Comparison of BAT with PSO for Path Planning Problems

[1]Yogita Gigras, [2]Oshina Vasishth
[1] Asst.professor, [2]Student
[1] Department of Computer Science and Engineering,
[1] ITM University ,Gurgaon, India

Abstract - **Path planning is a fundamental but an essential task in the field of robotics .The completion and correctness of a problem in planning a path depends upon the accuracy of the algorithms used. One category of algorithms used in Path planning has a great influence from nature and are known as bio-inspired algorithms. These algorithms imitate the nature for discovering the path in an artificial environment. Bio inspired algorithms in path planning has evoked large attention from research community in past two decades which can we seen by the way large number of algorithms are being developed today to improve and increase the efficiency of finding the path.**

This paper shows the growth in this field by comparing two algorithms of initial and recent phases in which one is PSO which was developed in 1995 while the other is BAT algorithm developed in 2010 These two algorithms are implemented in dynamic environment and the path is evaluated using matlab, while the performance of algorithms are compared or calculated by changing the number of population.

Index Terms - **Bat algorithm, Particle Swarm Optimization, Path Planning, Bio-inspired algorithms.**

I. INTRODUCTION

Path planning is a mechanism for finding efficient path between two distinct points by overcoming the disturbances caused by obstacles .Path planning has shown an immense growth in past years. There are large number of algorithms which are developed and can be broadly classified and pictorially represented as :

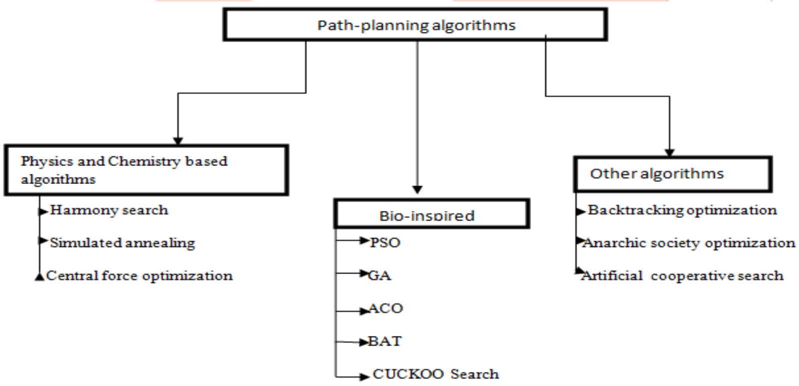

Figure 1: Division of path planning algorithms

Although all these categories of the algorithms are being used to find a path but the most evolving and attention gainer is the category of bio-inspired algorithms which is further divided in categories of algorithms based on Swarm intelligence(S.I) and non swarm(non S.I) Bio-inspired algorithms. Eg: of S.I based algorithms are : Bee colony optimization[12], Bat algorithm, Cuckoo search[11], Pso[9] etc and non SI based are genetic algorithm[14], Atmosphere clouds model[7], Brain Storm Optimization[13] etc.

Particle Swarm Optimization

Dr. Eberhart and Dr. Kennedy in 1995[9] first time implemented Particle swarm optimization which was based on the swarm behaviour of insects such as ants, a flock of birds etc. this swarm behaviour of insect in nature can be seen at the time of gathering food or searching a new colony(home) for themselves. During this action each member of the swarm contributes by sharing of information and takes a global step together to move. The number of individuals is known as particles and the whole population is come to be known as swarm. So the bigger is the number of particles the bigger is the swarm and represented as.
Swarm= {par1, par2, par3…..parN} where par1, par2, par3,….parN are the participant in forming a swarm .
In a search space these particles have their own position and velocity represented as follow
Particle position = {pos1, pos2, pos3,……..posN}
Particle velocity = {vel1,vel2,vel3…….......velN}

Each particle in swarm have its own best position value (X_{pbest}) .The particle which is having the best position which is calculated by using objective function or fitness function is considered to be the global best position (X_{gbest}).
After getting the two best values, the particle updates its velocity and positions using equation (a) and (b)

vel=vel+c1*(Rand1*(pbest-pos))+c2*(Rand2*(gbestx-pos)) (a)
pos=pos+vel (b)

Rand1 and Rand2 are random numbers between (0,1) and c1, c2 are learning factors. Usually the values of c1 and c2 is taken as 2.

Bat Algorithm

Xin-She Yang in 2010[10] developed an algorithm which was emulating the behaviour of bats, so this algorithm came to be known as bat algorithm. Bat algorithm uses one of the most important behavioural characteristic of bat known as echolocation to explore its path .Echolocation is a mechanism in which sound pulses are emitted by bat during his search for food and listen to the echo bouncing back from the obstacles which helps bat in finding obstacle free path. The sound pulses emitted by bats are usually short and loud and are emitted at a rate of 10 to 20 pulses per sec. Bats uses these reflected pulses as a source of information for finding their food .
The bat sound pulses are made from three mathematical factors, they are wavelength, loudness and frequency. As wavelength and frequency are inversely proportional, hence only one factor is used during implementation of the algorithm.
The bat algorithm uses the following steps :
Objective function ob(x), x=(x1,x2,…..xd).
1)Initialization : In this step the bat population location(loc),initial velocity(vel) ,maximum frequency[Fmax] and minimum frequency[Fmin] ,loudness[A] and pulse rate[P] are initialized.
2)Generation of new solution: this is implemented using two steps

a)by adjusting frequency –

F(i)=Fmin+(Fmin-Fmax)*β,

b)by updating velocity and location-

vel(i)=vel(i)+(x(i)-bestx)*F(i)
loc (i)= loc(i) + vel(i).

3)Local Search: it is done by choosing a result among the best result
And the condition applied is **(Rand>P)**
4)Generating new solution by flying randomly.

(Rand<A and ob(xi)<ob(x_*))

PROBLEM DEFINITION AREA

Considering a space of 120X120 dimension in which robot will move. The space consist of starting point(X1,Y1) in left bottom corner represented using red square while green diamond represent destination point (X2,Y2) on right upper corner .20 randomly generated obstacles are plotted in the space which will create hindrance for robot while finding a path .The aim is to find a path between starting and destination point by avoiding the obstacles using bat and pso algorithms.

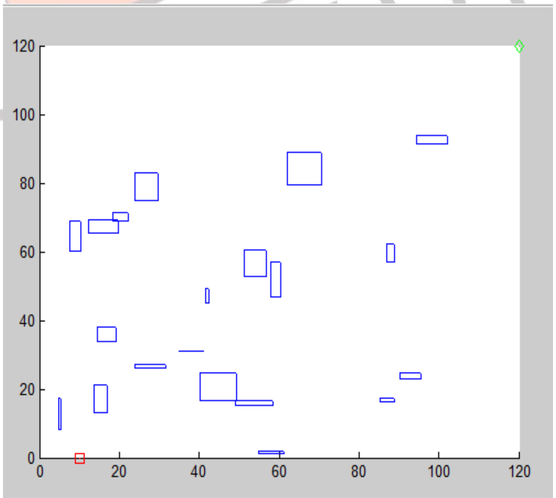

Figure 2: Problem Configuration

II. PATH PLANNING ALGORITHM FOR DETECTING COLLISION FREE PATH

This algorithm is designed to find an effective path between starting and destination point in an artificial environment after avoiding the obstacles.

Input: starting point(X1,Y1),destination point(X2,Y2), artificial environment having randomly generated obstacles.
Output: reaching to the destination point without colliding with obstacles

Algorithm:

Robot starts from the initial point and calculate the shortest distance between (X1,Y1) and (X2,Y2) using distance formula:

dist=sqrt((x2-x1).^2+(y2-y1).^2); (1)

Then the direction or the new coordinates are calculated using the following equations:

x1=x1+step*cos(theta); (2)

y1=y1+step*sin(theta); (3)

the theta is obtained as: **theta= atan((y2-y1)/(x2-x1)).** (4)

Condition 1: if no obstacle is encountered

If the robot encounters no obstacle during its search for the optimal path, it continues to apply the above formulas to calculate distance and directions (coordinates) and reaches to the final destination point.

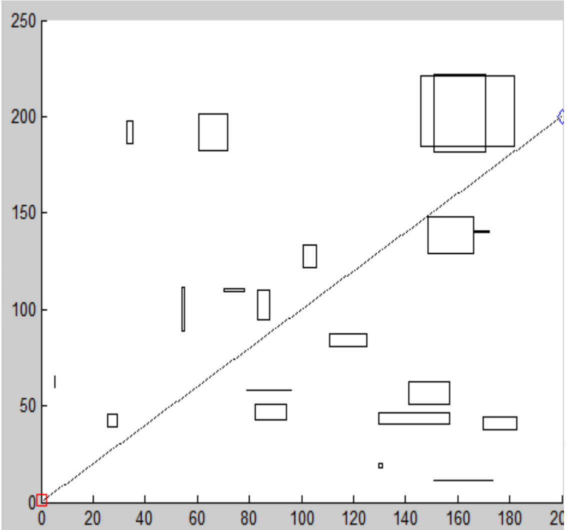

Figure3:Reaching to destination without encountering obstacle

Else

Condition 2: if obstacles are encountered

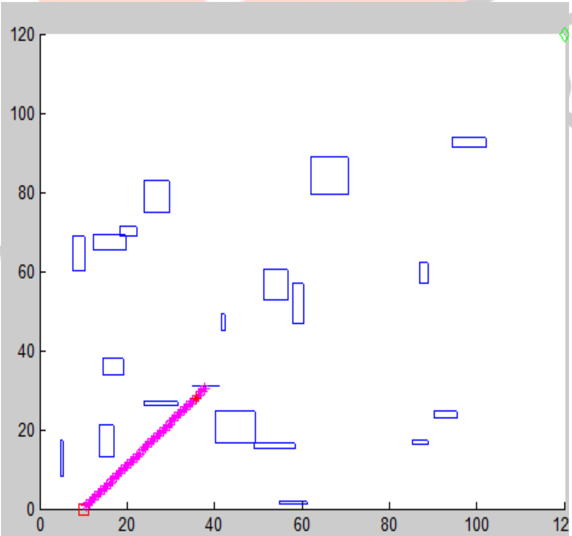

Figure 4: Obstacle encountered

If the robot encountered the obstacle in path while moving toward the destination path it will stop and take the following steps: Robot moves three steps back by using following equations:

x1=x1-3*cos(theta); (5)

y1=y1-3*sin(theta); (6)

theta= atan((y2-y1)/(x2-x1)). (7)

Now we can apply the path planning algorithms BAT and PSO to get escape from the obstacles and afterwards compares the performance of the algorithms.

Applying PSO

Initial random population is generated by assuming the size of the population as 10 particles .The position of these 10 particle is generated using the equation:

st1x(i)=cos(par(i)); (8)
st1y(i)=sin(par(i)); (9)

now the distance formula is applied which is acting as the objective function so that the particle having the minimum objective function could be selected and pso equation can be applied .
updating velocity on x and y coordinates.

vx(i)=v(i)+c1*(R1*(pbestx(i)-st1x(i)))+c2*(R2*(gbestx- st1x(i))); (10)
vy(i)=v(i)+c1*(R1*(pbesty(i)-st1y(i)))+c2*(R2*(gbesty- st1y(i))); (11)

the updated velocities are then used to updated the positons as:

posx(i)=posx(i) +vx(i); (12)
posy(i)=posy(i) +vy(i); (13)

hence whenever the obstacle is encounter the pso is applied and the new coordinates are generated which helps robot in avoiding the obstacles and finding the obstacle free path.

Applying BAT algorithm

Bat algorithm is recently developed algorithm which is applied as follow:
Generate the initial population of 10 bats .There positions are randomly generated as:

st1x(i)=cos(bat(i)); (14)
st1y(i)=sin(bat(i)); (15)

using the minimum objective function the best bat is selected and then the bat algorithm equations are applied separately on X-coordinate and Y-coordinate.

frequency updation

F(i)=Fmin+(Fmin-Fmax)*rand; (16)

velocity updation at X and Y coordinates

vx(i)=v(i)+(st1x(i)-bestx)*Q(i); (17)
vy(i)=v(i)+(st1y(i)-besty)*Q(i); (18)

location updation at X and Y coordinate

st1x(i)=st1x(i)+vx(i); (19)
st1y(i)=st1y(i)+vy(i); (20)

keep on applying bat algorithm whenever the robot encounter the obstacles.

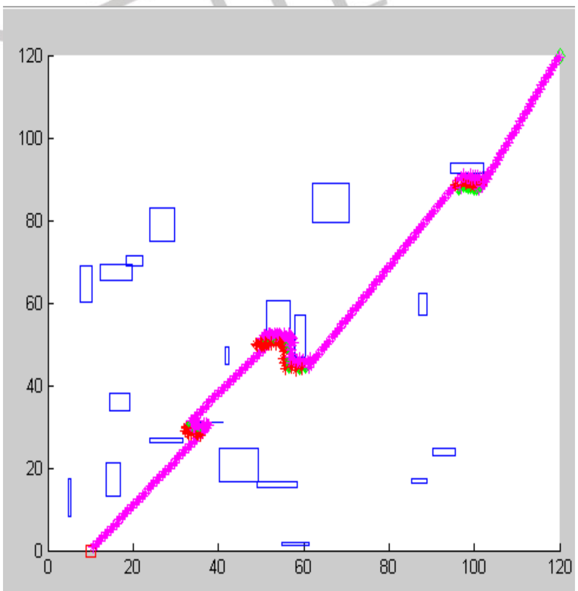

Figure 5: Our aim is obtained as robot finds the path between final and destination point after avoiding obstacle

III. RESULT AND DISCUSSION

The pso and bat algorithms for path planning are being compared by increasing the number of population against number of iteration being executed and we found that BAT algorithm is much more efficient than PSO.As iteration taken by BAT are lesser than PSO which also shows that time taken by BAT would be lesser than PSO in finding a obstacle free path. A comparison tables between bat and pso

Table 1. simulation results only for particle swarm optimization and bat algorithm .(not for whole process)

Population size	**PSO(no.of iteration)**	**BAT(no.of iteration)**
5	6	2
10	11	2
15	16	2
20	21	2
25	26	2

Table 2. simulation results applied on whole path planning using PSO and bat algorithm

Population_size	**PSO(no.of iteration)**	**Bat(no of ieration)**
5	859	213
10	729	442
15	435	338
20	630	242
25	759	204

IV. CONCLUSIONS

Both the techniques used in this paper are applied for path planning though both of them have wide scope of usage like Pso can be used to solve problems related to engineering field because of its exceptional features like proximity, quality, stability and adaptability while Bat algorithm can be applied in the field of optimization, image processing, feature selection, scheduling, and others.

REFERENCES

[1] Abdullah Zawawi Mohamed, Sang Heon Lee , Mahfuz Aziz, Hung Yao Hsu,Wahid MdFerdous. A Proposal on Development of Intelligent PSO Based Path Planning and

ImageBased Obstacle Avoidance for Real Multi Agents Robotics System Application,International Conference on Electronic ComputerTechnology (ICECT), 2010.

[2] Beatriz A. Garro, Humberto Sossa and Roberto A. Vázquez . Path Planning Optimization Using BioInspirited Algorithms. International Conference on Artificial Intelligence,2006;319 – 330.

[3] M. Sabry Hassouna, Alaa E. Abdel-Hakim, and A.A. Farag.Robust Robotic Path PlanningUsing Level Sets. IEEE International Conference on Image Processing,2005;

[4] X.S. Yang. Review of meta-heuristics and generalized evolutionary walk algorithm. International Journal of Bio-Inspired Computation,2011;77–84.

[5] Xin-She Yang .Bat Algorithm: Literature Review and Applications. International Journal of Bio-Inspired Computation. 2013.

[6] Gao-Wei yan and Zhan-ju Hao. A novel optimization algorithm based on atmosphere clouds model. International Journal of Computational Intelligence and Applications, 2013.

[7] Yogita Gigras1 and Kusum Gupta. Ant Colony Based Path Planning Algorithm for AutonomousRobotic Vehicles. International Journal of Artificial Intelligence & Applications (IJAIA), 2012.

[8] Kennedy, J., and Eberhart, R. Particle Swarm Optimization, Institute of Electrical and Electronics Engineers(IEEE),1995, 1942 - 1948 .

[9] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm, Nature Inspired Cooperative Strategies for Optimization (NISCO),Studies in Computational

Intelligence, Springer Berlin, 284,Springer,2010, 65-74.

[10] X.S. Yang, S. Deb. Cuckoo search via Lévy flights. World Congress on Nature & Biologically Inspired Computing (NaBIC), IEEE Publications,2009, 210 – 214.

[11] D.T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri , S. Rahim , M. Zaidi, The Bees Algorithm – A Novel Tool for Complex Optimisation Problems,2005.

[12] Yuhui Shi. An optimization algorithm based on brainstorming process. International Journal of Swarm Intelligence Research (IJSIR), 2(4):35–62, 2011.

[13] Eiben, Genetic algorithms with multi-parent recombination. Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature,1994, 78–87

[14] Sharad N. Kumbharana1, Prof. Gopal M. Pandey.2013.A Comparative Study of ACO, GA and SA for Solving Travelling Salesman Problem.

[15] Ant Colony System Based Mobile Robot Path Planning .2010. Song-Hiang Chia, Kuo-Lan Su, Jr-Hung Guo, Cheng-Yun Chung.

[16] M. Dorigo, V. Maniezzo & A. Colorni.1996."Ant System: Optimization by A Colony of Cooperating Agents.

[17] Wu, Q.2010. Car assembly line fault diagnosis based on robust wavelet SVC and PSO, Expert Systems with Applications.