Implementation and Comparison between PSO and BAT Algorithms for Path Planning with Unknown Environment

Dr. Mukesh Nandanwar[1], Anuj Nandanwar[2]

[1]*Assistance Professor (CSE), Chhattisgarh Engineering College, Durg, Chhattisgarh, India*
[2]*Ph.D. (Scholar), Chhattisgarh Engineering College, Durg, Chhattisgarh, India*

***Abstract*—The main challenges of robotics are its automation and detection capability. Robotic Path Planning is one of the main problems that deal with computation of a collision-free path for the given robot, along with the given map on which it operates. This paper focuses on Path planning and exploration problem in unknown environment with multi-robot system.**

Here we have dealt with two traditional nature-inspired algorithms namely Particle Swarm Optimization and Bat Algorithm for movement decision making with unknown environment and last comparing them different features i.e. moves, time, coverage and energy.

***Keywords*—BAT, Bio-inspired Algorithm, Multi-Robot system, Path Planning, PSO.**

I. INTRODUCTION

Every automated and intelligent robotic system uses planning to decide the motion of root and real world. Robotics is an extremely inter-disciplinary area that takes inputs from numerous disciples and fields with varying complexities.

Soft Computing Tools and Techniques are finding increasing application in the field of robotics. Many of the robotic problems like planning, coordination etc. are complex problems that are now-a-days solved using the most sophisticated soft computing tools. The use of Soft Computing tools and techniques lies at each of the steps in multiple ways. At the time of information processing Soft Computing tools not only help in pre-processing and noise removal, but help in landmark and object detection as well. Similarly many Soft Computing tools are used for the map building and path planning. The high-end intelligent planning involving many specific problems such as multi-robot coordination again involves algorithms from the domain of Artificial Intelligent and Soft Computing. In many real-world applications, workspace of robots often involves various danger sources that robots must evade, such as fire in rescue mission, landmines and enemies in war field [1].

We would first discuss some of the basics of robotics. Here we open the world of robotics which has infinite possibilities to the readers. The robot is a much generalized concept that may be applied in numerous applications and domains. Each has its own set of hardware specifications and design. The good design of a robot plays a very important role in fuel economy, ability of the robot to pass uneven, rough terrain or to survive such extreme conditions. The 'autonomous' robots make extensive use of sensors to guide them in and enable them understand their environment. A mobile robot exploring an unknown environment has no absolute frame of reference for its position, other than features it detects through its sensors [2]. A map is a representation of the robotic world that useful for numerous purposes, especially for the purpose of path planning. Path planning is the key task in the field of Robotics.

The modeling environment and algorithm to find shortest, collision free path are the basic issues in the path planning problem of the robot motion planning [3]. Path planning in robot is the problem of devising a path or strategy which would enable the robot to move from a pre-specified source to a pre-specified goal. The map thus generated needs to ensure that the robot does not collide with the obstacles. The path planning would depend upon the size of the robot and the obstacles. In many cases the robot may assumed to be of point size. This happened when we are sure that the robot would be in most of the cases able to glide through obstacles, assuming the obstacle density is not high. Another important characteristic of the path planning algorithm is its optimality and completeness. The completeness refers to the property that the algorithm is able to find a solution to the problem and return the solution in finite time, provided a solution exists.

Path planning is one of the basic and interesting functions for a mobile robot [4]. The problem of robotic path planning has always attracted the interests of a significantly large number of researchers due to the various constraints and issues related to it. The optimization in terms of time and path length and validity of the non-holonomic constraints, especially in large sized maps of high resolution, pose serious challenges for the researchers [5]. Path planning can be classified into two categories global path planning with all

the information of the robot's known environment and local path planning in a partly or totally unknown environment [6]. Since the strength of a trapped person often declines with time in urgent and dangerous circumstances, adopting a robot to rescue as many survivors as possible in limited time is of considerable significance. However, as one key issue in robot navigation, how to plan an optimal rescue path of a robot has not yet been fully solved [7].

Particle Swarm Optimization (PSO) is a bio-inspired algorithm. It is used many field i.e. social modeling, computer graphics, simulation and animation of natural swarms or flocks with artificial neural networks and evolutionary computation [8]. PSO algorithm has been widely used in various engineering problems because of its simplicity and efficiency [9]. Particle Swarm Optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best unknown position but, is also guided toward the best unknown positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions [Cite].

Bat algorithm is also bio-inspired algorithm. It is a meta-heuristic optimization algorithm developed by Xin-She Yang in 2010. This bat algorithm is based on the echolocation behavior of microbats with varying pulse rates of emission and loudness. The idealization of the echolocation of microbats can be summarized as follows: Each virtual bat flies randomly with a velocity v_i at position (solution) x_i with a varying frequency or wavelength and loudness A_i. As it searches and finds its prey, it changes frequency, loudness and pulse emission rate r. Search is intensified by a local random walk. Selection of the best continues until certain stop criteria are met. This essentially uses a frequency-tuning technique to control the dynamic behavior of a swarm of bats, and the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters in bat algorithm [Cite].

Matlab 2013a is using for simulation purpose. All simulation experiments were performed on a Core 2 Duo CPU 1.80 GHz machine running Windows 7 with 3 GB of RAM.

II. LITERATURE REVIEW

This chapter focus on the literature that been carried out during research. This is an important phase of the research; because it can help to demonstrate the problem in details and previous work carried out that are related to our research work. It also help what are gap exist in our field, so we can do work on that area. This chapter describes the various nature inspired algorithms i.e. PSO and BAT, path planning problem with multiple obstacles into unknown environments and finding the optimized path for a robot.

Dai S. (2009) was proposed the path planning for moving intelligent machines. There are two main issues in navigation; obstacle avoidance and goal seeking. Goal seeking is the process of process of navigation from start to destination and obstacle avoidance is to complete the mission without colliding with any of the obstacles [10]. Jiang Y. et al (2007) was proposed an improved particle swarm optimization (IPSO). In the new algorithm, a population of points sampled randomly from the feasible space. Then the population is partitioned into several sub-swarms, each of which is made to evolve based on particle swarm optimization (PSO) algorithm. Compared with PSO, IPSO is also applied to identify the hydrologic model. The results show that IPSO remarkably improves the calculation accuracy and is an effective global optimization to calibrate hydrologic model [11]. Nasrollahy A. Z. et al (2009) was proposed PSO based approach for robot path planning problem in dynamic and unknown environments with moving obstacles and target in order to minimize the total cost [12].

Nakamura R. Y. et al (2012) was proposed a new nature-inspired feature selection technique based on the bats behavior, which has never been applied to this context so far. The wrapper approach combines the power of exploration of the bats together with the speed of the Optimum-Path Forest classifier to find the set of features that maximizes the accuracy in a validating set [13]. Fister Jr I. et al (2013) was presented a new swarm intelligence algorithm, based on the bat algorithm. The Bat algorithm is hybridized with differential evolution strategies. An experiment has shown that this algorithm improves significantly the original version of the bat algorithm [14].

III. METHODOLOGY

This chapter describes the methodology of the overall work include the multi robot path planning. It includes various concept and method that are using in path planning and exploration that are designed for solving the problem. Path planning problem with multiple obstacles refers to the calculation of the shortest distance from the starting point to the destination avoiding the obstacles coming in between.

The algorithms discussed above are based on the concept of population which represents a set of solutions. Each solution is associated with a dimension. Dimension represents the coordinates of the rectangular space. The sample environment consists of a two-dimensional rectangular space comprising of stationary obstacles about which we have a priori knowledge. The environment information includes the limits of the rectangular workspace and the shape, location and orientation of all the stationary obstacles in the given workspace. The workspace is assumed to have no mobile obstacles. In path

planning a source and a goal point is given at the start of the problem. Both the source and goal points lie within the limits of the workspace. The overall work flow according the algorithm-1 to algorithm-4.

A. Algorithm-1: Step for Robot Movement by NIA

1. Initialize Environment Matrix in Graph Form G = (V, E).
2. Initialize N robot-agents and Associated Supporting Data Structure.
3. Initialize function to count Moves, Time, Coverage and Energy.
4. Start the particle from source
   for i=1 to n_1 (n_1 is the number of iterations)
   for j=1 to N (N is the number of robot-agents)
   if Random () < = Threshold value
   Move the agent using PSO / BAT (Algorithm-2 or 3)
   else
   Move using Directional Movement (Algorithm-4)
   end of if
   end of for j
   end of for i
   for j=1 to N (N is the number of robot-agents)
   if condition = = true
   Determine fitness of the particle
   Update the global and iteration bests
   end of if
   end of for j
5. Continue the loop until stop condition is true

B. Algorithm-2: Step for Robot Movement by PSO

1. Initialize position and velocity of all particles randomly in N dimension space.
2. Evaluate the fitness of each particle.
3. Compare the each individual particle fitness value with its pbest value. If the current value is better than its previous best make this current value as pbest.
4. Find gbest among all the particles. This gbest has the best fitness value.
5. Update the velocity and position according to equations.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.1)$$

$$v_i(t+1) = \omega v_i(t) + c_1\varphi_1(P_{ibest} - x_i) + c_2\varphi_2(P_{gbest} - x_i) \quad (3.2)$$

Where φ_1 and φ_2 are random variable, that is uniformly distributed within [0, 1], c_1 and c_2 are the weight change factors, ω is the inertia weight, P_{ibest} represent the best position of the particle and P_{gbest} the best position of swarm.

6. Repeat step 2-5 until a stop criterion is satisfied or a predefined number of iteration are completed.

C. Algorithm-3: Step for Robot Movement by BAT

1. First initialize the bat population x_i and velocity v_i.
2. Define pulse frequency f_i at x_i and initialize pulse rate r_i and loudness A_i.
3. While (t < max number of iteration)
   a. Generate new solutions by adjusting frequency and update velocity and location by using equations.

$$f_i = f_{min} + (f_{max} - f_{min}) \times \beta \quad (3.3)$$

$$v_i^{t+1} = v_i^t + (x_i^t - x_e) \times f_i \quad (3.4)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3.5)$$

   Where $\beta \in [0,1]$ is a random vector drawn from a uniform distribution. Here, x_e is the current global best solution. The values of the frequency f_{min} and f_{max} depend on the domain size of the problem of interest.
   b. If (rand > n)
   Select a solution among the best solution and generate the local solution around it using equation.

$$x_{new} = x_{old} + \varepsilon \times A_{mean}^t \quad (3.6)$$

   Where $\varepsilon \in [-1,1]$ is a random vector drawn from a uniform distribution. Here A_{mean}^t is the average loudness for all bats at time step t. $r_i \in [0,1]$ is the pulse rate.
   End if
   c. Generate new solution
   d. If (rand < A_i and f(x_i) < f(gbest))
   Accept the new solution and increase r_i and reduce A_i
   End if
   e. Rank the best and find the current gbest
   End while

D. Algorithm-4: Step for Robot Movement by directional Movement

1. Detect the present position P_{old}
2. If (target unknown)
   a. Select P_D from cluster direction
   b. Calculate unit vector for each dimension is

$$\delta = \frac{(P_D - P_{old})}{\|P_D - P_{old}\|} \quad (3.7)$$

3. Calculate new variation for each dimension as

$$V_K = \delta_k * D \quad (3.8)$$

Where $\delta_k \in \delta$ and $D \in S$ (Detection range)

$$P_{new} = P_{old} * V_k \quad (3.9)$$

Where $V \in (V_1, V_2, \ldots, V_k)$ and V is the variation.

IV. SIMULATION RESULTS

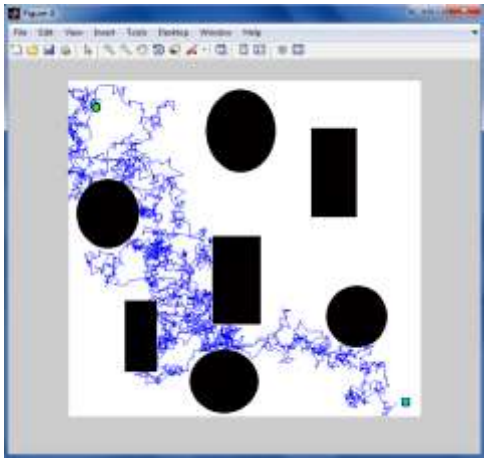
The problem of path planning deals with the determination of a path which navigates the robot in such a way that no collision occurs. In order to solve the problem I assume that the input is already available in form of a map. The region is traversable and may be used for the purpose of travelling. The black regions here signify the presence of obstacles. We are showing the results for path tracking and path searching by explored map to reach up to target. The summaries of experimental parameters are showing in the table 1.

PARAMETERS	VALUES
Map size	500x500
Number of maps	1
Sensor range	1 to 15
Target	Unknown
Number of robots	3, 6, 9 & 12
Start & ending point	[40,40] & [480,480]
Algorithms	PSO and BAT
Features	Move, Time, Coverage, Energy

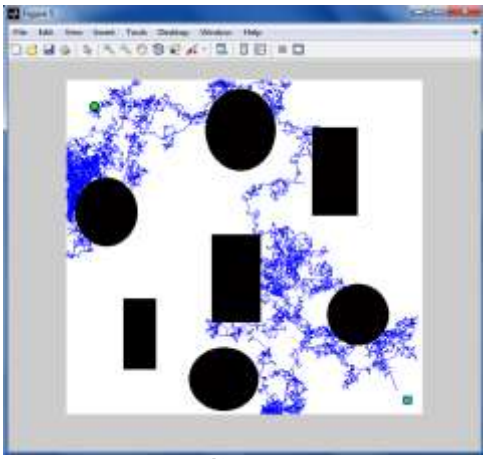
Table 1: Experimental Parameters

A. *Simulation Result Analysis*

PSO and BAT algorithms are useful for planning a path in static, complex state space environment. Figure 1 (a) and (b) are shown the explored area for outdoor environment. We can observe that the explored area is more when using PSO with respect to BAT algorithm, means the faster algorithm is BAT than PSO.



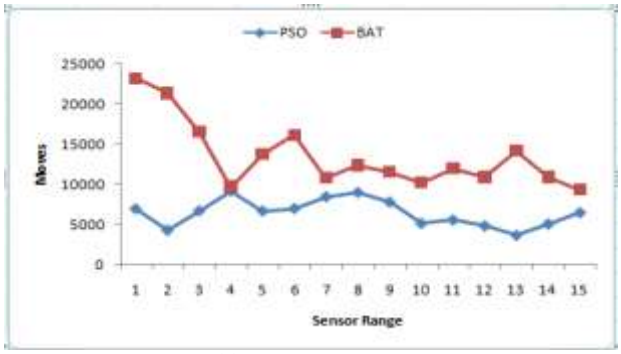
(a) PSO



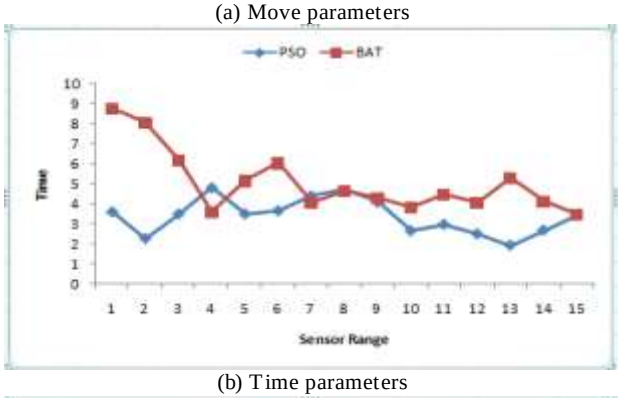
(b) BAT

Fig 1: Explored Area to reach in unknown target

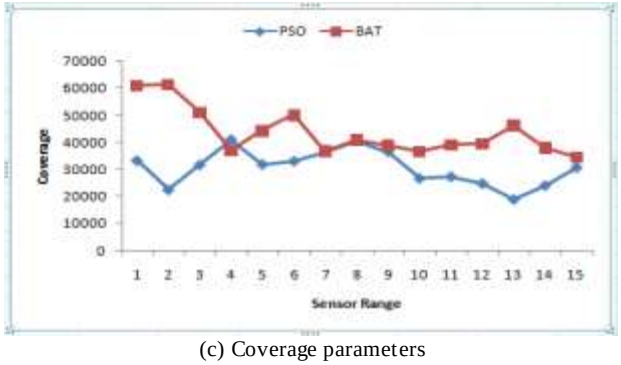
B. *Comparisons of Simulation Result among PSO and BAT Algorithms with varying Agents*



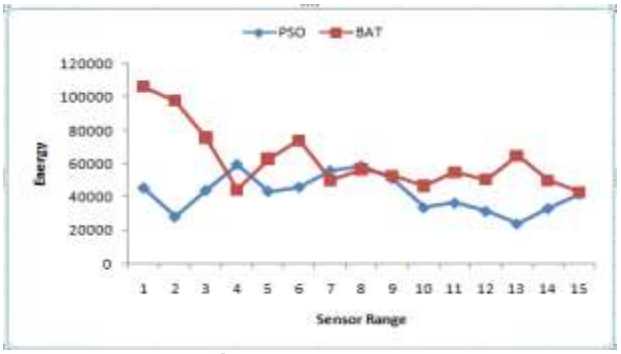
(a) Move parameters



(b) Time parameters



(c) Coverage parameters



(d) Energy parameters

Fig 2: Plot of parameters v/s sensor range at agents = 12

In this section the various plots (figure 2) and tables (table 2 to 5) of the simulation for the multi-agent based movement is being provided and is being compared with PSO and BAT Algorithm for four parameters namely moves, time, coverage and energy in unknown environments.

The graph and tables are shown the fastest algorithms are BAT with respect to PSO for all the parameters.

Sensor Range	PSO	BAT	PSO	BAT	PSO	BAT	PSO	BAT
1	275812	535758	146808	190140	148652	303500	6965.6	232186
2	228956	460058	90208	248616	6958.0	115930	4317.8	213844
3	205456	254600	98406	208068	6688.8	259592	6724.6	165382
4	193904	209800	77406	110742	5442.4	220576	9184.4	9671.6
5	271578	288264	100212	298574	6919.2	160130	6700.6	137742
6	118682	202462	120328	235648	7131.0	129198	7010.4	161524
7	197402	255804	114784	275634	8355.0	9992.6	8470.8	108380
8	285828	252948	61574	186650	7263.6	8222.0	8998.2	123430
9	136618	258486	65768	144756	6942.8	163294	7812.8	115332
10	137210	269020	105486	137052	4374.8	178038	5154.4	101970
11	258406	379160	119028	117904	5927.2	9680.2	5584.6	119504
12	7521.2	193078	6754.0	195448	5658.6	113654	4840.0	108778
13	134966	309080	4578.2	163858	7110.2	115178	3680.6	141578
14	109196	389756	60766	139002	5952.4	103126	5089.2	108694
15	30968	260800	65228	142846	6419.0	136070	6487.6	93256

Table 2: Move analysis for varying number of Agents

Sensor Range	PSO	BAT	PSO	BAT	PSO	BAT	PSO	BAT
1	3.61958	5.26346	3.87440	3.63794	5.85540	8.58072	3.59384	8.76602
2	2.99840	4.47614	2.36830	4.69902	2.69680	3.28796	2.24196	8.04464
3	2.70064	2.43146	2.57816	3.96962	2.59674	7.25252	3.48384	6.15672
4	2.53672	2.04956	2.01132	2.08496	2.14964	6.20798	4.81106	3.58094
5	3.55198	2.83824	2.62526	5.62170	2.70554	4.54774	3.49230	5.14338
6	1.58174	1.93224	3.22126	4.49142	2.79388	3.61758	3.64326	6.02996
7	2.59010	2.48994	3.01214	5.16462	3.24360	2.81930	4.40398	4.04006
8	3.74566	2.44718	1.65066	3.48398	2.82590	2.31908	4.66866	4.62600
9	1.80430	2.48526	1.71902	2.71130	2.67730	4.55960	4.09700	4.29338
10	1.77828	2.70606	2.78532	2.56736	1.68166	5.05396	2.63668	3.81886
11	3.36814	3.73230	3.14578	2.23144	2.26646	2.72914	2.95456	4.45018
12	0.97998	1.85946	1.76644	3.64134	2.19034	3.20658	2.47640	4.04100
13	1.76776	2.98266	1.19044	3.08738	2.76210	3.19524	1.89368	5.29670
14	1.41560	3.75938	1.62176	2.62462	2.26726	2.88006	2.65666	4.10056
15	0.39992	2.49680	1.69116	2.67164	2.52342	3.79870	3.42340	3.45838

Table 3: Time analysis for varying number of Agents

Sensor Range	PSO	BAT	PSO	BAT	PSO	BAT	PSO	BAT
1	892954	981090	549284	562310	563480	836642	332980	610506
2	734658	914192	416202	674486	340568	384534	225204	613600
3	696906	668360	430828	612812	308916	670616	316904	511372
4	751910	645888	366432	402408	273232	607410	409640	370150
5	956346	726640	426408	763092	315760	473096	318710	442324
6	464348	548782	475142	711542	331050	438472	330578	500346
7	663882	745752	487848	738622	378266	373760	365382	367172
8	858592	667302	291262	579560	342154	296934	406822	408164
9	519384	702930	313364	493596	341672	514140	365738	387994
10	541344	719814	440722	480522	222446	528820	267638	365100
11	900174	756728	504528	418482	289042	356500	273784	390658
12	348144	530276	314116	614496	265256	378208	248486	394550
13	487522	777516	233552	522518	347880	419004	189776	462428
14	479628	843672	284928	438778	279372	389468	241110	379440
15	170818	728584	306012	472164	300022	437968	307730	344826

Table 4: Coverage analysis for varying number of Agents

Sensor Range	PSO	BAT	PSO	BAT	PSO	BAT	PSO	BAT
1	1792862	2441558	946664	86682.8	964412	1385800	454100	1063870
2	1489796	2098474	589338	1135102	453650	53185.4	282384	98115.4
3	1326748	1160776	635518	95220.4	436588	1186804	439744	75585.0
4	1264342	96090.8	508156	50921.4	354314	1005294	593400	44386.0
5	1770180	1312278	655178	1353860	451284	72654.2	433362	62900.4
6	76601.0	93123.0	788284	1082828	467350	59209.2	458900	74009.6
7	1272088	1171302	749006	1255904	54111.0	45654.0	555852	49662.0
8	1848462	1153094	401908	84715.8	475958	37708.4	584448	56460.6
9	88725.4	1176614	428586	66402.0	458864	74570.6	508644	52790.0
10	89106.4	1223840	685554	62709.8	284054	81058.8	336104	46514.0
11	1674176	1726724	776710	54084.8	388814	44198.8	364276	54693.8
12	48586.8	87862.0	443346	88998.8	368134	52542.2	315036	50317.2
13	87937.4	1411522	299616	75159.6	464078	52990.6	240842	64630.0
14	71661.4	1772634	391628	63194.6	392834	47100.8	331404	49773.8
15	20387.4	1192196	425404	65311.0	414834	62015.6	417590	42846.4

Table 5: Energy analysis for varying number of Agents

V. CONCLUSIONS

In this paper, we particularly interested use PSO and BAT algorithms in multi robot path planning and area exploration. Path planning is the problem where objective is to reach from source to target without collide the obstacle. The fastest algorithm is BAT with respect to PSO in unknown environment.

REFERENCES

[1]. Zhang Y., Gong D. W. & Zhang J. H. (2013). Robot path planning in uncertain environment using multi-objective particle swarm optimization. Neuro-computing 103, 172-185.

[2]. Lu F. & Milios E. (1997). Robot pose estimation in unknown environments by matching 2d range scans. Journal of Intelligent and Robotic Systems 18(3), 249-275.

[3]. Mahajan P. B. & Marbate P. (2013). Literature review on path planning in dynamic environment. International Journal of Computer Science and Network 2(1), 115-118.

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

[5]. Kala R., Shukla A. & Tiwari R. (2012). Robotic path planning using hybrid genetic algorithm particle swarm optimisation. International Journal of Information and Communication Technology 4(2-4), 89-105.

[6]. Chakraborty J., Konar A., Chakraborty U. K. & Jain, L. C. (2008, June). Distributed cooperative multi-robot path planning using differential evolution. In Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on (pp. 718-725). IEEE.

[7]. Geng N., Gong D. W. & Zhang Y. (2014). PSO-based robot path planning for multi-survivor rescue in limited survival time. Mathematical Problems in Engineering 2014.

[8]. Banks A., Vincent J. & Anyakoha C. (2007). A review of particle swarm optimization. Part I: background and development. Natural Computing 6(4), 467-484.

[9]. Ercan F. M. & Li X. (2013). Particle swarm optimization and its hybrids. International Journal of Computer and Communication Engineering 2(1), 52-55.

[10]. Dai S., Huang H., Wu F., Xiao S. & Zhang T. (2009, November). Path planning for mobile robot based on rough set genetic

algorithm. In 2009 Second International Conference on Intelligent Networks and Intelligent Systems (pp. 278-281). IEEE.
[11]. Jiang Y., Hu T. Huang C. & Wu X. (2007). An improved particle swarm optimization algorithm. Applied Mathematics and Computation 193(1), 231-239.
[12]. Nasrollahy A. Z. & Javadi H. H. S. (2009 November). Using particle swarm optimization for robot path planning in dynamic environments with moving obstacles and target. In Computer Modeling and Simulation 2009. EMS'09. Third UKSim European Symposium on (pp. 60-65). IEEE.
[13]. Nakamura R. Y., Pereira L. A., Costa K. A., Rodrigues D., Papa J. P. & Yang X. S. (2012 August). BBA: A binary bat algorithm for feature selection. In Graphics Patterns and Images (SIBGRAPI) 2012 25th SIBGRAPI Conference on (pp. 291-297). IEEE.
[14]. Fister Jr I., Fister D. & Yang X. S. (2013). A hybrid bat algorithm. arXiv preprint arXiv:1303.6310.