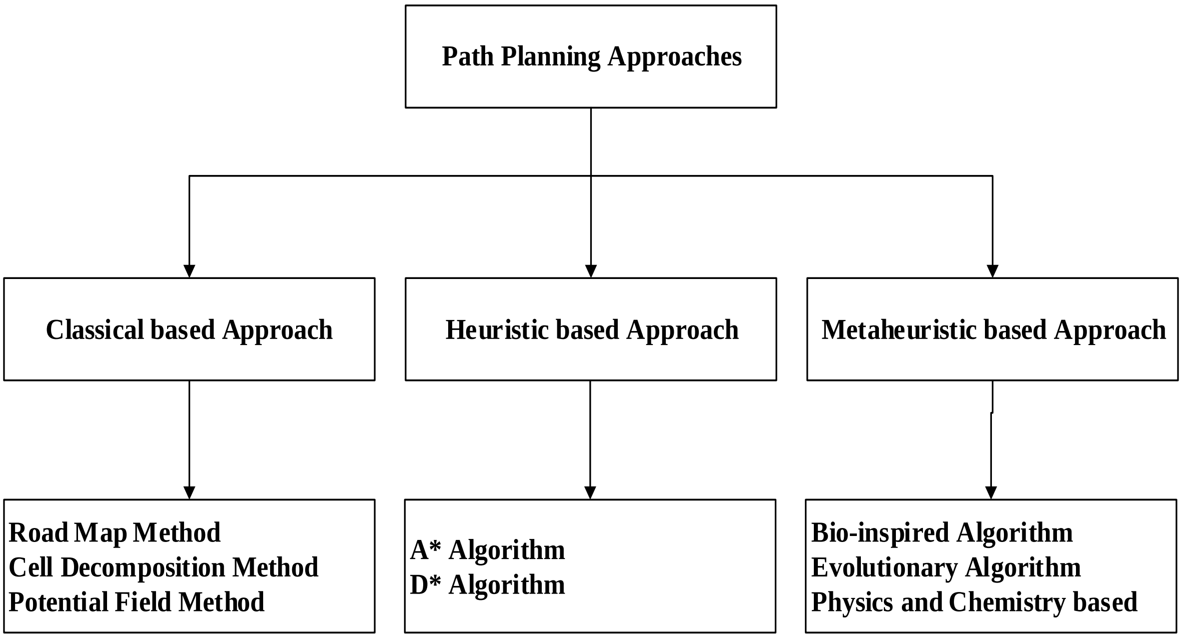

Figure 1: Path Planning Approaches

This research uses the developed modified Bat Algorithm by Haruna, Mu'azu [12] to solve path planning problem of AMR. The BA is metaheuristic search algorithm that that mimic the echolocation behavior of bats. For navigation, bats uses echolocation to determine a collision free path while searching for food. Thus, the aim of bat's echolocation is to serve as hunting strategy [13]. Research has shown that the BA can have better performance over GA and PSO [11], and it can address real world and engineering optimization problems [14, 15].

2. Modified Bat Algorithm

Bat algorithm is a nature inspired metaheuristic search algorithm that works based on the echolocation characteristic of bats. Bats emits a sound pulse called sonar with high loudness and listened to the reflected sound (i.e. echo) that falls upon obstacles within search space. While navigating during the searching for food, bats uses the difference in time between emitted and reflected sound to determine an obstacles free path and distance. They use variations of Doppler Effect to differentiate targets [11]. The following are rules considered for the implementation of BA:

a. Bats sense distance, and also differentiate between food and background barriers using echolocation;

b. Bats automatically adjust the frequency (or wavelength) of their emitted pulses and depending on the proximity of the target, they adjust the rate of pulse emission;

c. When searching for prey, the loudness varies to a large value and when homing with prey, the loudness varies to a minimum constant value.

However, due to the fact that the bat algorithm at times may get trapped into local minima as a result of its high rate of exploitation, Haruna, Mu'azu [12] developed a modified bat algorithm by incorporating elite opposition based learning at the initialization phase of to increase the diversity of the solution search space. This enables the algorithm to avoid being trapped in local minima. Inertia weight was also integrated into the bat algorithm at the local search phase so as to improve the intensification capability of the algorithm by ensuring balance between diversification and intensification during the algorithm's search process.

Equations (6) is the mathematical implementation of elite opposition based learning while equation (7) is use for generating the dynamic search boundaries

The modified bat algorithm is implemented using the following steps:

Step I: Initialize the bats parameters: bat population, initial position x, initial velocity (v), frequency $(Q_{\max})$ and $(Q_{\min})$, loudness (l), pulse emission rate (r) are defined.

Step II: select the best candidate as an elite candidate (x_e) from the initial solution (x)

Step III: update the search boundaries and generate the opposite solutions

$$\tilde{x} = \delta(\mathrm{p}_i + q_i) - x_e \tag{1}$$

Where $i = 1, 2, ..., \mathrm{m}$ and $j = 1, 2, ..., \mathrm{D}$, m is the bat population, D is the search dimension, $\delta \in U(0,1)$, (p_j, q_j) is the dynamic search bound and can be calculated as [12]:

$$p_j = \min\left(x_{i,j}\right) \text{ and } q_j = \max\left(x_{i,j}\right) \tag{2}$$

Step IV: Generate new solution by updating Q, v and x in the equations below:

$$Q_i = Q_{\min} + \left(Q_{\max} - Q_{\min}\right) * \beta(0,1) \tag{3}$$

$$v_i^n = v_i^{n-1} + \left(x_i^t - x_{best}\right) * Q_i \tag{4}$$

$$x_i^n = x_i^{n-1} + v_i^n \tag{5}$$

Step V: Perform local search by choosing a solution among the best solution based on the condition $rand > r$. A local solution is then generated around the best solution using equation (4)

$$x_{new} = x_{old} + \varepsilon\lambda l^n \tag{6}$$

Where λ is a linear valued of inertia weight defined as [12].

$$\lambda = \left(\frac{itr_{\max} - itr}{itr_{\max}}\right)^m \tag{7}$$

Step VI: Generate new solution by flying randomly using equation (8)

$$\left(rand < l \text{ and } obj\left(x_i\right) < obj\left(x_{best}\right)\right) \tag{8}$$

The flowchart of the modified bat algorithm is given in Figure 2 [12]:

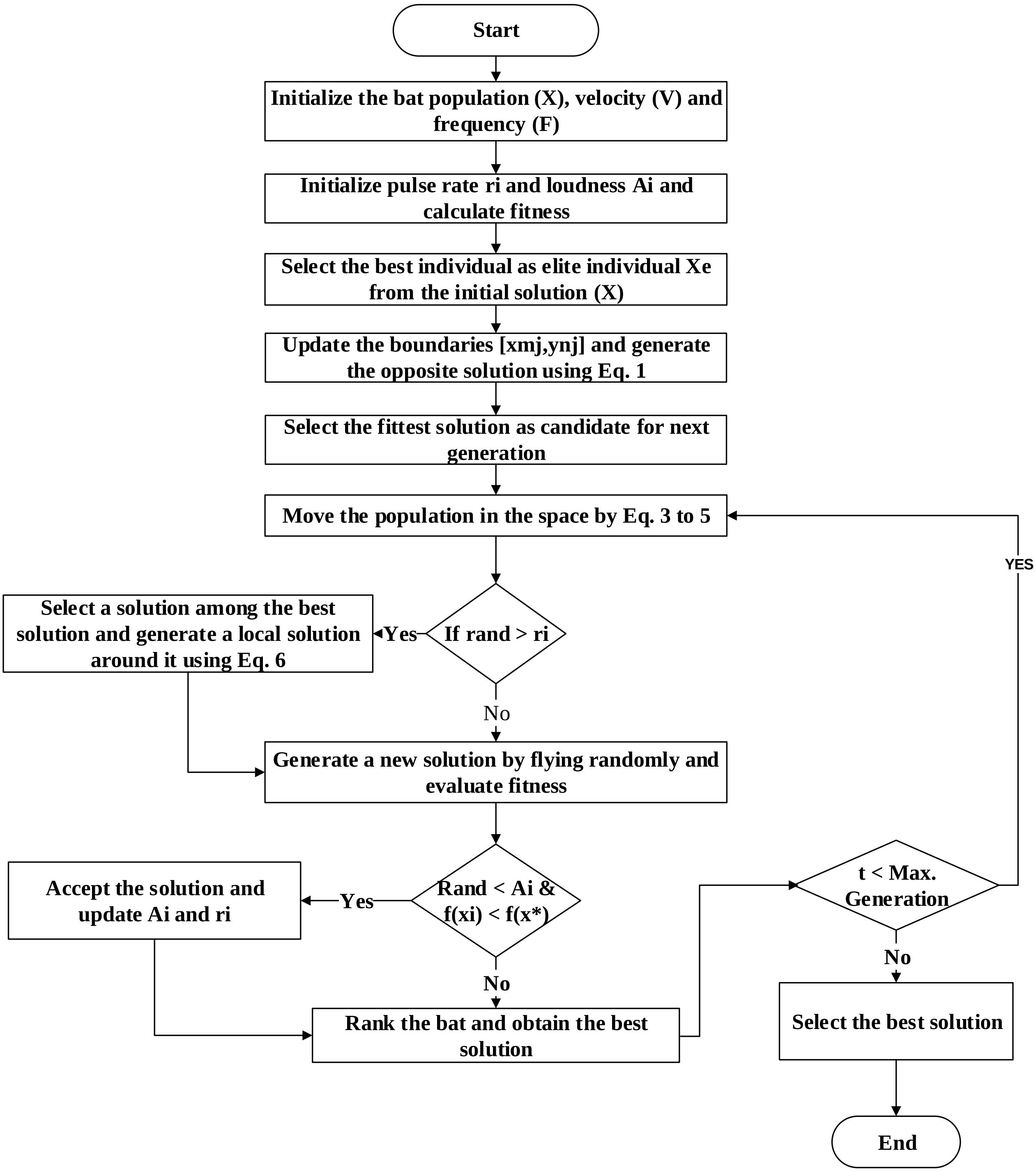

Figure 2: Flowchart of modified BA [12]

The objective function to be minimized by the modified BA is formulated as a function of the coordinates of the source and target locations. This function is an unconstrained optimization function and is given in equation (9):

$$\min \quad dist = \sqrt{(x_t - x_1)^2 + (y_t - y_1)^2} \tag{9}$$

Where 'min' means the path with minimum distance to the target location while (x_1, y_1) and (x_t, y_t) are respectively the coordinates of the source and target locations.

3. Path Planning in Unknown Static Environment

In this section, a modified BA which is inspired by the hunting behavior of bats (i.e. echolocation) is developed for on-line path planning of AMR in an unknown static environment. The problem to solve here is the path planning problem, which is the determination of an optimal collision free path for AMR to move from source position with coordinate (x_1, y_1) to a target destination with coordinate (x_t, y_t) in an unknown static environment populated with obstacles. The objective of this problem is for the AMR to navigate to the target in a minimum time via a shortest path without making contact with the obstacles in the environment. The position of the obstacles is acquired by the AMR through sensor, then the modified bat algorithm provides the optimal path for the AMR to reach its target by minimizing the objective function. However, during the search process of optimal path by the algorithm, the positions of the global best bat in each iteration are selected while the AMR move to these positions in sequence. Figure 3 presents the simulation environment populated with randomly generated obstacles.

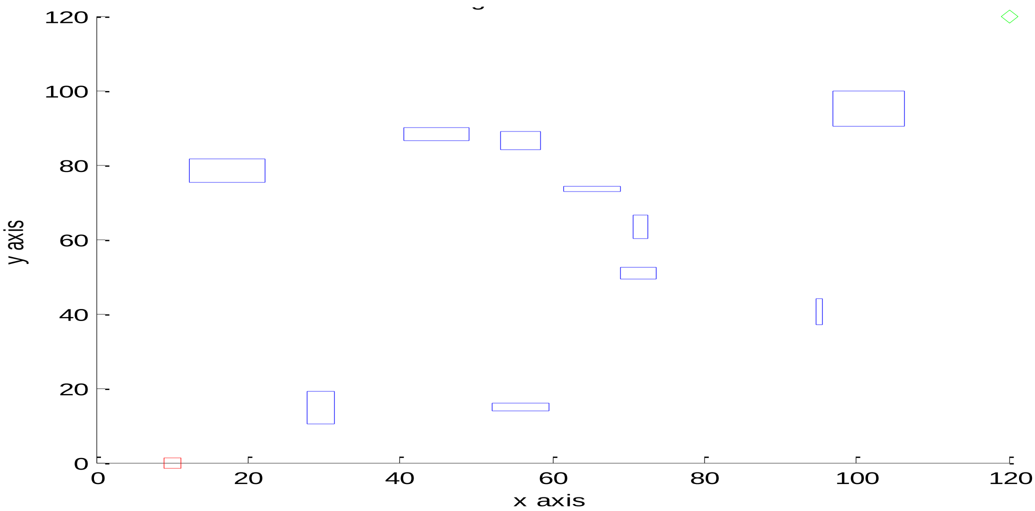

Figure 3: Simulation Environment Containing Obstacles

From Figure 3, the red square with coordinate (10,0) represent the AMR source position while the green diamond with coordinate (120,120) represent target destination of the AMR. The objective of the AMR is to reach the target destination in an unknown static environment containing obstacles via a collision free optimal path using the developed modified bat algorithm.

As the AMR move towards the target location in the environment populated by obstacles, the position of the best bat to be selected should be based on the path with minimum distance between the source and the target locations as computed using the objective function.

The direction of movement of the AMR α, is determine using equation (10):

$$\alpha = a\tan\left(\frac{|(y_t - y_1)|}{|(x_t - x_1)|}\right) \tag{10}$$

As the direction of movement of the AMR is computed, AMR moves on incremental basis on both horizontal direction (x-axis) and vertical direction (y-axis) using equations (11) and (12) respectively so as to update the position of the next coordinates:

$$x_{new} = x_{old} + \cos(\alpha) \tag{11}$$

$$y_{next} = y_{old} + \sin(\alpha) \tag{12}$$

The nearest obstacle to the robot can be detected with aid of a distance sensor which work based on the equation (13):

$$dist_{R-O} = \sqrt{\left(x_O - x_{new}\right)^2 + \left(y_O - y_{new}\right)^2} \tag{13}$$

where $\left(x_O, y_O\right)$ is the coordinate of the obstacle and $\left(x_{new}, y_{new}\right)$ is the coordinate of the AMR respectively computed using equations (11) and (12).

However, if the AMR does not sense any obstacle(s) in its target path during the search for the optimal path, then it will travel towards its target in a relatively straightforward manner using equation (11) and (12). But if the robot detects an obstacle along its path base on equation (13), the modified bat algorithm is implemented to steer the AMR by generating the best position that will provide an optimal collision free path to reach the target.

The following are the steps of the path planning technique for path planning of AMR:

Step I: Initialization of the source and target locations of the AMR.

Step II: Update the AMR position as it moves towards the target until an obstacle is sensed.

Step III: If AMR sensed an obstacle along its path as it moves towards the targetstep IV is implemented to generate the best position that will provide an optimal path.

Step IV: Generation of the bat population, each representing candidate solution and determine the elite solution using elite opposition based learning.

Step V: Compute the current best solution and find the global best solution.

Step VI: Generation of new solution by flying randomly

Step VII: The AMR proceed towards the bat's best position.

Step 8: Steps II – VII is repeated until the AMR generate bat's best position or reaches the target destination.

4. Simulation Result

Here, the simulation experiments are executed using the modified BA in 2 dimensional path planning environment containing obstacles. The experiments are performed using MATLAB R2018b on a PC with specifications: corei3 processor running on 2.30GHz, 4GB RAM and 500GB hard drive.

A. Modified Bat Algorithm Simulation Parameters

The performance of the modified BA depends on the appropriate selection of its control parameters (population size, frequency, velocity, position, pulse rate, loudness, iteration and search dimension). Table 1 present the appropriate values selected for these parameters:

Table 1 Modified BA Simulation Parameters

SN	Simulation Parameter	Value
1	Population Size	25
2	Frequency Range	0 – 2
3	Velocity	0
4	Pulse rate	0.5
5	Loudness	0.25

B. Path Planning Simulation Environment

The simulation environment is 120*cm* by 120*cm* square environment with different obstacle sizes placed randomly at different locations as shown in Figure 4.

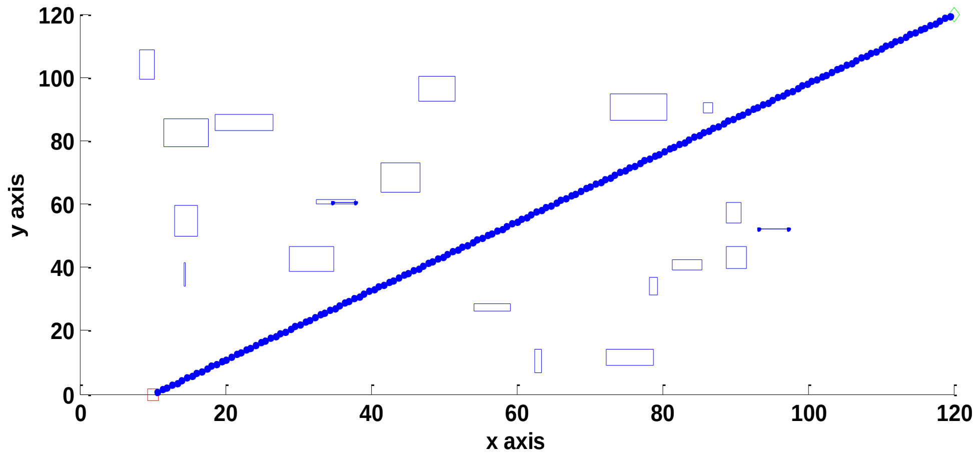

Figure 4: Optimal Collision Free Path Generated by AMR without Encountering Obstacle

From Figure 4, as the AMR move from the source position towards the target destination without encountering obstacles, the AMR continues to move using equations (11) and (12) until the target destination is reached.

However, when the AMR detect obstacles along its path using equation (13), the modified bat algorithm was implemented to steer the AMR within the environment by generating a collision free optimal path as shown in Figure 5.

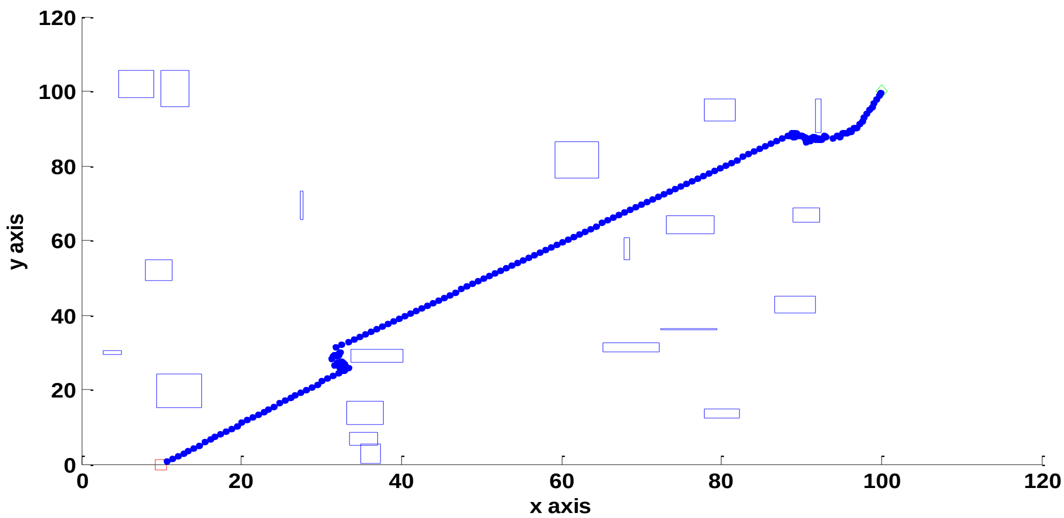

Figure 5: Implementation of Modified Bat Algorithm for Path Planning of AMR

To determine the effectiveness of the developed modified bat algorithm for path planning of AMR, three different environments with size 100*100, 120*120 and 160*160 was used to generate optimal collision free path under different number of obstacles. The time taken for each scenario of obstacles was taken and recorded. The performance of the modified bat algorithm for path planning of AMR was then plotted based on time taken to reach the target against environment complexity as shown in Figure 6.

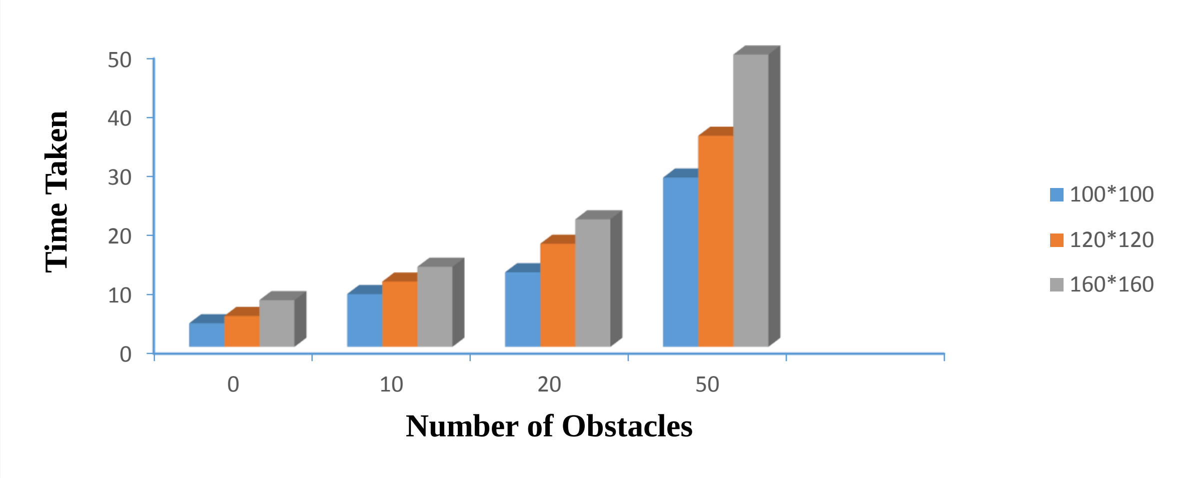

Figure 6: Performance of Modified Bat Algorithm for Path Planning AMR under Different number of Obstacles and Environment Size.

From Figure 6, as there is no obstacle in the environment, the time taken by the AMR to reach the destination were 3.9886, 5.2262 and 7.9126 seconds for 100*100, 120*120, 160*160 environments size respectively. When it was 10 obstacles, the time taken were 8.9503, 11.0742 and 13.5766 seconds for 100*100, 120*120, 160*160 environments size respectively. When 20 obstacles were used, the time taken were 12.6500, 17.4909 and 21.6400 seconds for 100*100, 120*120 and 160*160 environments size respectively. While for 50 obstacles, the time taken were 28.7084, 35.7726 and 49.5078 seconds for 100*100, 120*120, 160*160 environments size respectively. This indicates that as the size of the environment increases, the time to reach the destination increases under different complexity. The complexity here is based on the number of obstacles, i.e. the higher the number of obstacles, the more the environment become complex and vice versa.

To validate the performance of the developed path planning technique, BA and firefly algorithm were used for the implementation of the path planning technique in an unknown static environment. The simulation environment of 100*100, 120*120 and 160*160 were used as the environment sizes. Figure 7 presents the implementation of modified BA for path planning of AMR.

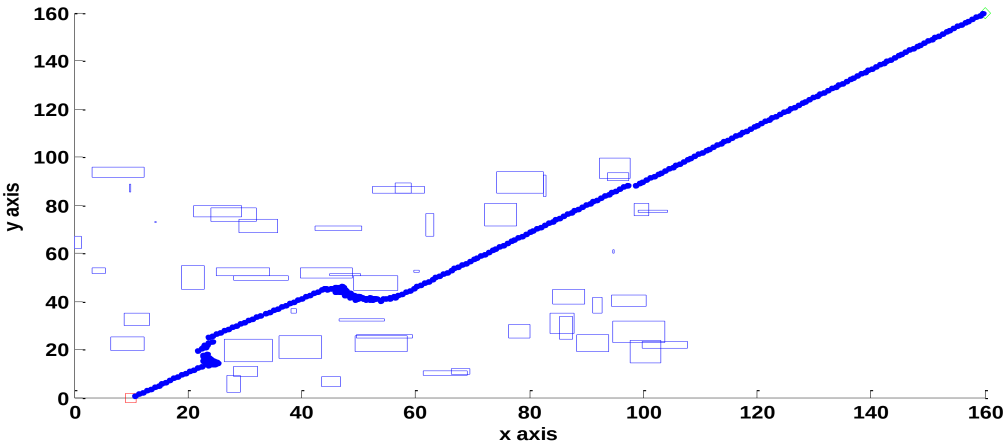

Figure 7: Implementation of Modified Bat Algorithm for Path Planning of AMR

From Figure 7, the time taken for AMR to reach its target location from the source location using modified bat algorithm path planning technique were recorded and tabulated in Table 2.

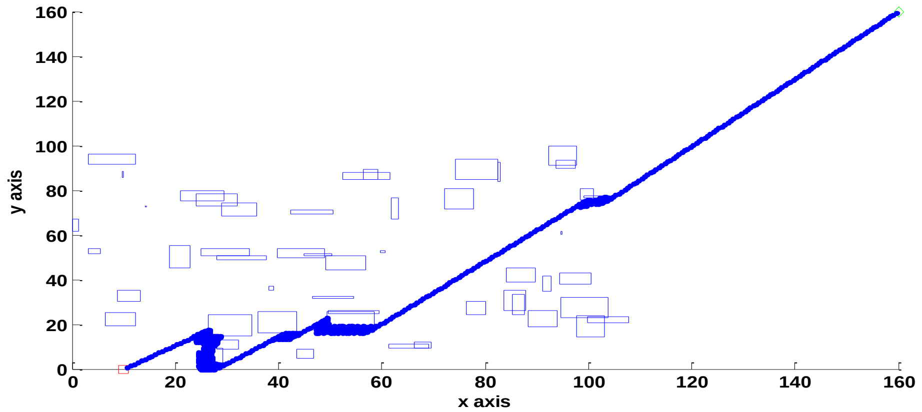

Figure 8: Implementation of Bat and ABC Algorithms for Path Planning of AMR

Figure 8, is the path planning implementation using different techniques (Bat and Artificial Bee Algorithms) for AMR path planning in simulation environments populated with 50 obstacles. The time taken by the AMR as it reached its target location is also recorded in Table 2:

Table 2: Time Taken in Environment with 50 Obstacles

Path Planning Technique	**Simulation Environment** 100*100	120*120	160*160	**Average Time (seconds)**
BA based	44.5963	61.6376	92.9651	66.3997
ABC Based	51.7663	72.6395	101.5169	75.3076
Proposed Technique	28.7084	35.7726	49.5078	37.9944

From Table 2, the time taken by AMR is recorded and the average time is computed and also recorded. For visibility of the difference in time taken as recorded by both path planning technique, Figure 9 is plotted.

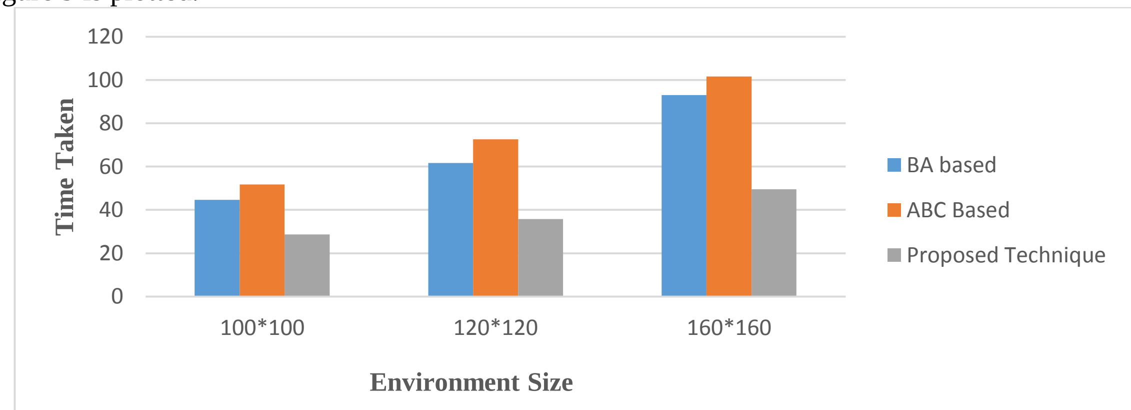

Figure 9: Performance of Path Planning Techniques for AMR under Different Environment Size Populated with 50 Obstacles.

From Figure 9, the average time taken by the robot to move from the source to the target in 100*100, 120*120 and 160*160 simulation environments with 50 number of obstacles using bat algorithm (BA), artificial bee colony (ABC) and proposed technique are respectively recorded as 66.3997 *seconds*, 75.3076 *seconds* and 37.9944 *seconds*. This indicate the superiority of proposed technique over BA and ABC as it provides optimal path between source and target with minimum average time.

Conclusion

This paper presented a dynamic path planning technique for autonomous mobile robot in an unknown static environment. The path planning algorithm known as modified bat algorithm which is a bio-inspired metaheuristic algorithm was modelled based on the hunting behavior of bats. The ability of the developed algorithm to obtain an optimal collision free path was determined in a developed simulation environment. The performance of the algorithm was determined based on path generation and time taken by the AMR to reach its target destination. The results obtained shows the effectiveness of the developed algorithm in different environment with different complexity. Future work will compare the performance of the developed algorithm with other nature inspired metaheuristic algorithms. Also, the performance of the developed algorithm can be validated by including moving obstacles in the environment and the developed path planning algorithm will be implemented in an indoor experimental set up.

Reference

[1] Alihodzic, A. and M. Tuba. *Improved hybridized bat algorithm for global numerical optimization*. in *Computer Modelling and Simulation (UKSim), 2014 UKSim-AMSS 16th International Conference on*. 2014. IEEE.

[2] Bora, T.C., L.d.S. Coelho, and L. Lebensztajn, *Bat-inspired optimization approach for the brushless DC wheel motor problem.* IEEE Transactions on Magnetics, 2012. 48(2): p. 947-950.

[3] Chu, S.-C., P.-W. Tsai, and J.-S. Pan. *Cat swarm optimization*. in *Pacific Rim International Conference on Artificial Intelligence*. 2006. Springer.

[4] Dorigo, M. and S. Thomas, *Ant Colony Optimization. Cambridge, vol. 9, Dec. 2002*. 2004, MIT Press.

[5] Eberhart, R.C. and J. Kennedy. *A new optimizer using particle swarm theory*. in *Proceedings of the sixth international symposium on micro machine and human science*. 1995. New York, NY.

[6] Hossain, M.A. and I. Ferdous, *Autonomous robot path planning in dynamic environment using a new optimization technique inspired by bacterial foraging technique.* Robotics and Autonomous Systems, 2015. 64: p. 137-141.

[7] Haruna, Z., et al. *Development of a modified bat algorithm using elite opposition—Based learning*. in *Electro-Technology for National Development (NIGERCON), 2017 IEEE 3rd International Conference on*. 2017. IEEE.

[8] Noel, M.M., *A new gradient based particle swarm optimization algorithm for accurate computation of global minimum.* Applied Soft Computing, 2012. **12**(1): p. 353-359.

[9] Reshamwala, A. and D.P. Vinchurkar, *Robot path planning using an ant colony optimization approach: A survey.* Int. J. Adv. Res. Artificial Intelligence, 2013. **2**: p. 65-71.

[10] Raja, P. and S. Pugazhenthi, *Optimal path planning of mobile robots: A review.* International Journal of Physical Sciences, 2012. 7(9): p. 1314-1320.

[11] Tang, K.-S., et al., *Genetic algorithms and their applications.* IEEE signal processing magazine, 1996. 13(6): p. 22-37.

[12] Yang, X.-S., *Engineering optimization: an introduction with metaheuristic applications*. 2010: John Wiley & Sons.

[13] Yang, X.-S. and S. Deb. *Cuckoo search via Lévy flights*. in *Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on*. 2009. IEEE.

[14] Yang, X.-S., *A new metaheuristic bat-inspired algorithm*, in *Nature inspired cooperative strategies for optimization (NICSO 2010)*. 2010, Springer. p. 65-74.

[15] Yang, X.-S. and A. Hossein Gandomi, *Bat algorithm: a novel approach for global engineering optimization.* Engineering Computations, 2012. 29(5): p. 464-483.