International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 02 - Issue 11 || November 2017 || PP. 26-31

Enhanced Bat Inspired Algorithm for Solving Travelling Salesman Problem

M. Sornam[1]
[1]*(Department of Computer Science, University of Madras, India)*

Abstract: Bat-Inspired Algorithm is evocative meta-heuristic algorithm planted on the echolocation behavior of bats. The proficiency of echolocation of bats can find their quarry in complete darkness. This Paper presents the enhancement of Bat-Inspired Algorithm by combining particle swarm optimization (PSO) for solving travelling salesman problem (TSP). This new algorithm improves the veracity of finding the nearest best solution and reduces the computational cost.
Keywords: Bat inspired algorithm, particle swarm optimization, Travelling Salesman problem

I. Introduction

Nature Inspired new meta-heuristic search algorithm called Bat-Inspired Algorithm was developed by Xin-She Yang in 2010. The method which is used to solve the conjunctional problem near-optimally in bounded time is called heuristic. A heuristic can also give the optimal solution on some trails. There is a proof [1] that it will give optimal solution for all the problems and for all trials on each problem. While conveying a heuristic, one should always aim to attain the objective of the problem of interest as much as possible in terms of solution accuracy and add steps minimally and efficaciously such that the heuristic takes least time to solve the problem. Travelling salesman problem in which a salesman has to instigate his tour from his base city exactly once and retaliate to his base city. In this process, his organization's objective may be to minimize the total distance of his travel or total time of his travel. The Travelling salesman problem comes under combinatorial pigeonhole; development of a heuristic for this problem is inevitable.

In this paper, we intend to propose a new meta-heuristic method, namely, the Bio-inspired Bat algorithm combined with the particle swarm optimization (PSO), based on the echolocation behavior of bats. The potency of echolocation of micro bats is enthralling as these bats can find their quarry and discrepant different types of insects even in complete darkness [1]. The main aim of the paper is to solve travelling salesman problem (TSP) using enhanced bat algorithm.

First formulate the bat algorithm with PSO by idealizing the local best solution, global best solution and echolocation behavior of bats. Then describe how it works and make comparison with different benchmark function with other existing algorithms.

The paper is organized as follows. In the next section, particle swarm optimization algorithm is explained. In section 3, bat algorithm is discussed. In section 4, proposed algorithm is narrated. In section 5, experimental results are illustrated. Finally, the conclusions are given in section 6.

II. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a technique introduced by James Kennedy and Russell C. Eberhart in 1995[6]. It is used to examine the search space of a given problem to find the framework or parameters required to maximize a vital objective. It originates from two separate conceits: The idea of the swarm intelligence based on the observation of swarming habits by certain kinds of animals and the field of evolutionary computation.

PSO Algorithm

The Particle Swarm Optimization algorithm (PSO) is a novel population-based putative search algorithm and a preference solution to the complex non-linear optimization problem. The PSO algorithm basically erudite from animal's activity or behaviour to solve optimization problems. In PSO, each member of the population is called a swarm. Originating with a randomly initialized population and moving in randomly chosen trajectory, each particle goes through the penetrating space and remembers the best previous positions by itself and its neighbours. Particle of a swarm confabulate good positions to each as well as dynamically acclimate their own position and velocity derived from the best position of all particles. The next step begins when all particles tend to fly towards better and better positions over the searching mechanism until the swarm move closer to an optimum of the fitness function [7].

Fitness function:

$$f(Si) = \quad 1/\sum_{i=1}^{N} d[C_{n(i)}\,,\ \ C_{n(i+1)mod\ N}] \tag{1}$$

Framework of a PSO Algorithm

Initialize population of particles
do
  for each particle p with position x_p(fitness value) do
    if (x_p is better than $pbest_p$) then
      $pbest_p \leftarrow x_p$
    end_if
  end_for
  Define $gbest_p$ as the best poition found so far by any of p's neighbors
  for each particle p do
    $v_p \leftarrow$ Compute_velocity(x_p,$pbest_p$,$gbest_p$)
    $x_p \leftarrow$ update_position(x_p,v_p)
  end_for
while(target is not reached)

Every particle has a fitness value which is determined by target function and it has also a celerity which determines its destination and distance. All particles search in the solution space for their best positions and the positions of the best particles in the swarm. PSO is initially a group of random particles, and then the optimum solutions are found by perennial searching. In every iteration, a particle will follow two best to renew itself; the best position found for a particle called p_{best}; the best position found for the whole swarm called g_{best} .

$$v_p(t)= wv_p(t-1)+c_1.rand_1.(pbest_p(t-1) - x_p(t-1)) + c_2.rand_2.(gbest_p(t-1)-x_p(t-1)) \tag{2}$$

$$x_p= x_p(t-1)+v_p(t) \tag{3}$$

All particles will determine following steps through the best experiences of individuals themselves and their companions. Finally it reaches the destination. The advantages of PSO are that the convergence speed of the swarm is very high when comparing with GA.

III. Bat Algorithm

Meta-heuristic Bat-Inspired Algorithm was developed by Xin-She Yang in 2010. Some of the idealized rules are:

1. All bats use echolocation to intuit distance, and they also 'know' the difference between quarry/prey and background barricades in some magical way;
2. Bats fly randomly with velocity v_i at position x_i with the fixed frequency f_{min}, varying wavelength λ and loudness A_0 to search for quarry. They can automatically adjust the wavelength (or frequency) of their emitted pulse and adjust the rate of pulse emission $r \in [0,1]$, depending upon the proximity of their target.
3. Although the loudness can vary in many ways, we presume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min}[1].

First, initializing bat population: position x_i, velocity v_i and frequency f_i, The movement of the virtual bats is given by updating their velocities and positions at time step t.

$$f_i = f_{min} + (f_{max} - f_{min}) \beta, \tag{2}$$

$$v_i^t = v_i^{t-1} + (x_i^{t+1} - x_*) f_i \tag{3}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{4}$$

where β ∈ [0,1] denotes a randomly generated vector from uniform distribution. The variable x_* denotes the current global best location (solution), which is located after comparing all the solutions provided by the m bats.

After selection of one solution among the current best solutions of bat, a random number is applied; if its random number upper to pulse emission rate r_1, a new solution will be accepted around the current best solution; it can be represented by,

$$x_{new} = x_{old} + \varepsilon A^t \quad (5)$$

where $\varepsilon \in$ [-1, 1] is a random number, while $A^t = < A_i^t >$ is the average loudness of all the bats at current generation.

IV. Loudness and Pulse Emission

Moreover, the loudness A_i and the pulse emission rate r_i will be updated, and a solution will preferred if a random number is less than loudness A_i and $f(x_i)<f(x_*)$. A_i and r_i are updated by

$$A_i^{t+1} = \alpha A_i^t,\ r_i^{t+1} = r_i^0[1 - exp(-\gamma t)] \quad (6)$$

where α and γ are constants. For any 0 < α < 1 and γ > 0, we have

$$A_i^t \longrightarrow 0\ ,\ r_i^t \longrightarrow r_i^0\ , \text{as t} \longrightarrow \infty. \quad (7)$$

Loudness and pulse emission will be updated only if the new solutions are ameliorated. The algorithm is repeated until maximal numbers of cycles is fulfilled.

V. Enhanced Bat Algorithm

Bats are naturally using a type of radar system by emitting ultrasonic sounds in a certain frequency and hearing the echoes of these sounds. These sounds enable them to locate and avoid the object in their path as well as reach the quarry. Exploration ability of the bat algorithm is favorable. But, it needs high number of monotony to lead to a desirable solution. Here, with defining dynamic θ scale factor parameter (a factor that limits the step sizes of random walks for local search), an amelioration is proposed for this algorithm. Proper tuning of this parameter reduces the number of the iterations hence the computational time. In addition, this improvement can provide easy adjustment of the bat algorithm for discrete optimization problems.

Dynamic scale factor θ parameter may be formulated as:

$$\theta(\text{iter}) = \theta_{\max} \exp((\text{Ln}(\theta_{\min}/\theta_{\max}) / \theta_{\max})\text{iter}) \quad (8)$$

This formulation is inspired from [9]. Instead of using Eq. (4), for improved version the following equation can be used:

$$x_{new} = x_{old} + \varepsilon * \theta(\text{iter}) \quad (9)$$

where iter stands for current iteration. The proposed algorithm can be used to solve Travelling Salesman Problem with more veracity and less computational cost. $\theta_{\max}$ and $\theta_{\min}$ can taken as 1 and 0.001 respectively whereas in TSP should be taken as 15 and -15. These values are tuning the parameter in the problem. These need to be well-adjusted to make the microbats to reach the quarry/prey.

Pseudo code of the Enhanced Bat Algorithm (EBA)

Initialize Dimension D and Inertia Weight w
Create a swarm with n bats
Initialize the bat population x_i(i=1,2,…..,n)
Randomly Initialize frequency f_i, pulse rate r_i and loudness A_i
Generate the starting solution by nearest neighbor method
while (t<max no of iteration)

Calculate the fitness of each $f(x)$
for each bat, let current best $c_{best} = x$
    Generate new solution by adjusting frequency and
    updating velocities and solution
    if($rand > r_i$)
        select the location among the current best solution
        if(a better solution is found)
            update the current best
        Generate a local solution around the selected solution by a local random walk
    Generate a new solution by flying randomly
    if($rand < A_i$ & $(f(x_i) < f(g_{best}))$)
        Accept the new solution
        Increase r_i and reduce A_i
Sketch the global best solution
Stop iteration if target is reached

The proposed algorithm improves the veracity of finding the nearest best solution to solve the travelling salesman problem and reduce the computational cost. The starting solution of each bat is generated by using nearest neighbor method, by using this method to find the nearest best solution in the population. Calculate the fitness of the each bat in the population, then the bat's position is adjusted and selected depending upon the *current best* values of the bats. It compares the fitness value of each bat with current

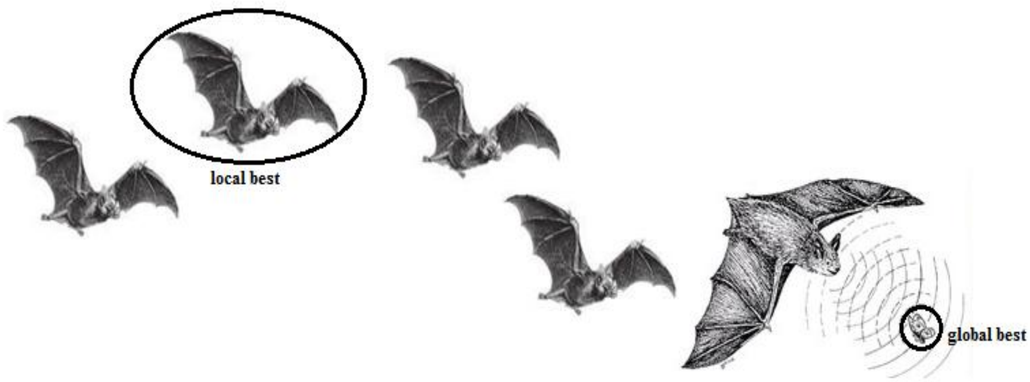

Fig. 1. Enhanced Bat algorithm

Best solution of the bats. The loudness A_i and the rate r_i of pulse emission have to be updated in each iteration. As the loudness usually decrease once a bat has found a quarry, while the rate of pulse emission increase, the loudness can be chosen as any value of convenience. The loudness and emission rate will be updated only if the new solution is improved, which means that these bats are moving towards the optimal solution.

VI. Experimental Results

To validate the performance of the proposed algorithm, different benchmark functions have been used to evaluate the accuracy rate and is compared with the other heuristic algorithm, including particle swarm optimization(PSO) [6,11], bat algorithm(BA) [1].

Ackley's function

$$f(x) = 20 + e - 20\exp\left[-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right] - \exp\left[\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right] \quad (10)$$

Griewangk's function

$$f(x) = \frac{1}{4000}\sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (11)$$

Rastrigin's function

$$f(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)] \quad (12)$$

Schwefel's function

$$f(x) = -\sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|}) \quad (13)$$

Shubbert's function

$$f(x) = [\sum_{i=1}^{d} i cos(i + (i+1)x)] \, . \, [\sum_{i=1}^{d} i cos(i + (i+1)y)] \quad (14)$$

De Jong's sphere function

$$f(x) = \sum_{i=1}^{d} x_i^2 \, , -10 \le \; x_i \; \le 10, \quad (15)$$

Table 1 shows the evaluation of different benchmark functions with the meta-heuristic algorithm in the form of mean and standard deviation.

Table 1. Comparison of EBA with PSO & BA

Benchmark Functions	PSO		BA		EBA	
	Mean	SD	Mean	SD	Mean	SD
Ackley's	22576	4216	6582	2322	363	152
Griewangk's	53960	4518	9643	4312	697	261
Rastrigin's	74670	3346	11992	3015	987	231
Schwefel's	14232	1207	8729	696	642	222
Shubert's	22816	3984	10482	3246	973	263
Sphere's	16924	1096	5173	397	287	143

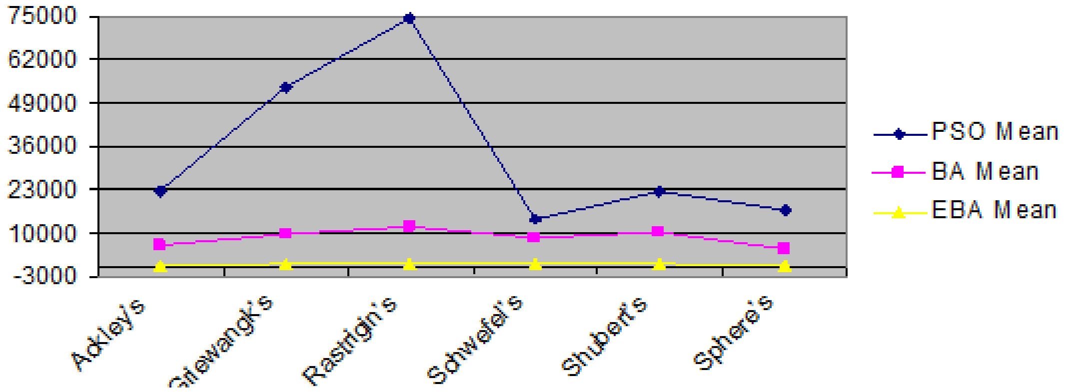

Fig. 2. Comparison of Benchmark Function using Mean

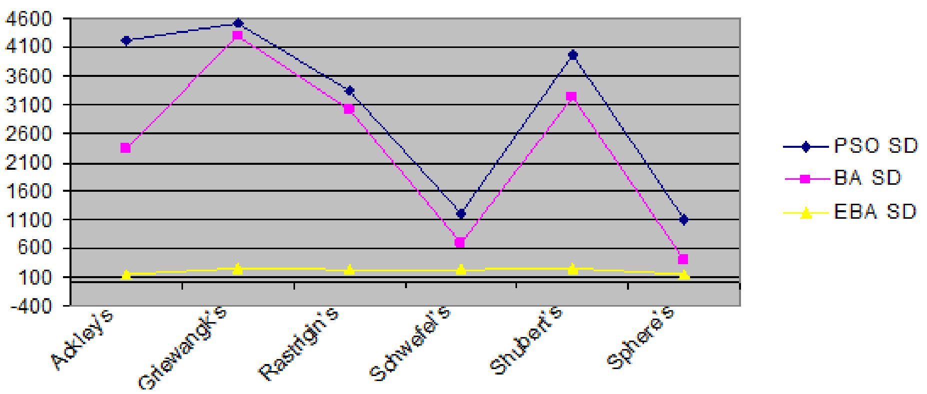

Fig. 3. Comparison of Benchmark Function using Standard Deviation

Test case	Problem	Optimal in TSPLIB	PSO	BA	EBA
Eil51	51	426	436.9	429.7	421.3
St70	70	676	697.5	686.9	676.2
Eil76	76	538	560.4	554.8	539.9
KroA100	100	21282	21310	21310	21310.0

Table 2. Travelling salesman problem comparison with different test cases

In order to examine the proposed algorithm, four TSP test cases such as Eil51, St70, Eil76 and KroA100 have been taken for experimental result. Test cases were chosen from TSPLIB[14]. Table 2 shows the optimal solution of each test cases and compared with meta-heuristic bat algorithm, PSO algorithm and enhanced bat algorithm. The comparison illustrate the efficacy of proposed algorithm. The number of cities in these test cases differs from 52 to 1432. The proposed algorithm improves the veracity of finding nearest neighbor to reach the quarry, thus should be used in dynamic scale with fixed population towards an optimal solution.

VII. Conclusion

In this paper, an enhanced bat algorithm is catalogued for solving travelling salesman problem by combining particle swarm optimization with classic bat algorithm. The new algorithm shows great proficiency in solving TSP with the dynamic problem scale. An experimental result shows that the algorithm is a providential approach for solving the TSP in terms of veracity toward an optimal solution compared with some algorithms in the literature. In accretion enhanced bat algorithm can be used to solve the travelling salesman problem with beneath computational cost.

References

[1]. Xin-She Yang, "A New Meta-heuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization", (NISCO 2010)(Eds. J.R. Gonzalez et al.), Studies in Computational Intelligence, Springer berlin,284, Springer, 65-74(2010).
[2]. Pallavi, Jayakumari. T, "Bio Inspired hybrid bat algorithm with Naïve Bayes Classifier to feature selection",in:International journal of science and research(IJSR 2013).ISSN(Online):2319-7064
[3]. Yang. J and Honavar. V, "Feature subset selection using genetic algorithm", IEEE Intelligent Systems and Their Application, vol, 13,no.2.pp.44-48,1998.
[4]. Jensen. R and Shen. Q, "Finding rough set reducts with Ant colony optimization", in proceeding of the UK Workshop on Computational Intelligence, pp.15-22,2003
[5]. Yassine Saji, Mohammed Essaid Riffi, Belaid Ahiod, "Discrete bat-inspired algorithm for travelling salesman problem", IEEE transactions,2014.
[6]. James Kennedy and Russell Eberhart. "Particle Swarm Optimization ". In Proceedings of the IEEE International Conference on Neural Network, volume IV, 1942-1948, Piscataway, NJ, 1995. IEEE Press.
[7]. Elizabeth F, Goldbarg. G, Marco C. Goldbarg and Givanaldo R. de Souza, "Particle Swarm Optimization Algorithm for the Travelling Salesman Problem". ISBN 978-953-7619-10-7, September 2008.
[8]. Xuesong Yan, Can Zhang, Wenjing Luo, Wei Li, Wei Chen and Hanmin Liu, "Solve Travelling Salesman Problem Using Partcile Swarm Optimization Algorithm": International Journal of Computer Science Issues, Nov 2012.
[9]. Mahdavi M, Fesanghary M, Damangir E. "An improved harmony search algorithm for solving optimization Problems", *Applied Mathematics and Computation*, **188**(2007)
[10]. Kaveh. A and Zakian. P, "Enhanced Bat Algorithm For Optimal Design Of Skeletal Structures", Asian Journal of Civil Engineering (BHRC) VOL. 15, NO. 2 (2014).
[11]. Kennedy, J. and Eberhart, R., "Swarm Intelligence". Academic Press, (2001).
[12]. Guanghui LIU, Heyan HUANG, Shumei WANG, Zhaoxiong CHEN, "An Improved Bat Algorithm with Doppler Effect for Stochastic Optimization". International Journal of Digital Content Technology and its Applications, 2012.
[13]. Alberto Ochoa, Lourdes Margain, Julio Arreola, Alejandro De Luna, Geovani Garia, Erika Soto, Saul Gonzalez, Kyrk Haltaufoerhyde & Vinicius Scarandangotti, "Improved solution based on Bat Algorithm to Vehicle Routing Problem", in a Caravan Range Community, HIS 2013.
[14]. http://www.iwr.uniheidelberg.de/groups/comopt/software/TSBLIB95