International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 5, Issue 4, July - August 2016 ISSN 2278-6856

A comparative study of continuous multimodal global optimization using PSO, FA and Bat Algorithms

Radha A. Pimpale[1], P.K. Butey[2]

[1]PhD Scholar, Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur, Maharashtra India

[2]Associate Professor and Head, Kamla Nehru Mahavidhalaya,
Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur, Maharashtra India

Abstract

This paper introduces the nature-inspired metaheuristic algorithms for optimization of standard Benchmark function, including Firefly algorithm, PSO algorithms and Bat algorithm. We have implemented these algorithms in MATLAB. We have considered here how this algorithms work on continuous multimodal global optimization benchmark function. All these are the evolutionary nature inspired optimization metaheuristic algorithms and are inspired by the nature. to find out optimal solutions of continuous, multimodel Benchmark function. The stimulation results of this experiment were analyzed and compared to the best solutions found and compare with time. So the Bat algorithm in each continuous, multimodel Benchmark function optimization function seems to perform better and efficient.

Keywords:-Firefly algorithm, Metaheuristic algorithm, PSO, fireflies, bat.

1.Introduction

The goal of global optimization is to find the best possible possible solution from the given constraints. The objective function is minimize or maximize and the constraints are the relationships [10, 11, 13]. Optimization algorithms can be divided in two basic classes: deterministic and stochastic (or probabilistic). Deterministic algorithms are almost all local search algorithms, and they are quite efficient and aim in finding the local optima. Stochastic (or probabilistic) algorithms are optimization methods that generate and use random variables and have ability to local as well as global solution. There are three Metaheuristic algorithms like PSO, Firefly algorithm, Bat algorithms having aim to find the optimal solution.

Optimization algorithms have exploration and exploitation problem. In the optimization term, exploration means visiting new areas of the search space which have not been investigated before. Exploration is a procedure by which we try to find novel and better solution states. Parameters such as convergence, elapse time, stability of algorithms are considered

Particle swarm optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling which aims at finding the food [1]. The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its *pbest* and *lbest* locations. Each swarm moves in the search space in a cooperative search procedure. Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations. The group of possible solutions is a set of particles, called swarms.

PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored). This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called *gbest*. When a particle takes part of the population as its topological neighbours, the best value is a local best and is called *lbest*.

After finding the two best values, the particle updates its velocity and positions with the PSO. The PSO has algorithmic parameters:(a) Vmax or maximum velocity which restricts Vi(t) within the interval [−Vmax, Vmax] (b) An inertial weight factor ω (c) Two uniformly distributed random numbers φ1 and φ2 respectively that determine the influence of p(t) and g(t) on the velocity update formula. (d) Two constant multiplier terms C1 and C2 known as “self-confidence” and “swarm confidence”, respectively following equation 1(a) and 1(b).

$$v_{t+1} = v_t + c_1 r_1 (g - x_t) - c_2 r_2 (p - x_t) \quad 1(a)$$

$$x_{t+1} = x_t + v_{t+1} \qquad 1(b)$$

PSO Algorithm

For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (pbest) in history set current value as the new pbest
End
Choose the particle with the best fitness value of all the particles as the gbest
For each particle
Calculate particle velocity according equation (a)
Update particle position according equation (b)
End

While maximum iterations or minimum error criteria is not attained Particles' velocities on each dimension are clamped to a maximum velocity Vmax. If the sum of accelerations would cause the velocity on that dimension to exceed Vmax, which is a parameter specified by the user. Then the velocity on that dimension is limited to Vmax.

Firfly Algorithm

Firefly algorithm is a population based algorithms, The Firefly algorithm was developed by Xin-She [Yang XS (2008) Nature-Inspired Metaheuristic Algorithms and it is based on idealized behaviour of the flashing characteristics of fireflies. For simplicity, we can summarize these flashing characteristics as the following three rules [2]:

a) All fireflies are unisex, so that one firefly is attracted to other fireflies regardless of their sex.

b) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less bright one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If no one is brighter than a particular firefly, it will move randomly.

c) The brightness of a firefly is affected or determined by the landscape of the objective function to be optimized

For simplicity we can assume that the attractiveness of a firefly is determined by its brightness or light intensity which in turn is associated with the encoded objective function. In the simplest case for an optimization problem, the brightness *I* of a firefly at a particular position X can be chosen as $I(x) \propto 1f(x)$. However the attractiveness is relative, it should vary with the distance *rij* between firefly i and firefly *j*. As light intensity decreases with the distance from its source and light is also absorbed in the media, so we should allow the attractiveness to vary with degree of absorption [3].

Step 1: Attractiveness and light intensity

In the firefly algorithm, there are two important issues: the variation of the light intensity and the formulation of the attractiveness. We know, the light intensity I(*r*) varies with distance r monotonically and exponentially, that is:

$$I(r) = I_0 e^{-\gamma r^2} \qquad (2)$$

Where I_0 the original light intensity and γ is the light absorption coefficient. As firefly attractiveness is proportional to the light intensity seen by adjacent fireflies, we can now define the attractiveness β of a firefly by Equation (2)

$$\beta(r) = \beta_0 e^{-\gamma r^2} \qquad (3)$$

Where *r* is the distance between each two fireflies and β_0 is their attractiveness at r = 0 i.e., when two fireflies are found at the same point of search space. The value of γ determines the variation of attractiveness with increasing distance from communicated firefly.

Step 2: Distance

The distance between any two fireflies *i* and *j* at x*i* and x*j* respectively, the Cartesian distance is determined by equation (3) where x*i,k* is the *kth* component of the spatial coordinate of x*i* the *ith* firefly and **n** is the number of dimensions.

$$r_{ij} = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \qquad (4)$$

Step 3: Movement

The firefly *i* movement is attracted to another more attractive (brighter) firefly *j* is determined by:

$$x_i(t+1) = x_i + \beta_0 e^{-\gamma r^2} (x_j - x_i) + \alpha (rand - 0.5) \qquad (5)$$

Where the second term is due to the attraction while the third term is randomization with *α* being the randomization parameter and *r and* is a random number generator uniformly distributed in [0, 1].

Why firefly algorithms is special [4]

1.The first response that can be made regarding this question is the moment strategy of the fireflies. Eqn (1) is the moment equation and is consist of 2 main parts. We can mention them as information based movement and random moment. These two parts are responsible for providing proper exploitation and the exploration over the search space for finding optimal solutions.

2.The given firefly's brightness and its distance with another brighter firefly are taken into the account when modifying its solution. Random part will create some random solution which can lead the solution to a good or bad situation.

Fireflies Algorithms [5]

```
initialize n fireflies to random positions
loop max Epochs times
  for i := 0 to n-1
    for j := 0 to n-1
      if intensity(i) < intensity(j)
        compute attractiveness
        move firefly(i) toward firefly(j)
        update firefly(i) intensity
      end for
    end for
  sort fireflies
end loop
return best position found
```

Bat Algorithms

The bat algorithm is a metaheuristic algorithm for global optimization. It was inspired by the echolocation behavior of microbats, with varying pulse rates of emission and loudness. [6] [7]

The following approximate or idealized rules were used.

1. All bats use echolocation to sense distance and they also 'know' the difference between food/prey and background barriers in some magical way;

2. Bats fly randomly with velocity vi at position xi with a frequency fmin , varying wavelength λ and loudness A0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission r $\in$ [0,1], depending on the proximity of their target;

3. Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A0 to a minimum constant value Amin how their positions xi and velocities vi in a d-dimensional search space are updated. The new solutions xtiand velocities vti at time step t are given by

$$f_i = f_{min} + (f_{max}+f_{min})\beta, \quad (6)$$
$$v_i^{t+1} = v^t_i + (x_i^t - x_0)f_i, \quad (7)$$
$$x_i^{t+1} = x_i^t + v_i^t \quad (8)$$

Initially, each bat is randomly assigned a frequency which is drawn uniformly from [fmin, fmax]. For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$X_{new} = x_{old} + \varrho A_t, \quad (9)$$

Where ϱ is a random number vector drawn from [−1,1], while At = <At> is the average loudness of all the bats at this time step. T

Loudness and Pulse Emission

The loudness Ai and the rate ri of pulse emission have to be updated accordingly as the iterations proceed. As the loudness usually decreases once a bat h as found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. For example, we can use A0= 100 and Amin= 1. For simplicity, we can also use A0 = 1 and Amin = 0, assuming Amin = 0 means that a bat has just found the prey and temporarily stop emitting any sound. Now we have

$$A^{t+1}_i = \alpha A^t_i, \quad r^t_i = r^0_i[1-\exp(-\gamma t)] \quad (10)$$

where α and γ are constants. In fact, α is similar to the cooling factor of a cooling schedule in the simulated annealing (Kirkpatrick *et al*., 1983). For any $0 < \alpha < 1$ and
$\gamma > 0$,
$A^t_i \to 0$, $r^t_i \to r^0_i$, as $t \to \infty$

Bat Algorithms [12]

```
Objective function
                    f(x), x = (x1, ..., xd)T
Initialize the bat population
                    xi (i= 1,2, ..., n) and vi
Define pulse frequency fi at xi
Initialize pulse rates ri and the loudness Ai
while (t < Max number of iterations)
     Generate new solutions by adjusting frequency,  and
      updating velocities and locations/solutions [equations
       (9) and (10)]
if (rand> ri)
    Select a solution among the best solutions
   Generate a local solution around the selected best
solution
end if
Generate a new solution by flying randomly
if (rand < Ai & f(xi)< f(xi+1))
     Accept the new solutions
     Increase ri and reduce Ai
end if
    Rank the bats and find the current best xi
end while
```

RESULTS AND DISCUSSION

Consider unconstrained global optimization benchmark function in order to demonstrate how the firefly, PSO and Bat algorithm works, we have implemented it in matlab and the results are shown below. In order to show that both the global optima and local optima can be found simultaneously, Simulations were carried out by assigning equal size of population to be 100,300 and 500 for the all the algorithms. All the mentioned algorithms are run with 1000 iterations,, then the results are recorded. The stopping criteria is set to maximum number of iterations. For fair analysis and comparison all the algorithms are well tested on standard unconstrained global optimization benchmark function.

In particle swarm optimization parameters are considered as Maximum Number of Iterations are 1000,population size of particle are considered as 100 and 300 and 500, Inertia Weight w=1, Inertia Weight Damping Ratio wdamp=0.99 and Personal Learning Coefficient c1=1.5,and Global Learning Coefficient=2.0.The dimensions of the problems are set to be 30, for every population the results are recorded and presented in two different ways viz. a) best solution i.e global minimum b) elapsed time

In Firefly Algorithms optimization parameter Maximum Number of Iterations are 1000, population size of fireflies are considered as 100 and 300 and 500, alpha=0.5, beta=0.1 and gamma =1 , The dimensions of the problems are set to be 30, for every population the results are recorded and presented in two different ways viz. a) best solution i.e global minimum b) elapsed time

In Bat Algorithms parameter considered as Maximum Number of Iterations are 1000, population size of fireflies are considered as 100 and 300 and 500, Loudness =0.5 (constant or decreasing), Pulse rate (constant or decreasing)=0.5 , Lb=-2*ones(1,d),Ub=2*ones(1,d); The dimensions of the problems are set to be 30, for every population the results are recorded and presented in two different ways viz. a) best solution i.e global minimum b) elapsed time

Table 1 Comparative performance of Algorithms (Population=100)

Benchmark function	PSO	FA	Bat
Sphere Function	4.3267 (34.1273)	1.1878 (63.5433)	7.136 (24.94)
Ackley function	8.881 (22.8763)	8.8818 (18.1611)	0.00216 (37.69)
Rastrigin function	0 (31.4657)	0 (36.9632)	57.7098 (42.49)
Griewank function	0.007396 (30.4459)	0 (0.03925)	5.942 (32.11)
Rosenbrock Function	0.2898 (47.1184)	1.5731 (62.3811)	29.40 (46.25)
Schwefel Function	798.5128 (32.1073)	801.3398 (31.4254)	3.2196 (48.50)

Table 2 Comparative performance of Algorithms (Population=300)

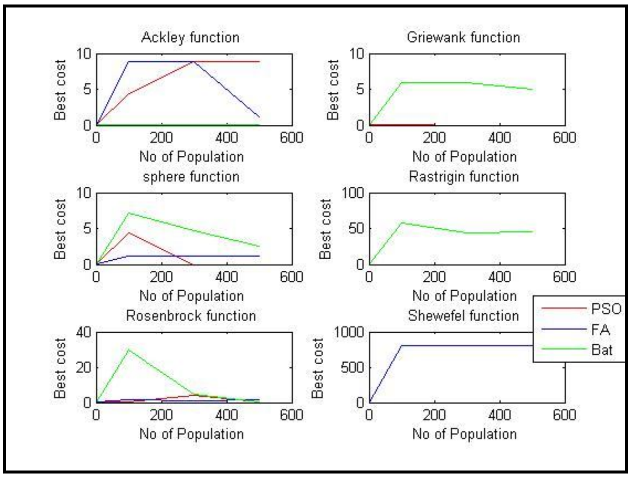
Benchmark function	PSO	FA	Bat
sphere Function	0 (110.3773)	1.1671 (556.952)	4.7428 (30.10)
Ackley function	8.8818e-16	8.8818(451.717)	0.002579 (38.95)
Rastrigin function	0 (99.24.2)	0 (386.0997)	43.7803 (43.35)
Griewank function	0 (95.975335)	0 (0.3090)	5.942 (32.11)
Rosenbrock Function	4.0364 (122.3400)	1.2441 (560.4124)	4.4691 (46.74)
Schwefel Function	798.5128 (79.837988)	808.3921 (279.2434)	5.7732 (49.64)

Table 3 Comparative performance of Algorithms (Population=500)

Benchmark function	PSO	FA	Bat
Sphere Function	0(136.2281)	1.0810 (1560.2457)	2.5845 (30.67)
Ackley function	8.8818(175.3049)	8.8818 (451.7176)	0.002277 (41.56)
Rastrigin function	0(151.844925)	0 (1319.7696)	45.7697 (44.04)
Griewank function	0.007396(30.445990)	0 (0.8410)	5.0217 (34.77)
Rosenbrock Function	0.0729(222.7411)	1.4270 (1533.43)	0.6223 (47.38)
Schwefel Function	798.5128(167.4429)	808.3333 (1557.870)	2.8866 (49.66)

Performance of Multimodel function with given algorithm

Ackley's function has one global optimum and many minor local optima. It is better for the Bat algorithm among the six standard benchmark function as its local optima are shallow. It has been observed that Ackley function has best performance than other function. Griewank's function is that it is good for higher population for best cost . Rastrigin's function is a complex multimodal problem with a large number of local optima. When attempting to solve Rastrigin's function, algorithms may easily fall into a local optimum. Sphere function is differentiated in every algorithum. Firefly algorithms reaches towards optima in Sphere Benchmark function. Schwefel's function is due to its deep local optima being far from the global optimum



Comparison Graph for Time required to run Different Algorithms

Comparing the coverage and the execution speed , among these six multimodal, n dimensional global optimization benchmark function with PSO algorithms, fireflies and Bat Algorithms it has been observed that Bat algorithms is better for convergence and execution speed.

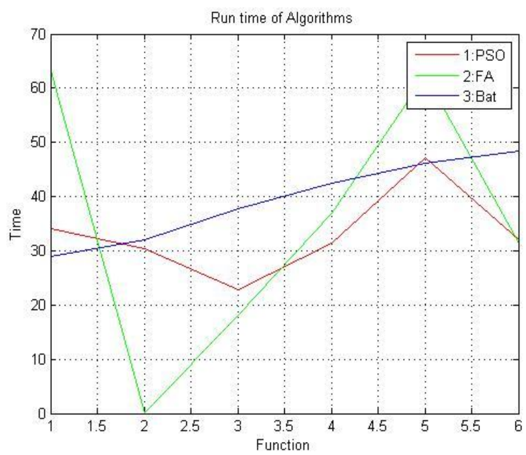


Fig 2 Comparison of Time required to run Different Algorithms

References

[1]. Kennedy, J. and Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks , Vol. 4. 1995, 1942–1948

[2]. Yang XS (2008) Nature-Inspired Metaheuristic Algorithms. Luniver Press.

[3]. Yang, X. S., (2011). Bat algorithm for multi-objective optimisation,Int. J. Bio-Inspired Computation, Vol. 3, No. 5, pp. 267–274

[4]. MKA Ariyaratne, WPJ Pemarathne, ”A Review of Recent Advancements of Firefly Algorithm; A Modern Nature Inspired Algorithm” , Proceedings of 8 th International Research Conference, KDU, Published November 2015

[5]. ztok Fister Jr.a, Matjazˇ Percb,c,d, Salahuddin M. Kamalc, Iztok Fister ,“A review of chaos-based firefly algorithms: Perspectives and research challenges”, . Fister Jr. et al./Applied Mathematics and Computation 252 (2015) 155–165

[6]. J. D. Altringham, Bats: Biology and Behaviour, Oxford University Press, (1996).

[7]. P. Richardson, Bats. Natural History Museum, London, (2008), Yang, X. S. (2010). "A New Metaheuristic Bat-Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)". Studies in Computational Intelligence **284**: 65–74. arXiv:1004.417

[8]. .X. S., Yang, A New Metaheuristic Bat -Inspired Algorithm, in: Nature Inspired Cooperative Strategies for Optimization (NISCO 2010) (Eds. Cruz, C.; Gonzalez, J. R.; Pelta, D. A.; Terrazas, G), Studies in Computational Intelligence Vol. 284, Springer Berlin, pp. 65 – 74, 2010.

[9]. [9] P. Agarwal and S. Mehta, “Nature-Inspired Algorithms: State-of-Art, Problems and Prospects”, International Joournal of Computer Applications, 2014, Vol. 100, No.14, pp. 14-21.

[10]. I. Fister, X. S. Yang, J. Brest, an d D. Fister, “A Brief Review of Nature-Inspired Algorithms for Optimization”, ElektrotehnißSki Vestnik, Jul, 2013.

[11]. WengKee Wong, Nature-Inspired Metaheuristic Algorithms for Generating Optimal Experimental Designs, 2011

[12]. Yazan A. Alsariera, Hammoudeh S. Alamri, Abdullah M. Nasser, Mazlina A. Majid, Kamal Z. Zamli, ”Comparative Performance Analysis of Bat Algorithm and Bacterial Foraging Optimization Algorithm using Standard Benchmark Functions”, 2014 8th Malaysian Software Engineering Conference (MySEC).

[13]. S K Subhani Shareef E. Rasul Mohideen Layak Ali ,”Directed Firefly algorithm for multimodal problems”, 2015 IEEE International Conference on Computational Intelligence and Computing Research

AUTHORS

Prof. Radha A. Pimpale is presently working as Assistant Professor at Priyadarshini Institute of Engineering and Technology Nagpur. She has 11 research papers published in reputed national and international journals. She is research scholar at Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur. She perusing Ph. D in under the guidance of Dr P.K. Butey. She has total experience of 15 years in teaching and research in the field of Computer Science

Dr. Pradeep Butey is resently working as associate professor and Head of the Department of Computer Science, Kamla Nehru Mahavidyalaya, Nagpur, He has done his doctorate in Computer Science from RTM Nagpur University, Nagpur, in Mar 2011. His areas of interest include data mining, fuzzy logic, optimization, RDBMS. He has 18 Research papers published in reputed International Journals. He is Chairman Board of Studies in Computer Science of the University, He has vast experience of 12 years in the field of teaching and research in Computer Science. He has worked as a member of selection committee at many instances. He is a Member of Academic Council RTM Nagpur University.