ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

A Comparison between Bat Algorithm and Cuckoo Search for Path Planning

Yogita Gigras[1], Kusum Gupta[2], Vandana[3], Kavita Choudhary[4]

Assistant Professor, Dept. of CSE, ITM University, Gurgoan, Haryana, India[1]

Assistant Professor, Dept. of CSE/IT, Banasthali University, Banasthali, Rajasthan, India[2]

M.Tech Scholar, Dept. of CSE/IT, Banasthali University, Banasthali, Rajasthan, India[3]

Assistant Professor, Dept. of CSE, ITM University, Gurgoan, Haryana, India[4]

ABSTRACT: Here, metaheuristic search techniques are applied for robotic path planning. Robot has to move from its starting point called source point to final point called as destination point with minimum number of moves and iteration. Here both cuckoo search and bat algorithms are applied for the proposed problem and simulation results are compared. The techniques are applied for different number of population and bat algorithm provide better results as compared to cuckoo search.

KEYWORDS: Metaheuristic, cuckoo search, levy flight, bat algorithm.

I. INTRODUCTION

To solve any particular problem heuristic techniques were used. As these techniques do not provide any optimal solution so they were categorized into problem dependent techniques. Heuristic techniques were greedy in nature, that's why they were trapped into local optima and unable to provide global optimum solution. Metaheuristic approach is opposite to heuristic because of its problem independent nature. Metaheuristic algorithms provide current and global best solution in search space. It provides a good balance between local and global solution and also improve the convergence of algorithm.

For robotic path planning metaheuristic search algorithm is better technique because it is problem independent. It uses random solutions to achieve the goal. It uses hit and trial method to reach the goal.
There are many metaheuristic search techniques and classified with their properties i.e. based on swarm intelligence: Firefly algorithm, Ant colony optimization, particle swarm optimization, and on the basis of solitary intelligence: bat algorithm and cuckoo search etc. Solitary intelligence based techniques are used for robotic path planning.

Robots have to reach their destination with minimum moves and with avoiding obstacles. Path for robots can be static or dynamic. If it is static (fix), the path is known to robots. And if path is dynamic (not fixed), the robot have to choose its own path from source to destination. As path can be static or dynamic, the obstacles can be static or dynamic. If the obstacles are static means they are fixed or they remain fixed during the whole path. If the obstacles are dynamic then every time they will be at different position and robot has to identify to avoid during choosing the required path.

Robots are used, specially where human beings are unable to work or have their life risk. There are many areas where robots are used such as military, medical, mining, education, daily routine tasks etc. Recently robotic path planning is widely used in research area. These days robots are programmed according to their task. They can do many complex tasks which are very difficult for human beings to do. Metaheuristic search techniques are used where global optimization in needed. They does not guaranteed for optimal solution. Cuckoo search and bat algorithms are the hottest topic of metaheuristic search technique for path planning. Both are based on solitary intelligence because a single agent is required for its purpose. Cuckoo Search (CS) is based on blood parasitism behaviour of some cuckoo species where bat algorithm is based on hunting behaviour of bats.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

II. RELATED WORK

Srhan et al.[3] proposed cuckoo search for" Traveling Salesman problem" for 20 to 442 cities and found optimal path that reduce the cost and time both. Mohanna et al. [4] applied improved cuckoo search for their problem of feed forward neural network training and found overall better performance in all criteria. Moe Zaw et al. [5] also applied improved cuckoo search with levy flight for a text based web document clustering system and got better performance.

Shukla et al.[6] implemented " Adaptive & Discrete Real Bat Algorithms for Routing Search Optimization of Graph Based Road Network". For their purpose they implement the three variable dependent Weibull Cumulative Distribution Function as Weibull Coded Binary Bat Algorithm (WCBBA), another as Real Bat Algorithm (RBA) and the third as the hybrid of the two. After implementation they conclude it new discrete version of bat algorithm exclusively for graph based problem for first time and used for route determination in a graph network. Xin-she Yang [16] introduced "A new Metaheuristic Bat-Inspired Algorithm". In this paper he tried to formulate a new nature inspired algorithm, based on the echolocation behavior of bat. He combines the advantages of existing algorithm into a new algorithm. Proposed algorithm was much simpler comparative to existing algorithms. It has advantages over existing algorithms someway. Fister et al.[17] proposeded "A Hybrid Bat Algorithm". In this paper they present a new swarm intelligence algorithm, based on bat algorithm. They hybridized the bat algorithm with differential evolution strategies. They tested on large-scale global optimization and as a result they found significant improvement.

III. EXISTING SYSTEM

A. ***Cuckoo search:***

i. Breeding behavior :

CS is also nature inspired algorithm. Cuckoos are extremely interesting birds because they lay their eggs in the nests of small songbirds whose eggs are also looking same. They lay their eggs either just before the host bird lays or just after it, so that host bird can't identify these. If the host bird identifies that the eggs are of another bird than the host bird it destroys these or build a new nest, if not then eggs grow up with host eggs. They lay their eggs same as host bird eggs. As first egg become chick, it pushes the host eggs outside the nest and share the host bird food. According to some studies cuckoo chicks can also copy their call same as call of host chicks to get opportunity of feeding. Because they lay their eggs into another nest, that's why they are also called parasitism. Intraspecific blood parasitisms are those birds that destroy the host bird eggs.

ii. Levy flight:

Cuckoo search uses levy flight instead of simple walk. Levy flight is random walk whose step length is drawn from levy's distribution. Levy flight is used to generate new solution by calculating the step size of any walk. To generate new solution the following Mantegna's algorithm is used:

$$\sigma^2 = \left\{ \frac{\Gamma(1+\beta) Sin\left(\frac{\Pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right)\beta\, 2^{\frac{\beta-1}{2}}} \right\}^{1/\beta} \quad (1)$$

here Γ is gamma function which is extended from factorial function of positive real number. β is the variable which control distribution *by* $0 \leq \beta \leq 2$. σ^2 is the variance. Variance is used to calculate step size of random walk by using the following equation:

$$\text{step size} = \left(0.01 \times \left(\frac{u}{|v|^{1/\beta}}\right) \oplus (X_i^t - X_{best})\right) \quad (2)$$

here 0.01 factor for controlling step of cuckoo step, X_i^t is current solution of cuckoo i at iteration t, X_{best} is the global best solution. here $\oplus$ entrywise multiplication.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Here following three rules are considered for cuckoo search:

1) At a time only one egg is laid by each cuckoo into randomly chosen nest;
2) For next generation only high quality eggs are selected and marked as best nest;
3) Total numbers of available host nests are fixed, and assumes a probability ***pa* ϵ *[0, 1]*** for discovering the cuckoo eggs by host bird. If the host bird identifies that egg are of another bird then either throw the egg away or abandon the nest, and build a completely new nest. Here we assume last assumption that is approximated by the fraction pa of the n nests is replaced by new nests (with new random solutions) [1].

Based on these above rules, the implementation can be done by considering the following equations. Only one egg is laid into one nest by each cuckoo. Each nest represents a solution. The best nest is chosen among nests for best solution. The new solution for is generated for cuckoo i by using the following equation:

$$X_i^{(t+1)} = X_i^t + \alpha \times \text{levy}(\lambda) \quad (3)$$

Where $\mathbf{x_i^{t+1}}$ is the new solution for cuckoo ***i*** at $\boldsymbol{t^{th}}$ iteration. Here $\boldsymbol{x_i^t}$ is the previous solution and $\boldsymbol{\alpha}$ denotes the step size that should be related to scale of problem of interest and should be greater than zero. In most cases, $\boldsymbol{\alpha = 1}$ is used because if step length will be too large then the next solution will also be too far from the current solution and if the step length is too short then the solution will be too near that will be not efficient for good solution. ***Levy* (λ)** is the random walk. Its step length is calculated from levy distribution for levy flight.

$$\text{Levy} \sim u = t^{-\lambda} \text{ and } (1 \leq \lambda \leq 3) \quad (4)$$

B. ***Bat Algorithm***

Xin-She Yang developed a new nature inspired algorithm “Bat Algorithm”[2010]. Microbats have special quality “sonar” to detect their chase and obstacles. They can also detect the cracks for their nest even in the dark. Microbats emits the sound pulse and wait for reverberate of sound that comes back after striking the chase or obstacles. Usually microbats use pulse rate i.e. 10-20 times per second. They can also detect tiny obstacles like human hair [13]. For navigation they use time delay between emissions and reverberate of sound. They can also detect distance, type of chase, its moving speed. They are able to differentiate targets by variations of Doppler effect introduced by the wing flutter rates of the target insects. They combine all their senses so that they can easily detect the chase and its navigation. They use [0.7,17] mm range of wavelength or [20-500] kHz inbound frequencies [14].

To develop algorithm the following three rules are considered:

1. Echolocation is used by all the bats to sense the distance, chase, food and obstacles.
2. Bats are used to fly randomly with some velocity vel_i at position S_i with fixed frequency $[feq_{max}, feq_{min}]$, varying wavelength λ , pulse rate rϵ[0,1] and loudness A.
3. Loudness varies from A_{max} to $A_{min.}$ A[0,1].

Microbats use varying loudness and frequency while their velocity, frequency and position remains fixed. Frequency can be varying corresponding to pulse rate and pulse emitted.

i. Movement of virtual bats:

For experiment, rules are updated with the frequency feq, position x_i and velocity vel_i in d-dimensional search space. The corresponding updated solutions for S_i^t and velocity vel_i^t at time step t are represented as:

$$feq_i = feq_{min} + (feq_{max} - feq_{min})U(0,1), \quad (5)$$

$$vel_i^t = vel_i^{t-1} + (x_i^t - best)feq_i, \quad (6)$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

$$S_i^t = S_i^{t-1} + vel_i^t, \quad (7)$$

here U(0,1) is used for random distribution, best is used for current global best location that is calculated by comparing all the solutions among all the bats. The current best solution is updated by using the following equation:

$$S_{new} = S_{old} + \boldsymbol{\varepsilon} A^t \quad (8)$$

where $\boldsymbol{\varepsilon}\epsilon[-1,1]$ is a random number between -1 to 1, and A^t is used for average loudness of all the bats at time step t.

IV. PROPOSED WORK

For path planning is planned from an initial point called 'source point' to final point called 'destination point' with fixed step size and with avoiding obstacles encountered during its path. The path selected should be optimal path.

a. Design Considerations:

For robotic path planning, we consider **X1**, **Y1** as source point with values 10 and 0 respectively from where robot has to start its path. We consider **X2, Y2** as destination point for robot with the values 100 and 100 respectively. The rectangular shape represents the source point and diamond shape represents the destination point in ***fig.1***. Here 20 static obstacles are used between source and destination point. Robot starts its move for with angle,' $\boldsymbol{\theta}$', and distance ***d***, where theta and distances are:

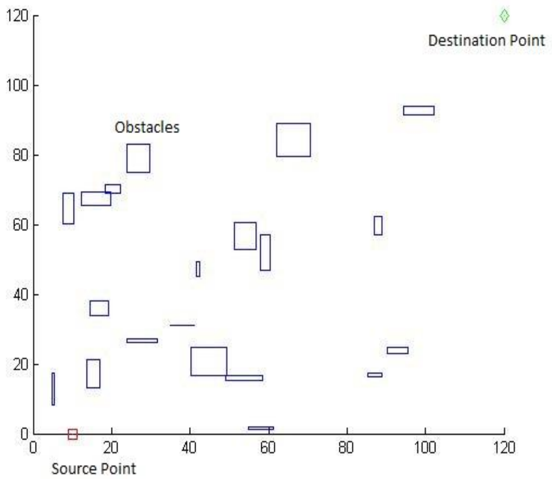

Fig.1. Twenty rectangular obstacles

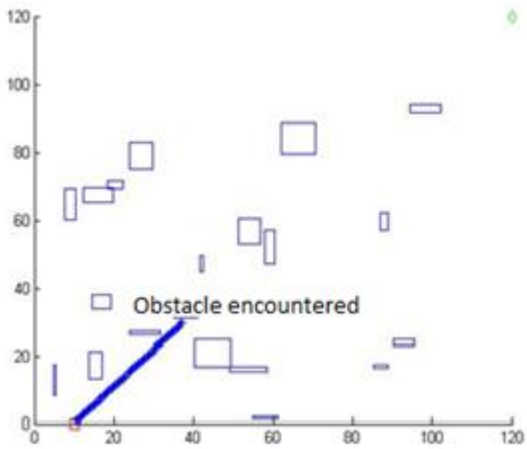

Fig.2. first obstacle encountered.

$$\theta = \tan^{-1}\left(\frac{|y2-y1|}{|x2-x2|}\right) \quad (9)$$

$$d=\sqrt{(x2-x1)^2 + (y2-y1)^2} \quad (10)$$

and with each move update its value of ***x1*** and ***y1***, by considering the following equation:

$$x1 = x1 + step \times \cos(\theta) \quad (11)$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

$$y1 = y1 + step \times \sin(\theta) \quad (12)$$

until any obstacle encountered or ***distance*** becomes zero. Here ***step*** is considered as 1. Robot encountered an obstacle at ***(X1, Y1)*** with values (37.70, 30.22) as shown in fig.2.

b. *Description of the Proposed Solution:*

As robot encounter the obstacle, it has to take some step back and here the proposed solution is applied for path planning.

i. Path planning using cuckoo search:

Here for cuckoo search the initial value of nest is initialized. Then the distance is calculated between the current and final position. Then minimum distance is calculated. According to minimum distance, select the best value of nest and declare it current position of robot. Then the value of sigma is calculated using the equation (1). Here the main process starts with calculating the cuckoo with levy flight, stepsize, and find the new solution until stopping criteria meet. Calculate the updated distance with new solution and compare both previous and updated distance. If updated distance is less than previous one then replace the value of d with updated distance if it is not then consider it worst solution and accept the previous solution. The process remains same until destination point achieved by it.

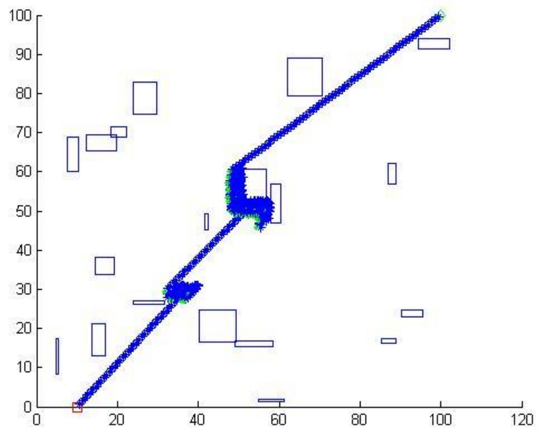

Fig 3 Path covered by robot using Cuckoo Search

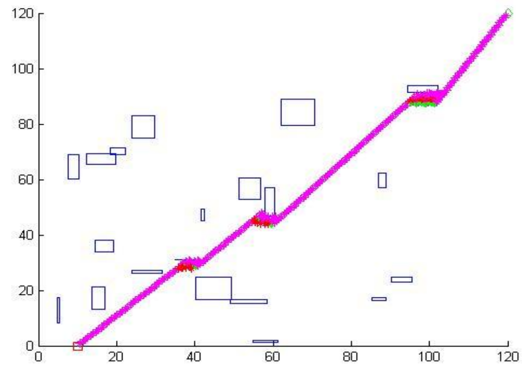

Fig 4. Path covered by robot using Bat Algorithm

ii. Bat Algorithm:

Bat algorithm is also applied for path planning. As obstacle encounters, robot has to start and has to take some step back and here bat algorithm is applied. Define the bat population (n), pulse rate (r), frequency (f) , loudness(A) and calculate the distance (d). Choose the best value of X1,Y1 CORRESPONDING TO minimum value of distance. Now generate modify the frequency with equation(5), update velocity and current position with equation (6),(7) respectively. Update the distance with current best position. If pulse rate is less than random then choose the solution among the best solution. Find out the local best solution by using equation(8).

Now generate the new solution. If random value is less than loudness and updated distance is greater than previous distance then accept the new solution as best solution. The process remains same for each encountered obstacle until stopping criteria meet.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

V. PSEUDO CODE

A. Pseudo Code for Cuckoo Search

Begin

1) *Choose the initial values of nest/cuckoos, say, x1, y1.*
2) *Measure the distance between initial and target values, say, d.*
   a. $d=\sqrt{(target\ value\ (x) - initial\ value\ (x))^2 + (target\ value(y) - initial\ value(y))^2}$
3) *Find the minimum value of d, say min (d).*
4) *Choose the best value of x1 and y1 according to value of d.*
5) *Calculate the value of sigma by using eq(i)*
6) *While(a<max iteration)or(stopping criterion)*
   i. *Evaluate cuckoo randomly by levy flight*
   ii. *Calculate its stepsize by using eq(2) and*
   iii. *new solution by using eq(3), say updated x1, y1.*
   iv. *Calculate the new updated distance with new solution, say, updated (d).*
   v. *Compare the distance*
   vi. *If (updated(d) < d) then*
      1. *Replace the value of d with updated(d)*
   vii. *Else*
      1. *Consider it worst solution and destroy these values.*
   viii. *End if*
7) *End while*

End

B. Pseudo Code for Bat Algorithm

Begin

1) *Define the bat population* S_i *(i = 1, 2, ..., n) and* vel_i
2) *Initialize frequency* feq_i *at* S_i
3) *Define initial pulse rates* r_i *and the loudness* A_i
4) *Measure the distance between initial and target values, say, d.*
5) $d=\sqrt{(target\ value(x) - initial\ value(x))^2 + (target\ value\ (y) - initial\ value\ (y))^2}$
6) *Find the minimum value of d, say min (d).*
7) *Choose the best value of x1 and y1 according to value of d.*
8) *while (stopping criteria)*
   a. *Adjust frequency for new solutions ,by using eg(5)*
   b. *update velocity and location/solution, by using eq (6) and (7) respectively*
   c. *calculate the updated distance with updated values,*
   d. *if (rand > pulse rate)*
      i. *Choose the best solution.*
      ii. *Select a local solution around best solution, by using eq(8)*
   e. *end if*
   f. *Find new solution with random flying*
   g. *if ((rand < loudness) & (d <updated(d)))*
      i. *Consider the new solutions*
      ii. *Do some increment for* r_i *and reduce* A_i
   h. *end if*
9) *end while*

End

VI. SIMULATION RESULTS

After applying proposed pseudo code of both cuckoo search and bat algorithm with different number of population and the results (number of iterations with different population) came from these are shown in table I and II. Bat algorithm found its best solution in second iteration while cuckoo search provide results in 2-3 iterations.

Table 1. simulation results for cuckoo search and bat algorithm

No. of population	No. of iteration		Complexity	
	Cuckoo search	Bat algorithm	Cuckoo search	Bat algorithm
10	2	2	20	20
20	3	2	30	20
30	2	2	20	20
40	2	2	20	20
50	2	2	20	20
60	2	2	20	20
70	2	2	20	20
80	3	2	30	20
90	2	2	20	20
100	3	2	30	20

When same algorithms are applied for proposed problem i.e. path planning then bat algorithm provide optimal path with minimum number of iterations as compared to cuckoo search as shown in Table 2. Complexity is compared for both bat and cuckoo search in terms of:

$$\textbf{Complexity} = \textbf{Number of population} \times \textbf{Number of iterations} \qquad (13)$$

Table 2. Simulation results for path planning using cuckoo search and bat algorithm

No. of population	Number of iterations		Complexity	
	Bat Algorithm	Cuckoo search	Bat algorithm	Cuckoo search
10	213	1083	2130	10830
20	442	478	8840	9560
30	338	555	10140	16650
40	242	1499	9680	59960
50	204	977	10200	48850
60	294	1556	17640	93360
70	213	1256	14910	87920
80	278	765	22240	61200
90	212	1238	19080	111420
100	260	687	26000	68700

As table 1 and 2 represents the complexity for both algorithms bat algorithm provide better results as compared to cuckoo search. For different number of population bat algorithm gives less complexity and as a conclusion bat algorithm is better algorithm as compared to cuckoo search in terms of iterations and complexity.

Here these algorithms are applied for static obstacles. Robot starts its journey from source, it does not penetrate the obstacle, it by-passes the obstacles encountered during its path, and reaches to final destination with minimum number of iterations. Bat algorithm provide optimal path between source and destination with minimum number of iterations.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

VII. CONCLUSION AND FUTURE WORK

Bat Algorithm and Cuckoo search provide optimal solution as compared to previous techniques used for path planning. Between Bat and Cuckoo search Bat algorithm provides better results in terms of iterations and elapsed time as shown in table 1 and 2 in previous section. The same codes are executed more than 200 times. With each execution Bat algorithm provide optimal path as compared to cuckoo search whether they follow the same path or different path. In both cases Bat algorithm is better option for robotic path planning in static environment.

A combination of these techniques can be implemented for future work and better results. Here Path planning is done for static environment. For future work this can be a challenging topic for planning a path for dynamic environment. Here planning is done for 2-dimensional search space. For future work this can be implemented for 3-dimensional search space by adding Z-coordinate with X and Y coordinates. Sensor can be used to detect the obstacle.

REFERENCES

1. S. Deb, Xin-She Yang” *Cuckoo Search via Lévy flights*”, World Congress on Nature & Biologically Inspired Computing,IEEE publications, pp.210-214, 2009.
2. Jati, G.K..,” Discrete cuckoo search for traveling salesman problem”, 7th International Conference on Computing and Convergence Technology, pp. 993-997, 2012.
3. Ala'a Abu-Srhan and Essam Al Daoud,” A Hybrid Algorithm Using a Genetic Algorithm and Cuckoo Search Algorithm to Solve the Traveling Salesman Problem and its Application to Multiple Sequence Alignment”, International Journal of Advanced Science and Technology, vol.61, pp. 29-38, 2013.
4. Ehsan Valian ,” Improved Cuckoo Search Algorithm for Feed forward Neural Network Training”, International Journal of Artificial Intelligence & Applications, vol.2, pp.36-43, 2011.
5. Ei Ei mon, Moe Moe Zaw, “A Text-based Web Document Clustering System by Using Improved Cuckoo Search Clustering System Based on Levy Flight”, International Conference on Advances in Engineering and Technology, pp. 63-67, 2014.
6. Mehmet Karamanoglu et al.,” Cuckoo Search for Business Optimization Applications”, National Conference on Computing and Communication Systems, pp.1-5, 2012.
7. S. Deb, X. S. Yang, “Engineering optimization by cuckoo search”, Int. J. Mathematical Modelling and Numerical Optimisation, vol. 1, 2010, pp. 330-343.
8. Khairul Najmy ABDUL RANI et al.,” Nature-inspired Cuckoo Search Algorithm for Side Lobe Suppression in a Symmetric Linear Antenna Array”, RADIOENGINEERING, VOL. 21, pp.865-874, 2012.
9. Ehsan Valian et al.,” Improved Cuckoo Search Algorithm for Global Optimization”, International Journal of Communications and Information Technology, vol.1, pp.31-44, 2011.
10. Akajit Saelim et al., “Migration Planning using Modified Cuckoo Search Algorithm”, International Symposium on Communications and Information Technologies,IEEE publications,pp.621-626,2013.
11. Prabhneet kaur, Taranjot kaur,”A Comparative Study of Various Metaheuristic Algorithms” International Journal of Computer Science and Information Technologies, vol. 5, pp. 6701-6704,2014.
12. Hanif Halim, Ismail,” Bio-Inspired Optimization Method: A Review”, International Journal of Information Systems, vol. 1, pp.12-17, 2014.
13. Xin-She Yang, ”A New Metaheuristic Bat-Inspired Algorithm”.
14. Duˇsan Fister et al., “A Hybrid Bat Algorithm”, ELEKTROTEHNIˇSKI VESTNIK 80,ORIGINAL SCIENTIFIC PAPER, pp. 1-7, 2013.
15. Adis ALIHODZIC, Milan TUBA, “Framework for Bat Algorithm Optimization Metaheuristic”, Recent Researches in Medicine, Biology and Bioscience, pp.157-162
16. De-yu Zhang and Zhi-guo Liu, “Study on Improved Algorithm for Mobile Agent Migration Path”, Third International Conference on Intelligent Network and Intelligent Systems, 2010.
17. Masayuki Higashino, Kenichi Takahachi, Takao Kawamura, and Kazunori Sugahara, “Mobile Agent Migration Based on Code Caching”, 26th International Conference on Advanced Information Networking and Applications Workshops, 2012, pp. 651-656.