Comparison between Different Meta-Heuristic Algorithms for Path Planning in Robotics

Yogita Gigras
The NorthCap University
Gurgaon
Haryana

Nikita Jora
The NorthCap University
Gurgaon
Haryana

Anuradha Dhull
The NorthCap University
Gurgaon
Haryana

ABSTRACT

Path planning has been a part of research from a decade and has been evolving with use of several heuristic as well as meta-heuristic techniques. In this paper, path planning is implemented using bee colony optimization algorithm which is self evolved with certain defined parameters. Artificial bee colony optimization algorithm is approached because of its efficiency, Performance and fewer parameters as compared with existing algorithms. It combines multiple objectives to solve complex strategies and further proves itself to be most prominent algorithm for navigation. Further it is compared with existing algorithms simultaneously.

General Terms

Path planning algorithms

Keywords

Path planning, artificial bee colony algorithm, Particle swarm optimization.

1. INTRODUCTION

We ask that authors follow some simple guidelines. In essence, we ask you to make your paper look exactly like this document. The easiest way to do this is simply to download the template, and replace the content with your own material. Artificial bee colony algorithm also known as greedy colonial convergence algorithm is a population based strategy as it can define its convergence in its best way. It is opted because of its significance in various diameters which will be elaborated later. ABC algorithm has been in vertical since 2005 discovered by karaboga. It is based on foraging behavior of honey bees and how they respond to changing position of food sources. Giovanni M. Bianco et al. [1] introduced its mapping paradigm which is inspired by bee's behavior at a very precise but on a large scale. They represented two landmarks to locate better solutions i.e. topological navigation or global potential function which globalizes area of interest and second level applies local landmark potential function which enhances the map to be finer. After which it got explored tremendously and researched further by karaboga et al. [2] as they presented a numerical optimization function to clarify actual calculations to perform such meta- heuristic techniques. Further they proposed improved performance of ABC with different approaches as well as functionalities [3] [4].

Further with the trial on performance they introduced [5] comparative study of artificial bee colony algorithm on the basis of mathematical computation with other existing meta-heuristic techniques. With such development Akay et al. [6] applied some constraints to the objective function and compared it with unconstrained heuristics. With such modifications they improved the performance without any exceptions left behind. Being through this journey they correlated such consummation on real systems for checking optimization on real parameters [7]. C.Zhang et al. [8] performed artificial bee colony system connected to clustering hypothesis.

F.S.Abu-Mouti et al. [9] featured an overview of all characteristics of artificial bee colony optimization algorithm with its known applications. P.Civicioglu et al. [10] kicked-off with comparison of some techniques to visualize the performance and time complexity. They compared artificial bee colony algorithm with Cuckoo-search, particle swarm optimization, differential evolution. With such enhancements and comparisons it has proved to be one of the efficient meta-heuristic techniques in today's scenario.

2. PATH PLANNING

Every technique has its characteristics as well as parameters. This algorithm is swarm based meta-heuristic technique created to solve optimization problems. Intelligent foraging behavior of honey bees inspired the researchers to develop optimized or nearly optimized solution to navigation problem. Therefore this multi-objective path planning has been widely adopted and practiced with different approaches. It includes two objectives:

- Avoid obstacles to prevent collision.
- To obtain shortest path from starting point to target point.

The following algorithm comprise of 4 components

- Employed bees,
- Onlooker bees,
- Scout bees and
- Food sources

where food sources represents one of the possible solution or updated position for the robot movement. Possible solution leads to shortest path connected from initial to goal position.

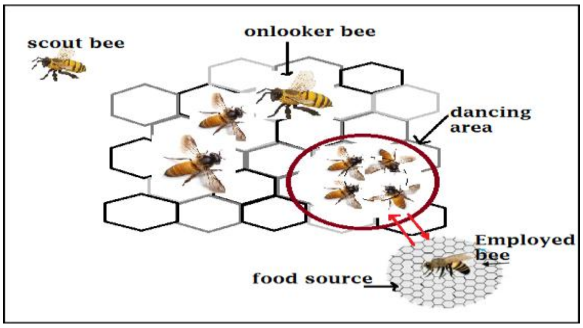


Figure 1: Artificial Bee Colony

In detail, the leading model describes the positivity by the behavior of bees such as searching for food sources rich in honey/food and abandon the food sources which results in poor quality. Functionality of these above mentioned components is:

Employed bee: It travels to various possible food sources which exist as the previous positions. And such bees move to the dance area after learning food source positions and dances or rotate with some defined velocity.

Onlooker bees: These bees wait around the dance area to examining the nature of the food source visited by employed bee so that they can calculate intensity of the food source for adaption of better results than previous one.

Scout bees: They carry out some random search for new food sources.

This food source virtualizes a possible solution of the path optimization problem and the amount of the food source or the nectar symbolizes the fitness function of the problem [11].

3. ABC ALGORITHM

- Initialization

For starting of the algorithm we first initialize the distributed population P of N p solutions i.e. food source position. Each one of the solution is a 2-Dimentional vector.

- Updating employed bee position

For updating new position employed bee calculates/memorizes nectar amount according to local information as equated below:

$$X'ij = Xij + c*(Xij-Xkj) \qquad (1)$$

Where c is some constant between [0, 1], X'ij is the new updated position and Xij is the previous position taken by bee to new food source.

Now nectar amount (position of the robot from goal point) of this new position must be evaluated and if it is better than previous one then this value is memorized and old one is discarded.

- Induction on food sources by onlooker bees

After the completion of processing of employed bees, they share the collected information on the dance area with onlooker bees. With this evaluated and collected information from employed bees they choose the food source according to probability function as given below:

$$Pi = \frac{fitness(Xi)}{\sum_{i=0}^{Np-1} fitness(Xi)} \qquad (2)$$

Where fitness (Xi) values indicate the nectar amount of the food source at position i. N p is the number of possible solutions which is equal to employed bees. In certain processing onlooker bees find out the most promising position or the food source which can be concerned further.

Now for position update (Xi') nearby the previous value Xi, a random variable k is taken as solution parameter and a random parameter c which is in range [-1, 1]. They are computed to get uniform random positions for next step of the robot, where k is any random number between 0 to N p -1 and not equal to i.

- Working of scout bees

After maximum number of iterations, food sources are abandoned which are not likely to be rich in nectar (far away from goal point). And the abandoned positions are replaced by scout bees to have random positions for further movement.

All these steps are repeated until a stopping criterion is met or robot has reached its destination.

3.1 Pseudo Code Of Artificial Bee Colony Algorithm

```
PROCEDURE abc

Initialize the population
Calculate the fitness function (fitness (Xi)) of the population
Initialize max_it
{       For each employed bee
        {
                Generate new modification (Xi') using (1)
                Evaluate fitness (Xi')
                Fitness with min value is chosen
                Position is updated
        }
        For each onlooker bee
        {
Select food source (Xi) depending upon probability (pi) using (2)
                Generate new modification (Xi') using (1)
                Evaluate fitness (Xi')
                Fitness with min value is chosen
                Position is updated
        }
Remember best solution obtained so far
For i=0: N p
{
         If (max_it>limit)
        Reinitialize position (Xi) by scout bees;
}
}
PROCEDURE move
{

Set initial and final position of robot as

(Xi, Yi), (Xi', Yi') respectively, threshold value e

        {

                Set (Xbest ← Xi), (Ybest ← Yi)
                Call abc (Xbest, Ybest)
                move_to (Xbest, Ybest);
                Until (besti – gi) > e
        }
}
Update
 Xbest = Xbest + cos (z);
Ybest = Ybest + sin (z);
```

Where Z is an angle from which robot is about to turn towards destination.

3.2 Stepwise Implementation Of Path Planning Using ABC Algorithm For Detecting Collision Free Path

Path planning is performed on MATLAB R2010a.

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 containing 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:

$$\textbf{MinDistance} = \sqrt{(Y2 - Y1)^2 + (X2 - X1)^2} \quad (3)$$

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

$$x1=x1+step*cos(theta); \quad (4)$$

$$y1=y1+step*sin(theta); \quad (5)$$

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.

But if the robot collides with an obstacle in between the path while moving toward the destination path it will stop and take some steps back to reroute the path.

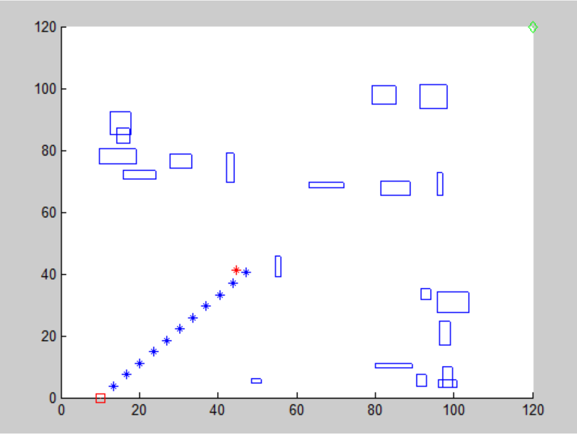


Figure 2: Obstacle encountered

Further artificial bee colony algorithm is applied so that robot selects most optimal path. And also avoids obstacles and steps forward towards destination which is represented below:

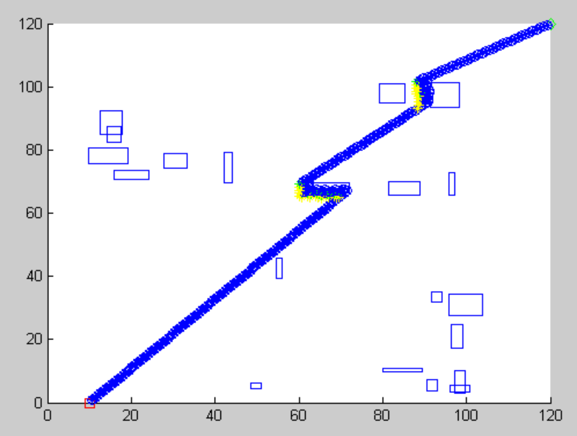


Figure 3: Robot reaching the destination using ABC algorithm.

As per the above figure, red square is the starting point og the robot and it has to reach the destination which is pointed green and some obstacles are set randomly on the artificial environment so that performance can be measured with different algorithms. Now it is determined that robot has reached the goal after escaping from obstacles and searched the shortest path using meta-heuristic techniques so the results are mentioned below.

4. PSO ALGORITHM

Dr. Kennedy et al. [7] introduced and implemented in 1995 which is also a nature based algorithm where a group of birds or particles indicates swarm behavior. The following individuals are said to be particles and the group of population is said to be a swarm. This type of swarm behavior of a flock of birds in nature can be viewed at the point of searching food or finding a new location to stay. During this process every particle of swarm bequests by fragmenting the information and move together globally with each step knowing each other's position. As compared to other algorithms it is known to be more useful as it has small population area; and also it converges fast as well as it has a very few parameters. But everything seems to have some drawbacks [12]. In its initial phase [8], it got trapped into its local minima, so to handle it, different versions of PSO algorithm has come up from which [wPSO] i.e. inertial weight PSO algorithm was invented adjoined with two parameters used (velocity and position).

Basically swarm is the pattern made by particles over the sky in certain fixed position and velocity of each particle.

Swarm (s) = {p1, p2, p3…..pN} where p1, p2, p3,….pN are the participants forming a Swarm .

In search space all these particles have their own position as well as velocity defined as below:

Particle's position = {ps1, ps2, ps3………psN}

Particle's velocity = {vl1, vl2, vl3…….........vlN}

After each step, every particle in the swarm updates its own best position (Xpbest). And the particle which is having same position repeatedly is known to be its best global position (Xgbest) calculated via fitness rule.

After obtaining two best values, the particle updates itself with new velocity and position using equation (a) and (b)

Vl = vl+c1*(Rand*(pbestx-ps))+c2*(Rand*(gbestx-ps))

Ps = ps+vl

Rand is the random number within range [0, 1] and c1, c2 are learning factors. Usually the value of c1 and c2 is taken as 2.

4.1 Pseudo Code Of Particle Swarm Optimization Algorithm

```
For each of the individual particle in swarm
        {
                Initializing individual particle

Repeat until the maximum number of iteration is accomplished
        {
                For every particle of the swarm
        {
Find the fitness function value
        If present value is less than the Pbest
                {
                Pbest ←current fitness value
                }
        If estimated Pbest is better than the Gbest
                {
                Set Gbest=Pbest
                }
        }

Calculate velocity of each of the individual particle of the swarm
Particle position is updated using the Gbest value of evaluated velocity        }
```

4.2 Step Wise Implemenation Of Path Planning Using Pso Algorithm For Detecting Collision Free Path:

Random population is initialized by taking population size or number of particles as 10. The positions of these 10 particles are inaugurated using the following equations:

setx(i) = cos (par(i));   (7)

set1y(i) = sin(par(i));   (8)

Now the distance formula as represented in (3) is executed which is described as the fitness/objective function so that specific particle having minimum value of objective function could be preferred and PSO derivations can be practiced.

Updating velocity on x and y coordinates as follows :

velx(i)=v(i)+c1*(R1*(pbestx(i)-set1x(i)))+c2*(R2*(gbestx-set1x(i));

vely(i)=v(i)+c1*(R1*(pbesty(i)-set1y(i)))+c2*(R2*(gbesty-set1y(i));

(a)

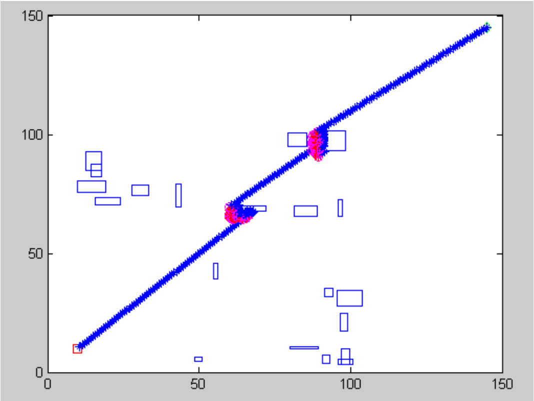
(b)

After this the refreshed velocities and positions are:

psx(i) = psx(i) +velx(i);   (12)

psy(i)=psy(i) +vely(i);   (13)

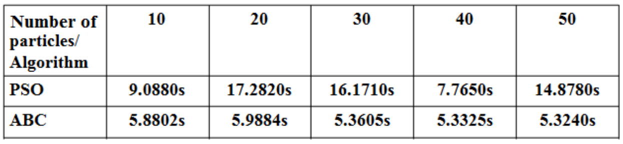
Using these equations the moving robot can be paused whenever any obstacle turns up and hence new positions are updated with rejuvenated velocities and the new coordinates are generated which helps robot in avoiding some obstructions and finding out the destruction free path. This is applied on the same coordinates as ABC was implemented so that it can measure elapsed CPU time by both the algorithm simultaneously. The following is the artificial environment generated for PSO algorithm where source and destination are set and whenever robot changes its path or reroutes the coordinates for path planning it generates some magenta colored points which indicates the changing of points for avoiding the hindrance:



5. RESULTS

After applying artificial bee colony algorithm and particle swarm optimization algorithm, the results on the basis of CPU Time elapsed is elaborated in the following table as follows:

Table 1 CPU Time elapsed when number of population versus algorithm implemented.



Number of particles/ Algorithm	10	20	30	40	50
PSO	9.0880s	17.2820s	16.1710s	7.7650s	14.8780s
ABC	5.8802s	5.9884s	5.3605s	5.3325s	5.3240s

In the given results, PSO algorithm keeps on fluctuating results and is coming out to be more than the time taken by ABC algorithm which proved ABC to be better algorithm due to its capability to converge efficiently and improve its performance with increase in population size.

6. CONCLUSION

As per results and findings it is concluded that ABC is superior then PSO however there are some contradictory discussions such as complexities. Both are evolving and have their own advantages over each other but ABC has come up with some better outcomes. As compared to PSO, ABC converges fast and is more efficient in performance. Also in particle swarm optimization, particles converge together and shows cohesive part but in artificial bee colony algorithm, bees converge separately which leads to more optimistic behavior. As shown by table no.1 PSO is not meant for large population, on contradictory way ABC is better for large population and has few parameters, high accuracy. Although ABC is better for real time problems but in near future, to enhance the capabilities some of the evolutionary techniques can be combined such as ABC and PSO or BAT for embellishing various complex and major applications such as job scheduling, clustering, mining etc requiring highly optimized solutions.

7. REFERENCES

[1] Giovanni M. Bianco," Getting inspired from Bees to perform large scales visual precise navigation" Proceedings of 2004 IEEEmSJ International Conference on Intelligent Robots and Systems September 28. October 2, 2004. Sendai. Japan, pp: 619-624

[2] Karaboga, Dervis. "An idea based on honey bee swarm for numerical optimization." Technical. Rep. TR06, Erciyes Univ. Press, Erciyes (2005).

[3] D. Karaboga and B. Baştürk, "On the performance of artificial bee colony (ABC) algorithm," Appl. Soft Computer., vol. 8, no. 1, pp. 687-697, 2008.

[4] D. Karaboga, B. Basturk, "On The Performance Of Artificial Bee Colony (ABC) Algorithm", Applied Soft Computing, Volume 8, Issue 1, pp. 687-697, January 2008.

[5] Karaboga, D. and Akay, B., "A comparative study of Artificial Bee Colony algorithm," Applied Mathematics and Computation, In Press, 2009.

[6] D. Karaboga, B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems" Applied Soft Computing, Volume 11, Issue 3, pp. 3021–3031, April 2011.

[7] B. Akay, and D. Karaboga, "A Efficient artificial bee colony Algorithm for Real-parameter Optimization, "Information Sciences, Vol. 192, No. 1, pp. 120-142, 2012.

[8] C. Zhang, D. Quyang, and J. Ning, "An Artificial Bee Colony Approach for Clustering," Expert Systems with Applications, Vol. 37, No. 7, pp. 4761-4767, 2010

[9] F.S.Abu-Mouti and M.E. El-Hawar, "An overview of artificial bee colony algorithm and its applications", IEEE Transactions on Power Systems, pp. 1-6, 2012.

[10] P.Civicioglu, E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms." Artificial Intelligence Review, 39(4): pp. 315-346, 2013.

[11] Preetha Bhattacharjee, "Multi-Robot Path-Planning Using Artificial Bee Colony Optimization Algorithm", IEEE conference 2011, pp: 219-224.

[12] Nikita, "Self Exploring Approach for path planning in robotic domain", conference: CSI - 2015; 50th Golden Jubilee Annual Convention,pp:1-6,2015