Factors Influencing Performance of Firefly and Particle Swarm Optimization Algorithms

Damanjeet Kaur, UIET, Panjab University, Chandigarh

***Abstract*— In this paper, two nature inspired meta heuristic approaches particle swarm optimization and firefly algorithm are discussed. Both the approaches are population based approaches and has wide applications in various problems. Various factors influencing its performance is compared on the basis of selection of size of population, number of iterations, quality of solution, convergence criterion and their simplicity of applicability on test functions. The performance of the two approaches is tested on different test functions.**

Index Terms— **Firefly algorithm, Particle Swarm Optimization, Performance Parameters**

I. Introduction

These days, nature-inspired metaheuristic algorithms are very popular because of their simplicity and ease of application. These algorithms showed promising results in various optimization problems and turned successful in NP hard problems . These evolutionary techniques are inspired from the behavior of swarm such as fish and bird. Particle swarm optimization and Firefly algorithms are common these days. Two decades ago, PSO was introduced by Kennedy and Eberhart [1-2] as an alternative to genetic Algorithm. PSO unlike GA, has no crossover between individuals, has no mutation and particles are never substituted by other individuals during the run. Instead the PSO refines its search by attracting the particles to positions with good solutions. The PSO remembers the best position found by any particle. Additionally each particle remembers its own previously best found position. PSO is implemented for various NP hard problems [3-14]. But later on it was realized that PSO sometimes go into local minima and does not provide the optimal solution of the problem. Then PSO is combined with other meta heuristic or local search methods in order to avoid trapping in local minima and get global minima of the problem. Firefly algorithm is another meta-heuristic novel population based approach developed by Xin-She Yang in 2008[15-17] . It is effectively applied for continues NP-hard problems [18-25]. It mimics the social behavior of fireflies. The flashing light of fireflies is a fantastic sight in the sky and fireflies normally attract mating partners and potential prey by using such flashes. Both genders join together by the rhythmic flash, the rate of flashing and the amount of time of flashing. Females respond to a male's unique and peerless pattern of flashing. It is possible to formulate optimization algorithms because the flashing light can be formulated in such a way that it is associated with the objective function to be optimized. Firefly algorithm is very efficient in finding the global optima with high success rates.

II. Particle Swarm Optimization and Firefly Algorithm

A Particle Swarm Optimization

Particle Swarm Optimization is like an evolutionary technique and it (PSO) was introduced by Kennedy and Eberhart in 1995 [1-2] as an alternative to Genetic Algorithms. The PSO technique has ever since turned out to be a competitor in the field of numerical optimization.

Initially a population of individuals are generated randomly corresponding to variables in the given search space. In PSO, each individual is termed, as particle while population is known as swarm. Basically, PSO is inspired by the flocking of birds in two-dimension space so each particle in the swarm has a position and velocity, which direct the flying of the particles. For each particle i, there is fitness value which is evaluated by the fitness function to be optimized, and one can find the optimal solution of the problem through the generation. In each iteration, each particle in PSO traces a trajectory in the search space; constantly updating a velocity vector by way of two kinds of search memories. One is the particle's best memory, called *pbest*, and the other is the swarm's best memory, called *gbest*. After iterations, the PSO can find the best solution according to the best solution memories based on the best solutions found so far by that particle as well as others in the swarm. The algorithm is expressed in following steps:

ISSN: 2278 – 1323

(i) *Generation of Population*

Initially a population of individuals is generated. In PSO, individuals are termed as particle and group of population is known as swarm. Each particle p at k^{th} iteration has velocity (Vel_u^k) and position (pos_u^k) within the search space.

(ii) *Fitness Function*

To find solution of the problem, fitness function is defined corresponding to the objective function. For each particle *p*, fitness function is evaluated.

(iii) *Find Local Best and Global Best*

After calculation of fitness function, the best fitness function for each particle and among swarm is found out. The best fitness function of particle *p* is known as Pbest and that of swarm is defined as *gbest*.

(iv) *At First Iteration*

After calculation of fitness, at iteration *u=1*, *gbest* is considered equal to *pbest*. Compare the fitness value of the particle *p* at *u+1* with that of the previous best one.

(v) *Updation of Velocity and Position Vector*

In next iteration *u=u+1*, update velocity and position vector of particle *p* using *gbest* and *pbest* till iteration *u+1* using following equations.

$$Vel_p^{u+1} = (wVel_p^u + c_1 rand_1(pbest - Pos_p^u) + c_2\, rand_2(gbest - Pos_p^u) \qquad (1)$$

$$pos_p^{u+1} = pos_p^u + Vel_p^{u+1} \qquad (2)$$

Eq. (1) consists of three terms on RHS; first term is the velocity in u^{th} iteration; second term is the cognition-only model and the third term is the so-called social-only model, these terms are utilized to change the velocity of particle.

a. After comparing the fitness of the particle and swarm, if the best fitness of the particle is superior to that of the swarm, then it modifies the memory of the swarm's best fitness and at the same time, every particle modifies the particle's velocity of the next generation.

b. If the search satisfies the termination condition then it stops; otherwise, it returns to step 2.

where rand1 and rand2 are random numbers generated in [0, 1]; c1 an c2 are acceleration constant; *w* is the inertia weight factor, it provides balance between global and local explorations. *w* often decreases from 0.9 to 0.4 during the iterations. It is generally set using the following equation:

$$w = w_{\max} - ((w_{\max} - w_{\min}) / k_{\max})*k \qquad (3)$$

where k_{max} is the maximum number of iterations and *k* is the current number of iteration.

(vi) *Stopping Criteria*

The above steps of subsection (ii) to (v) of Section A is repeated till the search satisfies the termination condition. The termination condition may be maximum number of iterations or the convergence criteria set.

B *Algorithm of Firefly Algorithm*

The proposed approach is based on firefly algorithm which mimics the behavior of fireflies. Fireflies flash light in the summer in sky in the tropical and temperate regions. The algorithm is based on the intensity of the flashes produced by the fireflies. They communicate each other with the help of intensity of flashes and fireflies tries to move toward the fireflies having high intensity of flashes. The light intensity changes with the distance from the other fireflies and some intensity is lost in medium. In this algorithm light intensity is calculated in the presence of some variables for intensity lost and distance between fireflies. The algorithm is sufficient random in nature and produces optimal solution in less computational time.

The algorithm of the proposed approach is as described below [7-9]:

The firefly (FA) is a recent nature inspired technique that has commonly been used for solving NP hard optimization problems. It is proposed by Xin-She Yang. For simplicity in describing Firefly Algorithm (FA), the following three idealized rules:

ISSN: 2278 – 1323

(a) All fireflies are unisex so that one firefly will be attracted to other fireflies regardless of their sex

(b) Attractiveness is proportional to their brightness, thus for any two flashing fireflies, the less brighter one will move towards the brighter one. The attractiveness is proportional to the brightness and they both decrease as their distance increases. If there is no brighter one 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.

(i) Light Intensity or Brightness

In firefly algorithm, brightness of each firefly is represented with the objective function to be maximized. For a maximization problem, the brightness can simply be proportional to the value of the objective function. In the simplest case for maximum optimization problems, the brightness I of a firefly at a particular location x can be chosen as $I(\mathrm{x})$ is proportional to $f(\mathrm{x})$.

(ii) Attractiveness towards Brightness

The movement of fireflies towards other high intensity fireflies is based majorly on attractiveness and light absorption. Fireflies having less brightness are attracted towards fireflies having more light intensity. The attractiveness (β) is relative, it should be seen in the eyes of the beholder or judged by the other fireflies. Thus, it will vary with the distance r_{ij} between firefly i and firefly j. In addition, 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 the degree of absorption. In the simplest form, the light intensity $I(r)$ varies according to the inverse square law:

$$I(r) = Is/r^2 \tag{4}$$

Where Is is the intensity at the source. For a given medium with a fixed light absorption coefficient γ, the light intensity I varies with the distance r. That is

$$\mathrm{I} = \mathrm{I}_0 e^{-\gamma r} \tag{5}$$

where $I0$ is the original light intensity.

As a firefly's attractiveness is proportional to the light intensity as seen by adjacent fireflies. It can be defined for a firefly as given below:

$$\beta = \beta_0 e^{-\gamma r^2} \tag{6}$$

Where β_0 is the at $r = 0$.In the implementation, the actual form of attractiveness function $\beta(r)$ can be any monotonically decreasing functions such as the following generalized form:

$$\beta(r) = \beta_0 \exp(-\gamma r_{ij}^m) \qquad m \geq 1 \tag{7}$$

(iii) Distance between Fireflies

The distance between two fireflies i and j at x_i and x_j is calculated using Cartesian distance as given below:

$$r_{ij} = \lVert x_i - x_j \rVert = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{8}$$

where $x_{i,k}$ is the k^{th} component of the spatial coordinate x_i of i^{th} firefly.

(iv) Movement of fireflies

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

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(rand - \frac{1}{2}) \tag{9}$$

In the above equation, the second term is due to the attraction while the third term is randomization introduced in the algorithm.

C Important Factors Influencing Performance of PSO and FA

In this section, various factors affecting performance of PSO and FA are discussed. Following are the major factors which need to be considered for proper convergence of solution.

ISSN: 2278 – 1323

(i) Size of Population

In evolutionary algorithms, size of population is important parameter for converge of the algorithm and quality of solution. In PSO, a large size of population is not considered. If a large size is considered it does not improve the quality of solution but increase the computational time. Therefore size of pos is kept small around 20-40. Similarly in Firefly algorithm a large size of population is not required for quality of solution. So it is can be concluded that both the algorithm does not require a large size population.

(ii) Selection of Parameters

In PSO, randomly within the search space velocity and position vectors are generated. In additions to this, acceleration constants (c1, c2) and inertia constant (w) are initialized within [0 1]. Inertia constant keeps on updating in the iterative process. Velocity and position vectors are modified for *pbest* and *gbest* of the population. While in Firefly Algorithm, various factors are β_0, α, γ which are initialized. In iterative process, β_0 is modified for given number of iterations.

(iii) Pre-tuning of parameters

In PSO and FA, pre-tuning of parameters is not required. All the parameters are updated during iterative process. But these parameters are important in other evolutionary algorithms like GA and ACS.

(iv) Randomness

In PSO and FA, randomly a population is generated and then variables of PSO and FA are generated randomly. But FA is very much random in nature. In third component of Eq (9) of firefly is randomization while PSO is not random in nature. In PSO, *gbest* and *pbest* always govern the updataion of population but there is no such term in velocity and position vector which is purely random in nature. FA has more randomness in nature which avoids local minima of the problem. In FA, the wrong selection of α can give a big or small step increment and takes away the solution in some other direction. It avoids local trapping but may take solution to another direction which is far away from global best. Therefore it is important to consider proper value of α, otherwise solution may be random but not the global one.

(v) Balance between local and global Minima

In PSO, there is balance between local and global minima during velocity and position vectors updation. PSO keeps a memory of its earlier iteration by storing values of pbest and gbest. So it can be concluded that there is balance between local and global minima in PSO. But in FA, the values of γ will decide the value β_0 which affects the search in local and global environment. If $\gamma \to 0$ which causes attractiveness $\beta = \beta_0$ .Thus, a flashing firefly can be seen anywhere in the domain. Thus, a single (usually global) optimum can easily be reached. This corresponds to a special case of particle swarm optimization (PSO). Subsequently, the efficiency of this special case is the same as that of PSO. While on the other hand $\gamma \to \infty$, $\beta(r) = \gamma(r)$, which means that the attractiveness is almost zero in the sight of other fireflies or the fireflies are short-sighted. This is equivalent to the case where the fireflies fly in a very foggy region randomly. No other fireflies can be seen, and each firefly roams in a completely random way. Therefore, this corresponds to the completely random search method. The value of α is another important factor which affects the performance by increasing and decreasing local search component of the movement of firefly.

(vi) Convergence Time

The convergence time of PSO and FA is not too high. But still FA convergence time is less as compared to PSO. In PSO, some earlier iterations data is stored and used in each iteration for updation. There is no information of earlier iterations in FA which makes it fast convergence algorithm.

(vii) Simplicity

PSO is simple to implement as compared to FA due to its less number of variables. Otherwise also PSO parameters are not problem specific. FA has more variables and most of them are random. The values of these variables in FA are more problem specific and their selection is important for the optimal solution of the solution.

(viii) Quality of Solution

The quality of solution for continuous variables of FA is better than PSO. Because of sufficient randomness, the solution obtained using FA is more promising than PSO. While PSO goes into local minima and can be improved using some other techniques in hybrid.

(ix) Applicability to Mixed Integer Problems

PSO is implemented in mixed integer problem in combination with other techniques while FA is successfully applied to continuous problems and is being applied in mixed integer variables by some researchers.

ISSN: 2278 – 1323

III. CONCLUSIONS

In this paper, two population based meta-heuristic approaches are developed. Both approaches are introduced and then important factors which needs to selected in a proper manner for quality solution and faster convergence. These parameters are so important which can increase or decrease the computational time without affecting or affecting the qulity of solution. The paper is conclusive study of two algorithms by keeping the important parameters into consideration. It is clear from the above discussion that PSO has limited applicability because of trapping in local minima which can be avoided by using in combination with PSO. Firefly does not face any problem regarding local minima because of sufficient randomness. PSO and FA are applied on continuous variables. Firefly is well developed for continuous variables and still in progress for discrete variables.

REFERENCES

[1]. J. Kennedy and R. Eberhart, "Particle Swarm Optimization," *IEEE Proceedings on 1st International Conference on Neural Networks (Perth, Australia)*, 1942-1948, 1995.
[2]. J. Kennedy and R. Eberhart, " A Discrete Binary Version of the Particle Swarm Algorithm," *IEEE Conference on Systems, Man, and Cybernetics, Orlando, FA*, 4104-4109, 1997.
[3]. K. Sedlaczek and P. Eberhard, "Constrained Particle Swarm Optimization of Mechanical Systems," *6th World Congresses of Structural and Multidisciplinary Optimization*, pp1-10, June 2005
[4]. L. C. Cagnina and S.C. Esquivel, "Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer," *Informatica* Vol *32*, 319–326, 2008.
[5]. H.Ganesan, G.Mohan Kumar, K.Ramesh Kumar, "Optimization of Machining Parameters in Tuning Process using Genetic Algorithm and Particle Swarm Optimization with Experimental Verification," vol. 3 No. 2, pp 1091-1102, Feb 2011.
[6]. F.Cus and U. Zuper, " Particle Swarm Intelligence based optimization of high speed end-milling, *Computational Materials Science and surface engineering*, vol 1, 148-154, 2002.
[7]. Z. A. Elizee, A. Babazadeh, S. Mohammad and S. Hosseini, "Optimizing Product Design through a Particle Swarm Induced Logistic Regression Model," *Majlesi Journal of Mechanical Engineering, Vol. 3/ No. 2/ Winter -2010*.
[8]. P.C. Fourie and A.A. Groenwold, "The particle swarm optimization algorithm in size and shape optimization," Structural Multidisciplinary Optimization, no 23, pp. 53-59, 2002.
[9]. E. Yang, A. T. Erdogan, T. Arslan and N. Barton, "An Improved Particle Swarm Optimization Algorithm for Power-Efficient Wireless Sensor Networks," *2007 ECSIS Symposium on bio-inspired learning and intelligent system for security*, 2007.
[10]. C.Y. Cheo and Fun Ye, "Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis," *IEEE International Conference on Networking, sensing Control Taiwan*, pp 789-794, 2004.
[11]. H. T. Geng, Y. H. Huang, J. Gao, and H. F. Zhu, "A self-guided particle swarm optimization with independent dynamic inertia weights setting on each particle," Applied Mathematics & Information Sciences, vol. 7, no. 2, pp. 545–552, 2013.
[12]. H. Yoshida, K. Kawata, Y. Fukuyama and Y. Nakanishi , "A particle swarm optimization for reactive power and voltage control considering voltage security assessment*," IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1232–1239, 2001..
[13]. B. Zhao, Q. Jiang, C. Guo, Y. Cao, "A Novel Particle Swarm Optmization Approach for Optimal Reactive Power Dispatch,"15th PSCC, Liege, 22-26, pp 1-6., August 2005.
[14]. M.A. Abido, "Optimal power flow using particle swarm optimization, "*International Journal of Electrical Power Energy Systems*," vol. 24, no. 7, pp. 563–571, 2002.
[15]. X. S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, Frome, UK, 2008.
[16]. X.-S. Yang, "Firefly algorithms for multimodal optimization," in Stochastic Algorithms: Foundations and Applications, vol. 5792 of Lecture Notes in Computer Science, pp. 169–178, Springer, Berlin, Germany, 2009.
[17]. X.-S. Yang, "Firefly algorithm, l´evy flights and global optimization,"in *Research and Development in Intelligent Systems*, vol. 26, pp. 209–218, Springer, Berlin, Germany, 2010.
[18]. A Postopoulous T,Valchos A "Application of FireFly algorithm for solving Economic Dispatch", *International Journal of Combinatronics* 2011.
[19]. X.-S. Yang, S. S. S. Hosseini, and A. H. Gandomi, "Firefly Algorithmfor solving non-convex economic dispatch problems with valve loading effect," *Applied Soft Computing Journal*, vol. 12, no. 3, pp. 1180–1186, 2012.
[20]. S. Łukasik and S. Zak, "Firefly algorithm for continuous constrained optimization tasks," in Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems, vol. 5796 of Lecture Notes in Computer Science, pp. 97–106, Springer, Heidelberg, Germany, 2009.
[21]. X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," International Journal of Bio-Inspired Computation, vol. 2, no. 2, pp. 78–84, 2010.
[22]. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Mixed variable structural optimization using firefly algorithm," Computers and Structures, vol. 89, no. 23-24, pp. 2325–2336, 2011.
[23]. G. K. Jati and Suyanto, "Evolutionary discrete firefly algorithm for travelling salesman problem," in Adaptive and Intelligent Systems, vol. 6943 of Lecture Notes in Computer Science, pp. 393–403, Springer, Heidelberg, Germany, 2011.
[24]. A. Kazem, E. Sharifi, F. K. Hussain, M. Saberi, and O. K. Hussain, "Support vector regression with chaos-based firefly algorithm for stock market price forecasting," Applied Soft Computing, vol. 13, no. 2, pp. 947–958, 2013.
[25]. X. S. Yang, "Multiobjective firefly algorithm for continuous optimization," Engineering with Computers, vol. 29, no. 2, pp. 175–184, 2013.

ISSN: 2278 – 1323