International Journal of Advanced and Innovative Research (2278-7844) / Volume 7 Issue 6

Bat Algorithm, Particle Swarm Optimization and Grasshopper Algorithm: A Conceptual Comparison

Deepak Nandal [1], Dr Om Prakash Sangwan [2]

[1] Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar-125001, Haryana, India
E-mail: gju.dpknandal@gmail.com
ORCID id- 0000-0002-9248-8757
[2] Department of Computer Science and Engineering, Guru Jambheshwar University of Science and Technology, Hisar-125001, Haryana, India
Corresponding author. Tel.: +91-8685000345;
*E-mail: sangwan0863@gmail.com

Abstract— Computer science, mathematics and operational research comprises of optimization problems. Use of the optimization techniques to solve the engineering problems and by making the most it from the available resources to find the optimum solution to the specific problem is the need of today. By performing exploration and exploitation in the given domain space various algorithms have been designed, the motive of this research paper is to find the best available approach of optimization from the recent researches. In this context, three types of metaheuristic approaches bat algorithm, particle swarm optimization and grasshopper optimization are selected and tested to find the optimum solution efficiently. Bat algorithm is inspired from echolocation behaviour of bats, particle swarm optimization is also nature inspired algorithm based on the swarm made by the birds in search of food in the space, while the grasshopper optimization is based on the swarm made by the grasshopper in the adult age and the behaviour is combined with the larval stage of the grasshoppers. Experimental results were analysed and it is observed that GHA performs better that PSO and BA.

Keywords- Cuckoo Optimization Algorithm (COA) Evolutionary algorithms, Nonlinear optimization, BAT, Grasshopper Algorithm.

I. INTRODUCTION

Optimization is one of the most critical part of science, which revolves around a lot of problems we face in daily life, in many of the cases there is limitation of the knowledge about the boundaries of the problems, parameters of the problem and also there are very much limitations are of the time as well[1]. And the way of doing with the least of the knowledge and resources, achieve the near to the best solution is certified as optimization. Optimization is the process of finding the most effective and very much favorable value and condition. By minimizing and maximizing the function set values of a problem and choosing the right input values from the given domain of values one can find the get the optimum solution the problem. Optimization problem can be classified into sub such as Nature inspired optimization techniques, Linear programming, Cluster based optimization techniques, Metaheuristic approaches, Approximation algorithms etc. The final aim of all such techniques and methods is to find the optimal solution to the problem, with in a specific time frame[1], [2].

Metaheuristic approaches are very much easy to implement in comparison to the other methodologies named earlier and they produce comparatively better solution to the problems, the most important point to be noticed here is that the metaheuristic approaches do not require any gradient information about the problem, as they are problem independent techniques, and once they are understood well, can be applied to various scientific optimization problems we face today[3]. Metaheuristic algorithms are general purpose algorithms and they can be applied to any optimization problem, as the metaheuristic approaches uses combinatorial approach and the solution is find out by exhaustive methods to achieve the optimum solution. They can also be viewed as top level general purpose methodologies which are used in guiding strategies in finding the underlying heuristics[4]. There are various metaheuristic approaches are being proposed by various researchers in the relevant field, so the main question arises what technique to follow.

In this paper three different metaheuristic approaches: Bat Algorithm, Particle Swarm Optimization and Grasshopper Algorithm are used for finding solutions and results are compared. Bat algorithm comprises of the ecological behavior of the bats and it is nature inspired metaheuristic approach[5], Particle Swarm is based on the group of birds flying in search of food in a space, these birds with the self-memory behavior and their social behavior can find the location of the food in that space[6]. Where-as in the grasshopper the two stages are the key to this technique, that is one is the early larval stage in which the movement is very slow and the coverage is also in a very small area, where-as in the adult hood the movement is fast and the coverage becomes very much wider[7]. This paper aims to compare the grasshopper with bat algorithm and Particle swarm optimization algorithm on the factors such as like minimum run rate and the success rate. Firstly, the bat, particle swarm optimization and the grasshopper are briefly explained and in the second part the experimentation part is explained and discussed in the second part. The experimental factors and the results are then show listed in the third part and finally the forth part concludes the paper.

II. BA, PSO and GHA ALGORITHMS

A. Bat Algorithm

Bats are stunning as well as splendid animals to work with. The echolocation capability of the bats is the key feature which creates the interest of the researchers in them, bats are the only mammals on earth which have wings as well as show the echolocation behavior as well. They are of variable sizes, some of the mammals in bats species are very small such as a small insect (weighs in grams) to the some of them are massive as of with the wing span around 2 meters and around 2 kg in weight as well. Certainly, some species in bats have a very good eyesight and most of them also have a very sensitive smell sense[5]. Nevertheless, we are here interested in the echolocation behavior of the bats the most and how it can be associated with the objective function that is to be optimized[2].

Bat algorithm is a metaheuristic algorithm which as usual will work on the approximation and the following are the appropriate direction of work is as follows:

- Bats uses echolocation behavior to sense the distance around them as well as the obstacles around them and also they can differentiate between their food in which they are interested and the things in which they are not interested;
- Let the velocity of a flying bats is ai on some position xi and the frequency is fixed $fmin$, loudness is Lo and the variable wavelength is $\propto$ for the search of food. The wavelength or the frequency can be automatically adjusted and the rate of emission $r \in [0,1]$ which totally depends on the proximity of their aim;
- Sometimes the loudness Lo may vary, so it is assumed that the loudness ranges from Lo to a minimum constant value $Amin$;[5]

In addition to the above assumptions and the approximation rules, some more approximations are also used for simplification. We can use any range of wavelength and frequency for a specific problem such as $[fmin, fmax]$ i.e. [10kHz, 1000kHz] with respect to the wavelength $[\propto min, \propto max]$ i.e. [0.1mm, 20mm]. It is well known that higher is the frequency shorter is the wavelength and most probably it will travel shorter distance as in case of bats, so it is also assumed that the $f \in [0, fmax]$ and the rate of pulse is in the range of [0,1], where 1 means max rate of pulse emission and 0 means no emission. Using the above assumptions, the bat algorithm is:

Algorithm 1 Bat algorithm

Objective function $f(X) = X(x1, \ldots\ldots, xz)^T$

Initialize the bat population $Xi\ (i = 1, 2 \ldots\ldots n)\ and\ ai$

Define pulse frequency fi and Xi

Initialize pulse rates ri and the loudness Li

while (t < Max number of iterations)

Generate new solutions by adjusting frequency,

And updating velocities and locations/solutions

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 $\left(rand < Li \,\&\, f(xi) < f(x*)\right)$

Accept the new solutions

Increase ri and reduce Li

end if

Rank the bats and find the current best $x*$

end while

Post-process results and visualization

B. *Particle Swarm Optimization*

Particle swarm optimization is a nature inspired metaheuristic approach which consists of the behavior of a group of flying birds in search of food in a space, but all the birds at first do not know about where the food is in the space, likewise in the case of bats, here also with the self-memory behavior and the social behavior can find out the location of the food for them[8]–[11]. One can also view this system as a simulated biological behavior of the group and the genetic algorithm, but the particle swarm optimization algorithm is more fast, simple, less genetic and a quality algorithm for natural selection and solving other likewise complex problems, to find out the optimum solution efficiently and quickly [8].

Every point in the group or bird or the i^{th} particle is represented as $Yi = (yi1, yi2, \dots \dots, yid)$. The previous best position (the best fitness value previous) of the i^{th} particle is registered and represented as $Zi = (zi1, zi2, \dots \dots, zid)$. The index of the particle in the group is represented by g. The rate of velocity of the group is represented as $Vi = (vi1, vi2, \dots \dots, vid)$. The particles are controlled by the following equation:

$$vid = vid + c1 * rand() * (zid - yid) + c2 * Rand() * (zgd - yid) \qquad (B_1)$$

$$yid = yid + vid \qquad (B_2)$$

Where c1 and c2 are the positive constants and rand () and Rand () are two random functions in the range [0,1].

In particle swarm optimization the group remains intact throughout the experimentation, the selection of individual particle is not performed as in case of genetic algorithms, only the velocity is recorded during the course of run and the comparison is performed of its own previous best position and the previous best position of its companions[8]. In particle swarm optimization the strategy parameter called the inertia weight is used to balance between the local and the global search, so the equation B_1 and B_2 changes to:

$$vid = W * vid + c1 * rand() * (zid - yid) + c2 * Rand() * (zgd - yid \qquad (B_{11})$$

$$yid = yid + vid \qquad (B_{12})$$

Where w is the inertia weight, where a large inertia weight describes a global search and a small inertia describes a local search.

C. Grasshopper Algorithm

Grasshoppers are insects and called as pest, because they cause heavy damage to the crops, they are also known as plant eaters. Grasshopper exists on the ground basically having very strong legs, which helps them to escape from sudden threats and predators. Some grasshoppers can also change colors and also form swarms. The size of the swarm or the group can be gigantic and can ruin the farm within a short span of time. They can form swarm while in the air so that they can migrate over long distances. The key features are as follows: the larval phase is very slow and steady, in contrast to the long and rapid movements in the adulthood, food seeking ability is also a key feature while selection of the grasshopper optimization technique[7].

The mathematical equation to show the simulation of the swarm creation and behavior of the grasshopper is shown as:

$$Xi = Yi + Gi + Wi \qquad (C_1)$$

Where the Xi represents the i^{th} grasshopper position,

Gi represents the gravity force,

And the wind advection is represented as Wi.

Now the random behavior of the equation can be represented as

$$Xi = r1Yi + r2Gi + r3Wi \qquad (C_2)$$

Where r1, r2 and r3 are the random numbers in the range [0,1].

$$Yi = \sum_{\substack{j=1 \\ j\neq i}}^{N} s(dij)\widehat{dij} \qquad (C_3)$$

Where dij is the distance between the ith and jth grasshopper and the $dij = |x_j - x_i|$,

The social forces are represented as

$$s(r) = f\, e^{-r \div l} - e^{-r} \qquad (C_4)$$

Where f indicated the attraction and l is the attractive length scale for the grasshopper.

The G component in the equation (C_1) is calculated as:

$$Gi = -g\widehat{e1} \qquad (C_5)$$

Where g is the gravitational constant and $\widehat{e1}$ shows a unity vector towards the center of the earth.

The W component of the equation (C_1) is:

$$Wi = u\widehat{e2} \qquad (C_6)$$

Where u is the drift component and $\widehat{e2}$ is a unity vector in the direction of the wind.

Now substituting the values if Y, G and W in equation (C_1),

$$Yi = \sum_{\substack{j=1 \\ j\neq i}}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{dij} - g\,\widehat{e1} + u\widehat{e2} \qquad (C_7)$$

Where $s(r) = fe^{-r \div l} - e^{-r}$ and N is the number of grasshoppers[7].

III. EXPERIMENT

Different functions are used in this paper for evaluating the success of the Bat algorithm, Particle swarm optimization and Grasshopper in the Table 1.

By comparing the numbers of the function evaluations for a given accuracy, we acquired the desired results. In the simulations the tolerance is fixed to $\leq 10^{-5}$ and for 10 Monte-Carlo runs each algorithm is run, so that effective analysis can be achieved. It is observed that in most of the cases the sufficient population is 20 to 60 and we have the size of the population is n = 10 to 250, so the population is fixed to n =40 for all the conditions.

Various benchmark functions used for testing[12]:

$$fo\ (\vec{x}) = \sum_{i=1}^{n} x_i^2 \qquad \text{(Sphere)}$$

$$F(\vec{x}) = -20.\exp\left(^{-0.2}\sqrt{\frac{i}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi.x_i)\right) \qquad \text{(Ackley)}$$

$$f_1(\vec{x}) = \sum_{i-1}^{n-1}(100(x_{i+1} - x_j^2)^2 + (x_i - 1)^2 \qquad \text{(Rosen} - \text{brock)}$$

$$f_2(\vec{x}) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10) \qquad \text{(Rastrigin)}$$

$$f_3\ (\vec{x}) = \frac{1}{4000}\sum_{i=1}^{n}(x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \qquad \text{(Griewank)}$$

A. *Experimental result analysis*

1) *Minimum run time analysis:* All the three algorithms BA, PSO and GHA are compared on the factor of run time. As shown in the figure 1 the GHA showed the minimum rum time amongst the three algorithms.

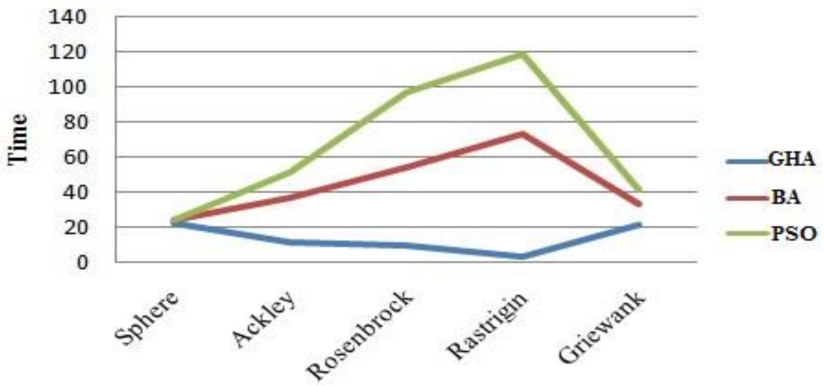

Figure 1: Comparative analysis of minimum run time among algorithms

2) *Success Rate:* The figure 2 shows the success rate for BA, PSO and GHA. The success rate of the GHA is found better while BA showed comparatively better results than PSO.

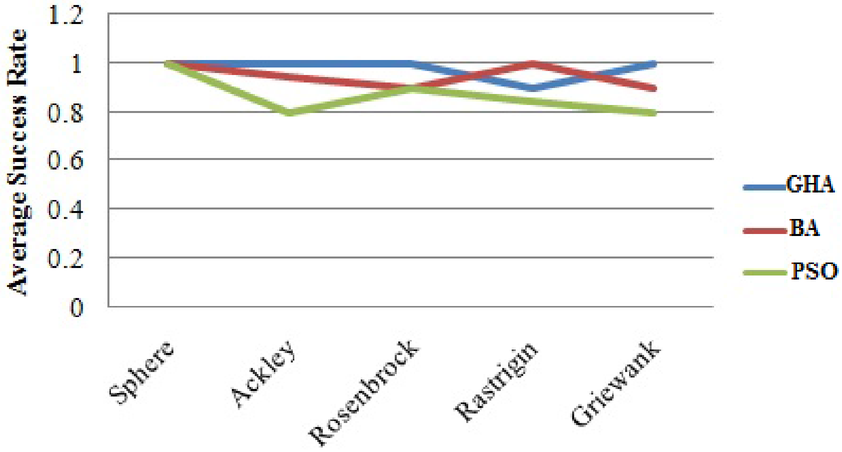

Figure 2: Comparative analysis of success rate among algorithms

IV. CONCLUSION:

From the experiments it is clear that the grasshopper outperforms the rest of the two metaheuristic approaches that are bat algorithm and particle swarm optimization in all the three aspects covered. The grasshopper also proves to be better in terms of speed of convergence, so GHA can be used as a powerful technique to solve the different optimization techniques we face in different areas of science today to achieve the best and optimum solution in a specific time period, as GHA takes less time for generating optimum or near optimum solution as compared to the BA and PSO. As in BA there is no provision to memorize and store the information about the previous better solution and because of this it ends up in missing the solutions.

In future, some more concepts and modifications can be added to it to fine tune the parameters and the effect of this can be measured.

REFERENCES

[1] A. Govan, “Introduction to optimization,” in *North Carolina State University, SAMSI NDHS, Undergraduate workshop*, 2006.

[2] R. T. Marler and J. S. Arora, “Survey of multi-objective optimization methods for engineering,” *Struct. Multidiscip. Optim.*, vol. 26, no. 6, pp. 369–395, 2004.

[3] N. Gupta and K. Sharma, “Optimizing intermediate COCOMO model using BAT algorithm,” in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2015, pp. 1649–1653.

[4] C. L. Simons and J. E. Smith, “A comparison of meta-heuristic search for interactive software design,” *Soft Comput.*, vol. 17, no. 11, pp. 2147–2162, 2013.

[5] X.-S. Yang, “A New Metaheuristic Bat-Inspired Algorithm,” in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, and N. Krasnogor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 65–74.

[6] P. V. G. D. Prasad Reddy and C. V. M. K. Hari, “Fuzzy Based PSO for Software Effort Estimation,” in *Information Technology and Mobile Communication: International Conference, AIM 2011, Nagpur, Maharashtra, India, April 21-22, 2011. Proceedings*, V. V. Das, G. Thomas, and F. Lumban Gaol, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 227–232.

[7] S. Saremi, S. Mirjalili, and A. Lewis, “Grasshopper Optimisation Algorithm: Theory and application,” *Adv. Eng. Softw.*, vol. 105, pp. 30–47, 2017.

[8] Eberhart and Y. Shi, “Particle swarm optimization: developments, applications and resources,” in *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, 2001, vol. 1, pp. 81–86 vol. 1.

[9] Y. Shi and R. C. Eberhart, “Empirical study of particle swarm optimization,” in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, 1999, vol. 3, p. 1950 Vol. 3.

[10] J. C. Lin and H. Y. Tzeng, “Applying Particle Swarm Optimization to estimate software effort by multiple factors software project clustering,” in *2010 International Computer Symposium (ICS2010)*, 2010, pp. 1039–1044.

[11] J. Khorasani, "A New Heuristic Approach for Unit Commitment Problem Using Particle Swarm Optimization," *Arab. J. Sci. Eng.*, vol. 37, no. 4, pp. 1033–1042, 2012.

[12] S. Arora and S. Singh, "A conceptual comparison of firefly algorithm, bat algorithm and cuckoo search," in *2013 International Conference on Control, Computing, Communication and Materials (ICCCCM)*, 2013, pp. 1–4.