Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2015, 2(1): 35-41

Review Article ISSN: 2394 - 658X

A Chronological Review and Comparison of Four Evolutionary Based Algorithms

Akash Saxena[1], Bhanu Pratap Soni[2] and Vikas Gupta[2]

[1]Department of EE, Swami Keshvanand Institute of Technology, Jaipur, India
[2]Department of EE, Malaviya National Institute of Technology, Jaipur, India
er.bpsoni2011@gmail.com

ABSTRACT

This paper is an effort to present a cutting edge review and comparative analysis of four swarm based algorithms. The evolutionary algorithms are stochastic methods that take inspirations from the biological systems or social behavior of the species. The work presented here is a comparison of the algorithms on the basis of solution quality; times elapsed and success rate while keeping maximum iteration and population size same for all algorithms. The algorithms which are compared in this work are Gravitational Search Algorithm, Cuckoo Search algorithm, Particle swarm optimization and Genetic algorithm. Each algorithm is presented with pseudo code and flow chart to facilitate the researchers and practitioners. A special attention is thrown to felicitate and explain the unique features of the algorithms in the sections. All the algorithms are applied on benchmark functions specially the multimodal function to exhibit the efficacy and constraints while choosing between global and local optima.

Key words: Evolutionary algorithms, gravitational search algorithm, crossover, mutation, particle swarm optimization

---

INTRODUCTION

In optimization the problem is specified by a specific set of n parameters $\{x_1, x_2, x_3 \dots \dots x_n\}$ and an objective function f(x), which is also called fitness function. The basic gradient method related with the optimization is first found by Newton where the complex optimization problems are dealt with the gradient method. The major pitfall of this approach was immersed as the conditions imposed on objective function to be differentiable and continuous one.

$$G = \frac{\Delta f}{\Delta x_1} + \frac{\Delta f}{\Delta x_2} \dots \dots \dots \dots \frac{\Delta f}{\Delta x_n} \quad (1)$$

It is clear from the above equation shows that if the limited knowledge is exist for the objective function; the researcher has no choice left rather than imparting new solution methodology like evolutionary algorithms. An acute need of population based strategies is required because single point-to-point searches are generally insufficiently robust to overcome local pathologies. Evolutionary algorithms are a class of optimization problems which hails inspiration from the social behaviour of swarms, species and the theory of natural evolutions [1].All evolutionary algorithms are characterized by their different operators and their behavioural differences but common attributes in the algorithms are the stochastic behaviour, random search and selection [2]. To understand the behaviour of natural species like how ants find the shortest route to a source of food and how birds are able to migrate to remote places and find their destination. The behaviour of such species is guided by learning, adaptation and evolution [3-5] later on this term is called as a social intelligence. To mimic the efficient behaviour of these species, various researchers have developed computational systems that seek fast and robust solutions to complex optimization problems. The first evolutionary-based technique introduced in the literature was the genetic algorithms (GAs) [6]. GAs was developed based on the Darwinian principle of the 'survival of the fittest' and the natural process of evolution through reproduction. GAs may require long processing time for a near optimum solution to evolve. Also, not all

problems lend themselves well to a solution with GAs [6]. An attempt is made to improve the performance of GA in terms of processing time and solution quality and those are reported in the literature [7-8]. Wong et al used genetic/ simulated annealing approach for solving the economic load dispatch problem [9]. Chanan singh et al used two functions death penalty and penalty function to fight with premature convergence of GA and processing time procedure was named as atavistic genetic algorithm [10].

In addition with the experiments with the GAs in 1995 Particle swarm optimization (PSO) algorithm originally developed by Kennedy and Eberhart [11], which was depiction of social behaviour of flocks of birds and schools of fish. Similar to GAs, the PSO is also an optimizer based on population. Further bacterial foraging, Ant colony optimization and Artificial Bee colony methods are developed by various researchers to address the problem of continuous and discrete optimization. The organization of the paper is as follows in section II, III and IV brief details of various operators and stage associated with the GA, PSO, CSA and GSA. Section V presents a comparison of the algorithms while a bench mark function is chosen for recite the comparison.

GENETIC ALGORITHM

The Genetic Algorithm is a global search technique for solving optimization problems firstly developed by Holland in 1975, which is based on the theory of natural selection, the process that drives biological evolution. GA has proved to be a very effective and proficient tool for operation and control of power system and other complex control systems. Genetic algorithm is based on Darwinian Theory, survival of the fittest principle. To perform the optimization process, the GA employs three operators to promulgate its population from one generation to another. The first operator is the "Selection" operator that mimics the principal of "Survival of the Fittest". The second operator is the "Crossover" operator, which mimics mating in natural populations [6]. The crossover operator propagates features of good ongoing designs from the current population into the potential population, which will have better fitness value on average. The last operator is "Mutation", which promotes miscellany in population distinctiveness. The mutation operator allows for global search of the design space and prevents the algorithm from getting fascinated in local minima. Fig. 1 shows the basic building blocks of evolution process.

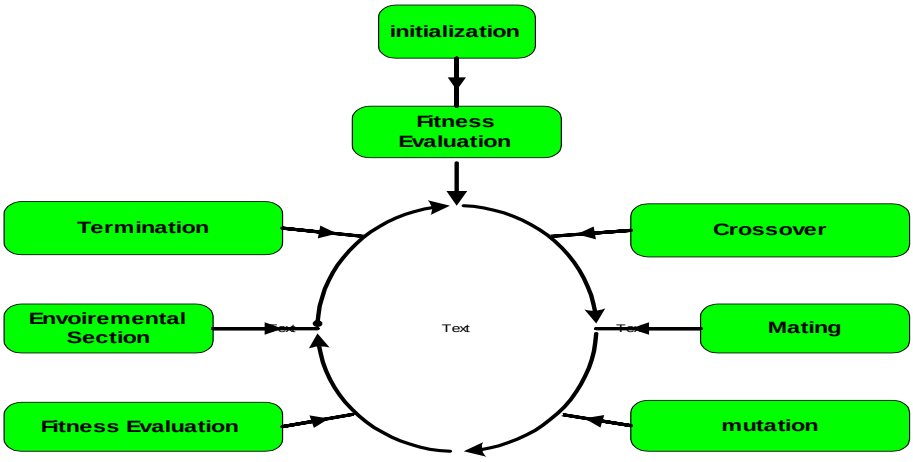

Fig.1 Basic Building blocks of Evolutionary algorithms

PARTICLE SWARM OPTIMIZATION

PSO is a kind of heuristic optimization algorithms. It is motivated from simulating certain simplified animal social behaviours such as bird flocking, and is first proposed by Kennedy and Eberhart in (R.ebehart,1995) [11], It is an iterative, population-based process. The particles are described by their two instinct properties: position and velocity. The position of each particle represents a point in the parameter space, which a possible solution of the optimization problem and the velocity is used to change the arrangement.

$$v_{ij}^{(k+1)} = v_{ij}^{k} + c_1 r_1 \left(P_{ij} - x_{ij}^{k}\right) + c_2 r_2 \left(P_{gj} - x_{ij}^{k}\right) \quad (2)$$

$$x_{ij}^{(k+1)} = x_{ij}^{k} + v_{ij}^{(k+1)} \quad (3)$$

$$i = 1,2,3.....N_p \qquad j = 1,2,3.....N_D$$

Where Np is the number of particles in the population; N_D is the number of variables of the problem (i.e. dimension of a particle); v_{ij}^k is the j^{th} coordinate component of the velocity of the i^{th} particle at iteration k; P_{ij} is the j^{th} coordinate component of the best position recorded by the i^{th} particle during the previous iterations; P_{gj} is the j^{th} coordinate component of the best position of the global best particle in the swarm, which is marked by g; x_{ij}^k is the j^{th} coordinate component of the current position of particle i at the k^{th} iteration; ω is the inertia weight, c_1, c_2 are the acceleration coefficients, r_1, r_2 are the uniformly distributed random values between 0 and 1.

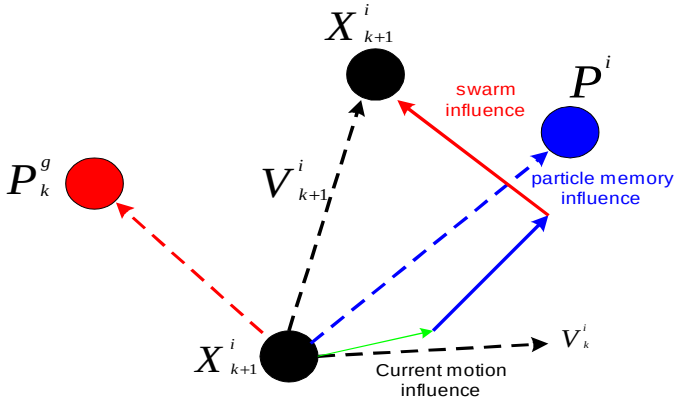

Fig. 2 Anatomy of particle

CUCKOO SEARCH ALGORITHM

The CSA is introduced by yang and Deb in 2009 [12-13]. The CS was inspired by compel brood parasitism of cuckoo species by laying their eggs in the nests of host birds. Some cuckoos have evolved in such a way that female scrounging cuckoos can reproduce or rather imitate the colors and patterns of the eggs of a few chosen host species. This reduces the probability of the eggs being abandoned. Yang and dev also suggested in their research that levy flights is useful for improve solution quality besides on random walk method. A Levy flight is a random walk in which the step-lengths are distributed according to a heavy-tailed probability distribution. Each egg in a nest represents a solution .the objective it to employ good quality solution in the nest and replace those which are not so good solution. The algorithm based on three idealized rule:

- Each Cuckoo laid one egg only, and dumps the egg in a randomly chosen nest.
- The best nest (with quality solutions) will carry over the next generation.
- The number of available nest is fixed and a host can identify an alien egg with probability P_a [0, 1].

Based on the foresaid rules while generating new solutions

$$x_i^{t+1} = x_i^t + \alpha \oplus Levy(\lambda) \tag{4}$$

Where α>0 is the step size, in this work it is taken 1.Random walk is a markov chain whose next location is depends on the current location (the first term in above equation) and the probability of the transition.

Primary observations about algorithm develop a sense of resemblance with hill climbing in combination with some large scale randomization. However the algorithm is a population based algorithm similar to GA and PSO but randomization of the patterns is done in a more efficient way as the step length is heavy tailed. Thirdly the parameters to be tuned is less than GA and PSO, For these reasons the Cuckoo search found to be very generic and used for wide no of optimization problems.

GRAVITATIONAL SEARCH ALGORITHM

Rashedi et al anticipated a new meta-heuristic algorithm called GSA in year 2009[14-16]. A beautiful analogy between Newton's gravitational laws with the optimization prototype of the era is presented in the algorithm. The postulates of the algorithm say that every particle attracts towards each other and force exerted between two objects (agents) is proportional to the mass of the objects and inversely proportional to square of the distance between them. Force causes a global movement of all objects towards the objects with heavier mass. Heavier mass is analogous to the agent which has higher fitness values. GSA proposes four prepositions of a gravitational mass: its position, inertial mass, gravitational mass (active and passive). The position of mass is representation of a solution and masses are specified by fitness of a function. It is assumed that given a system with N agents in search space represents solution to a problem. Equation represents space dimension and the position of the agent in x_i^d d^{th} dimension.

$$X_i = \left(x_i, \ldots\ldots x_i^d, \ldots\ldots x_i^n\right), \quad for\ i = 1,2,\ldots\ldots, N \tag{5}$$

According to the Newton's law of attraction the force exerted by i^{th} mass due to j^{th} mass at time t represented by equation(6).

$$F_{ij}^d(t) = G(t)\frac{M_{pi}(t) \times M_{qj}(t)}{R_{ij}(t) + \varepsilon}\left(x_j^d(t) - x_i^d(t)\right) \tag{6}$$

Where $M_{pi}(t), M_{qj}(t)$ are active and passive gravitational mass, $G(t)$ is gravitational constant at time t and R_{ij} is euclidian distance between i and j agents defined by equation (7).

$$Rij(t) = \left\| X_i(t), X_j(t) \right\|_2 \tag{7}$$

Force exerted on an agent i is randomly weighted sum of the forces exerted from other agents.

$$F_i^d(t) = \sum_{j=i\, j\neq i}^{m} rand_j\, F_{ij}^d(t) \tag{8}$$

Acceleration of the agent at time t in the d^{th} dimension on law of motion is used directly to calculate the force. In accordance with this law, acceleration is proportional to the force exerted and inversely proportional to mass of the agent.

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \tag{9}$$

Searching strategy of the algorithm is defined by updating velocity and position at time t and in d dimension.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{10}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{11}$$

The gravitational constant G, randomly at the starting and according time to control the search accuracy G is exponentially decayed.

$$G(t) = G(G_0, t) \tag{12}$$

$$G(t) = G_0\, e^{-\frac{\alpha t}{T}} \tag{13}$$

There α is a user specified constant, t is the current iteration and T is the total number of iterations.

$$M_{ai} = M_{pi} = M_{ii} \qquad i = 1,2\ldots\ldots, N \tag{14}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \tag{15}$$

A heavy mass has a higher pull on power and moves slower so at the end of iteration the masses obtain will be having high on gravity and value of fitness is more.

Equation (16), (17) and (18), where *fit(t)* represent fitness value of the agent at time t and best and worst masses in population. In order to solve optimization problem each agent is specified with the position after each iterations the fitness is calculated and position and velocity of the agents are updated with each iteration ease of use.

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{N} m_j(t)} \tag{16}$$

For minimization problem

$$worst(t) = \max_{j\in\{1\ldots..m\}} fit_j(t) \tag{17}$$

$$best(t) = \min_{j\in\{1\ldots..m\}} fit_j(t) \tag{18}$$

For maximization problem

$$best(t) = \max_{j\in\{1\ldots..m\}} fit_j(t) \tag{19}$$

$$worst(t) = \min_{j\in\{1\ldots..m\}} fit_j(t) \tag{20}$$

COMPARATIVE ANALYSIS

A bench mark function with 27 variables function is considered to show the efficacy of the various algorithms to reach the global maxima. A complex function with 27 Variables is observed in between range of x [0, 1].

$$f(x) = a(i)\sin\left(x(i) - \pi\right) + b(i)\left(x(i) - 10\right)^2 + c(i) \tag{20}$$

Function is optimized i.e. maximized in this range between it has three maxima in this range. To make the problem more explicit and pragmatic the range of variables are described as per table I and no of population and maximum iterations are kept constant for this optimization process so that the true comparison can be done and effect of various operators immersed in solution quality. Comparison of the optimization process by different algorithms are observed by running the same optimization 20 times, the standard deviation of the value of objective function is obtained. while running the GA performs very sluggish and solution quality obtained from it is very poor, Success rate of the algorithm is defined as the ability of the algorithm to provide potential solutions in each run, it is observed from here that GA lags in this case also Fig. 3 & 4 shows that while observing standard deviations in the value of objective functions GSA and CSA performs almost same and values of deviations obtained are 0.412 and 0.397, surprisingly GA performs poorer with the highest standard deviation of 2.23 as shown in Fig. 3. The success rate is compared and it is comes out be highest for CSA as shown in Fig. 4.

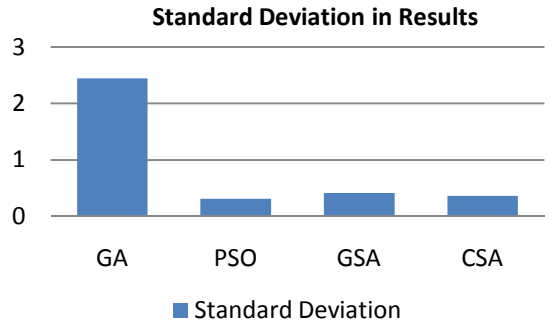

Fig. 3 Standard deviations of objective function after 20 runs

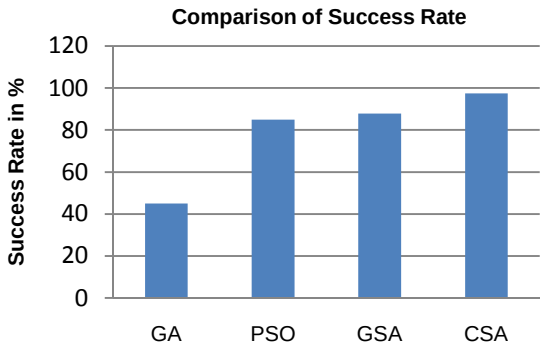

Fig.4 Success Rate of various different algorithms

Table -1 Various Coefficients for Objective Function

i	a	b	c
1	0.1208	0.03109	7.099
2	2.733	0.1466	18.84
3	0.09403	-0.00061	5.322
4	-0.7725	0.03602	0.6032
5	1.085	-0.0436	8.654
6	0.005116	4.15E-06	4.293
7	0.8223	0.01611	3.272
8	-2.182	0.1099	-6.856
9	0.05325	-0.00452	5.876
10	-0.9487	0.02624	1.64
11	-6.788	0.2705	-23.07
12	-0.0872	-0.00044	4.983
13	1.775	-0.03884	8.158
14	4.532	-0.1093	16.05
15	0.06301	-0.00075	5.47
16	0.9064	0.01638	3.525
17	-17.05	0.793	12.56
18	0.0945	-0.00045	5.202
19	-1.745	0.04161	0.5704
20	-2.33	0.0787	0.98
21	-0.01273	-0.00041	4.866
22	-0.3194	0.006244	3.551
23	-0.41	0.01295	2.92
24	0.08289	-0.00018	4.573
25	2.522	-0.08352	12.67
26	-0.4188	0.03031	1.17
27	0.000604	-0.0001	6.802

CONCLUSION

A Critical analysis is presented here to evaluate the proficiency of the different swarm based topologies. The objective function chosen over here is of 27 variables and has complex bounds of the variable. Observations reveal that GA gives the solutions which are near the lower and upper bound, similarly the standard deviations are also found maximum in case of GA. The objective function is a linear polynomial the decisive evaluation is performed on the basis of solution quality is performed and comparison of parametric variation is given standards for algorithms are given in Appendix B.

APPENDIX A

Pseudo Code for a GA Procedure

Begin;

Generate random population of solutions
(Chromosomes);
For each individual i ε population: calculate fitness (i);
For *i=1 to number of generations;*
Randomly select an operation
(Crossover or mutation);
If *crossover;*
Select two parents at random i_a and i_b;
Generate on offspring i_c=crossover (i_a and i_b);
Else If *mutation;*
Select one chromosome i at random;
Generate an offspring i_c=mutate (i);
End if*;*
Calculate the fitness of the offspring i_c;
If *i_c is better than the worst chromosome then*
Replace the worst chromosome by i_c;
Next i;
Check if termination=true;

End;

Pseudo Code for a PSO Procedure

Begin;

Generate random population of N solutions (particles);
For *each individual iε N: calculate fitness (i);*
Initialize the value of the weight factor, u;
For *each particle;*
Set p Best as the best position of particle i;
If *fitness (i) is better than P_{Best};*
pBest(i)=fitness (i);
End*;*
Set gBest as the best fitness of all particles;
For each particle;
Calculate particle velocity according to Eq. (1);
Update particle position according to Eq. (2);
End;
Update the value of the weight factor, u;

End;

Pseudo Code for CSA

Begin

Objective function f(x), x = $(x_1, \ldots, x_d)^T$
Generate initial population of n host nests xi (i = 1,2, …, n)
while *(t <Max Generation) or (stop criterion)*
Get a cuckoo randomly by L´evy flights Evaluate its quality /fitness F_i
Choose a nest among n (say, j) randomly
if *($F_i > F_j$),*
Replace j by the new solution;
end

A fraction (p_a) of worse nests are abandoned and new ones are built;
Keep the best solutions(or nests with quality solutions);
Rank the solutions and find the current best
end while
Post process results and visualization
end

APPENDIX B

a) Parameter for GA
   i. Population size=50,
   ii. Maximum no of generations =1000,
   iii. Crossover =8e-1
   iv. Mutation Probability =1e-3.
b) Parameter for PSO
   i. No. of Particle=50,
   ii. Inertia=0.4,
   iii. C_1 & C_2 =2.
c) Parameter for GSA:
   i. α=20;
   ii. G_0=100;
   iii. N=50;
   iv. Maximum Iteration = 1000;
d) Parameter for CSA:
   i. Number of nests (different solutions)=50
   ii. Discovery rate of alien eggs/solutions pa=0.25;

Acknowledgements
The authors acknowledge the support and encouragement of Malaviya National Institute of Technology, Jaipur and Swami Keshvanand Institute of Technology, Jaipur.

REFERENCES

[1] K D Jong, Evolutionary Computation: a Unified Approach, *The Genetic and Evolutionary Computation Conference (GECCO 07)*, **2007**.
[2] V Oduguwa, A Tiwari and R Roy, Evolutionary Computing in Manufacturing Industry: An Overview of Recent Applications, *Applied Soft Computing*, vol. 5, **2005**, p. 281-299.
[3] S K Pal, S Bandyopadhyay and S S Ray, Evolutionary Computation in Bioinformatics: A Review, *IEEE Transactions on Systems Man and Cybernetics Part C-Applications and Reviews*, **2006**, vol. 36, p. 601-615.
[4] C A Peña-Reyes and M Sipper, Evolutionary Computation in Medicine: An Overview, *Artificial Intelligence in Medicine*, **2000**, vol. 19, p. 1-23.
[5] T Mantere and J T Alander, Evolutionary Software engineering, A Review, *Applied Soft Computing*, **2005**, vol. 5, .p 315-331.
[6] J Holland, *Adaptation in Natural and Artificial Systems Ann Arbor*, MI: University of Michigan Press; **1975.**
[7] H Al-Tabtabai and P A Alex, Using Genetic Algorithms to Solve Optimization Problems in Construction Eng, *Constr Archit Manage*, **1999**, Vol. 6(2), p. 121–132.
[8] O J Mengshoel and D E Goldberg, The Crowding Approach to Niching in Genetic Algorithm, *Evolutionary Computation*, **2008**, vol.16, no.3, p. 315-354.
[9] K P Wong, Y W Wong, Genetic and Genetic/Simulated-Annealing Approaches to Economic Dispatch, IEE Proc., **1994,** Vol. 141 (5), p. 507- 513.
[10] Chanan singh et al, Atavistic Genetic Algorithm for Economic Dispatch with Valve Point Effect, *Electric Power Systems Research*, **2002**, 62, p. 201-207.
[11] J Kennedy and R C Eberhart, Particle Swarm Optimization, Proc IEEE Conference on Neural Networks, Piscataway, NJ, **1995,**Vol. 4, p. 1942-1948.
[12] X S Yang and S Deb, Cuckoo Search Via Lévy Flights, *IEEE World Congress on Nature & Biologically Inspired Computing (NaBIC 09)*, **2009**, p. 210–214.
[13] X S Yang, *Nature-Inspired Meta heuristic Algorithms*, 2nd Edition, Luniver Press, **2010.**
[14] E Rashedi, H Nezamabadi-pour and S Saryazdi, GSA: A Gravitational Search Algorithm, *Information Science*, **2009**, Vol. 179, p. 2232– 2248.
[15] E Rashedi, H Nezamabadi-pour and S Saryazdi, Filter Modelling using Gravitational Search Algorithm, *Engineering Applied Artificial Intelligence*, **2011**, Vol. 24, p. 117–122.
[16] E Rashedi, H Nezamabadi-pour and S Saryazdi, BGSA: Binary Gravitational Search Algorithm, *Nat Comput*, **2010**, Vol. 9, p. 727–745.