Scientific Journal of Impact Factor(SJIF): 3.134
e-ISSN(O): 2348-4470
p-ISSN(P): 2348-6406
International Journal of Advance Engineering and Research Development
Volume 1,Issue 12, December -2014

SURVEY OF DIFFERENT SWARM INTELLIGENCE ALGORITHMS

Monali Patil[1], Jasmine Irani[2], Vandana Jagtap[3]
[1]*patil.monali123@gmail.com*,[2]*jasmineirani26@yahoo.in*,[3]*vandana.jagtap@mitpune.edu.in.*
[1,2,3]*Computer Department, MAEER's MIT, Pune, India.*

***Abstract*—*Swarm intelligence is an increasingly popular domain in artificial intelligence and computer science with emergent properties. It takes inspiration from nature, especially biological systems such as ant colonies, flocks of birds, fishschools and bee hives. Most famous examples of swarm intelligence algorithm include Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO). The vital objective of these algorithms is to employ many simple agents applying almost no rule which in turn leads to an emergent global behavior. This paper presents review of swarm intelligence algorithms and their application for problem solving.*

***Keywords*—*Swarm Intelligence, Artificial Intelligence, Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO).*

I. INTRODUCTION

The expression Swarm Intelligence (SI) was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems[1].Swarm intelligence can be described as mutual behavior came out from social insects working under very few rules. Individual agents do not posses much ability by themselves. However in a colony of agents, each agent performs its own task independently which are related to each other in such a way that the colony has a capability of solving complex problems through cooperation. This collective behavior which comes out from agents is called as swarm intelligence. Apart from performing basic computations, swarms are expected to response to quality factors such as food, safety etc. and adapt to environment fluctuations.

Self organization is the main theme with limited restrictions from interaction among agents[2].The agent follows very simple rules and even thought there is no central structure available about how individual agent should behave, interaction between such agents lead to emergence of intelligent global behavior which is unknown to individual agent. Design of SI systems involves identification of analogies in swarm biology and IT systems, understanding modelling of realistic swarm biology and tuning models for IT applications. In principle, it should be a multi agent system that has self organized behavior and that exhibits some intelligent actions. The main goal of SI systems is performance optimization and robustness.

II. SWARM ALGORITHMS

The fundamental feature of swarm intelligence is that multiple self interested agents can work together without any central control. These agents as a population can exchange related data, by chemical message-carrier (pheromone by ants), by dance (waggle dance by bees) or by broadcasting ability (such as the global best in PSO). Therefore, all swarm based algorithms are population based. However, all population based algorithms are not swarm based algorithms. Also swarm based algorithms takes inspiration from nature, but not all of them may be considered under SI. For example artificial neural networks and evolutionary algorithms are geographically inspired algorithms but they do not come under swarm intelligence as here individual agents do not operate through demonstration space.

A. Particle Swarm Optimization (PSO) algorithm

Social behavior and changing movements of insects, birds and fish are the inspiration from nature behind particle swarm optimization. It was first introduced by Kennedy and Eberhart in 1985[3]. Imitating physical quantities such as velocity and position in bird flocking, artificial particles are constructed to fly inside the search space of optimization problem. The information for the particle includes knowledge gained from its previous experience and knowledge gained from the swarm. The value of the particle calculated using objective function updates information of particle and optimizes the objective of swarm. Therefore, swarm can converge to develop good resolution in local regions of the problem space; common objective can also be updated when any particle finds better objective that can lead the swarm in exploring different region of the problem space. PSO algorithms are easy to implement and achieves global optimal solution with higher probability [4].

PSO Algorithm

Basic algorithm as proposed by Kennedy and Eberhart,

$\boldsymbol{x}_k^i$ - Particle position

$\boldsymbol{v}_k^i$ - Particle velocity

$\boldsymbol{p}_k^i$ - Best remembered individual particle position

$\boldsymbol{p}_k^g$ - Best remembered swarm position

c_1, c_2 - Cognitive and social parameters

r_1, r_2 - Random numbers between 0 and

The positions will be updated simply by [],

$$\boldsymbol{x}_{k+1}^i = \boldsymbol{x}_k^i + \boldsymbol{v}_{k+1}^i,$$

With the velocity calculated as follows,

$$\boldsymbol{v}_{k+1}^i = \boldsymbol{v}_k^i + c_1 r_1 (\boldsymbol{p}_k^i - \boldsymbol{x}_k^i) + c_2 r_2 (\boldsymbol{p}_k^g - \boldsymbol{x}_k^i).$$

The importance of including best remembered swarm position $\boldsymbol{p}_k^g$ for each particle is to avoid the swarm being trapped into a local minimum.

1. Initialize
   (a) Set constants k_{max}, c_1, c_2.
   (b) Randomly initialize particle positions $\boldsymbol{x}_0^i \in \boldsymbol{D}$ in $\mathbb{R}^n$ for $i = 1, ..., p$.
   (c) Randomly initialize particle velocities $0 \leq \boldsymbol{v}_0^i \leq \boldsymbol{v}_0^{max}$ for $i = 1, ..., p$.
   (d) Set $k = 1$
2. Optimize
   (a) Evaluate function value f_k^i using design space coordinates $\boldsymbol{x}_k^i$.
   (b) If $f_k^i \leq f_{best}^i$ then $f_{best}^i = f_k^i$, $\boldsymbol{p}_k^i = \boldsymbol{x}_k^i$.
   (c) If $f_k^i \leq f_{best}^g$ then $f_{best}^g = f_k^i$, $\boldsymbol{p}_k^g = \boldsymbol{x}_k^i$.
   (d) If stopping condition is satisfied then goto 3.
   (e) Update all particle velocities $\boldsymbol{v}_k^i$ for $i = 1, ..., p$ with rule (2.1).
   (f) Update all particle positions $\boldsymbol{x}_k^i$ for $i = 1, ..., p$ with rule (2.2).
   (g) Increment k.
   (h) Goto 2(a).
3. Terminate

Applications of PSO

Particle swarm optimization algorithm is most suited for continuous variable problems. It has successfully been applied to a wide variety of problems. Applications of PSO include training of neural networks, optimization of electric power distribution networks, in structural optimization for optimal shape and sizing design and for topology optimization, process biochemistry and for system identification in biomechanics.

B. **Ant colony optimization (ACO) algorithm**

Ant colony optimization is one of the most recent techniques for approximate optimization. First ACO, called the ant system was proposed by Marco Dorgio in 1992. The inspiring source of ACO algorithms are real ant colonies. More specifically, ACO is inspired by the ants foraging behavior. At the core of this behavior is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food source. This characteristic of real ant colonies is exploited in ACO algorithms in order to solve optimization problems.

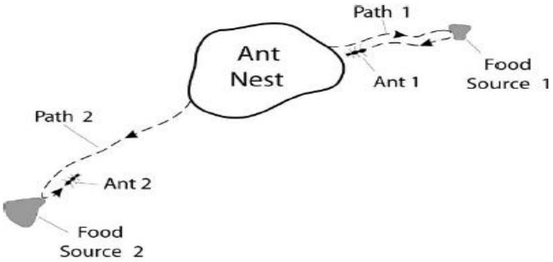

Fig. Ant foraging behavior

Every path followed by an ant is associated with a candidate solution for a given problem. When an ant follows any path, the amount of pheromone deposited on that path is proportional to the quality of the respective candidate solution for the target problem. When an ant has to choose between two or more available paths, the path with larger composition of pheromone has a greater probability of being chosen by the ants. As a result, the ants eventually converge to a short path, which is an optimal or near optimal solution for the target problem.

ACO Algorithm

```
Initialize trail
Do while
        Do until (Each ant completes a Tour)
                Local Trail update
        End Do
        Analyze Tours
        Global Trail update
End Do
```

Pheromone values for promising solutions will be increased and values for undesired solutions will be decreased by pheromone evaporation. Thus the best solution will have the highest concentration of pheromones. The performance of ACO algorithm largely depends on if an optimal local search procedure can be found and this is very problem specific [5].

Applications of ACO

ACO algorithms have been tested on a large number of academic problems. These include problems related to the traveling salesman, as well as assignment, scheduling, subset, and constraint satisfaction problems. For many of these, world-class performance has been achieved. ACO algorithms can adapt to changes in real time. This is of interest in network routing, urban transportation system, facility placement and in scheduling problems. However ACO algorithm is also applied for problems like data mining, image processing, set problems etc. This success with academic problems has raised the attention of a number of companies that have started to use ACO algorithms for real-world applications.

C. The Bees Algorithm

The inspiring source of bees algorithms is the food foraging behavior of swarms of honey bees. The bees algorithm in its basic formulation was created by Pham and his co-workers in 2005[6].

A colony of honey bees can extend itself over long distances and in multiple directions simultaneously to exploit a large number of food sources. A colony prospers by deploying its foragers to good fields. In principle, flower patches with plentiful amounts of nectar or pollen that can be collected with less effort should be visited by more bees, whereas patches with less nectar or pollen should receive fewer bees. When they return to hive those scout bees with patch above a certain quality threshold deposit their nectar or pollen and perform a dance known as waggle dance. This dance is essential for colony communication and contains three pieces of information regarding a flower patch i.e. the direction in which it will be found, its distance from the hive and quality rating [7]. This gathered information helps the colony to send its bees to the flower patches without using guides or maps. To decide upon the next waggle dance bees monitor its food level when they return to hive. This results in quick and efficient gathering of food in a colony.

Bees Algorithm

The algorithm requires a number of parameters to be set, number of scout bees (n), number of sites selected out of n visited sites (m), number of best sites out of m selected sites (e), number of bees recruited for best e sites (nep), initial size of patches (ngh) which includes site and its neighborhood and stopping criterion. The algorithm starts with the n scout bees being placed randomly in the search space.

```
Initialize population with random solutions.
Evaluate fitness of the population.
While (stopping criterion not met)
                // Forming new population.
        Select sites for neighborhood search.
        Recruit bees for selected sites (more bees for best e  sites) and evaluate fitnesses.
        Select the fittest bee from each patch.
        Assign remaining bees to search randomly and evaluate their fitnesses.
End While.
```

Applications of Bees algorithm

The bees algorithm has found many applications in engineering such as, optimization of classifiers or clustering systems, for mechanical designs like design of coil spring, in bioengineering, in data clustering for solving the local optimum and for multi objective optimization.

D. Grey Wolf Optimizer Algorithm

The grey wolf optimizer (GWO) algorithm imitates the leadership hierarchy and hunting mechanism of gray wolves in nature. The algorithm was proposed by Mirjalili et al. in 2014[8]. Wolves are social predators that hunt in packs. Four types of grey wolves such as alpha, beta, delta and omega are involved in simulating the leadership hierarchy. In addition, three important steps of hunting, searching for prey, encircling prey and attacking prey, are implemented in GWO algorithm to perform optimization.

Grey wolf belongs to candidate family. They are considered as apex predators which means that they are at the top level of food chain. The leaders are a male and a female, called as alphas. The alpha wolf is also called as dominant wolf since they are responsible for making decisions about hunting, sleeping place, time to wake etc and their orders are followed by the family. The second level in the hierarchy of grey wolves is beta. These are the subordinate wolves that help the alpha in decision making and in other activities. The lowest ranking grey wolf is omega. The omega plays the role of scapegoat. They are expected to submit to all the dominant wolves. If a wolf is not an alpha, beta or omega then the wolf is called as subordinate or delta. Delta wolves have to submit to alphas and betas, but they dominate the omega.

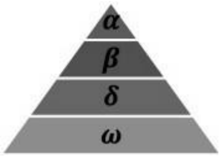

Fig. Social hierarchy of grey wolves.

GWO Algorithm

The grey wolf optimizer algorithm has only two parameters to be adjusted a and c which are used to assist candidate solutions to have hyper-spheres with different random radii. The adaptive values of A allow GWO to smoothly transition between exploration and exploitation. With decreasing value of A, half of the iterations are devoted to exploration and the other half are dedicated to exploitation.

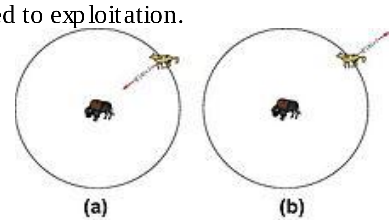

Fig. Effects of A on the exploration and exploitation of GWO algorithm.

Initialize the grey wolf population Xi (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α =the best search agent
X_β =the second best search agent
X_δ =the third best search agent

- while (t < Max number of iterations)
  - for each search agent
    - Update the position of the current search agent by above equations
  - end for
  - Update a, A, and C C
  - Calculate the fitness of all search agents
  - Update X_α, X_β, and X_δ
  - t=t+1
- end while

return X_α

Applications of GWO algorithm

Wolves that live and hunt together remain as a nuclear family which is different from other optimization algorithms, which usually move in relatively large groups. So GWO algorithm needs no intercommunication during the search. As a result, algorithm can explore large search space for a problem. Grey wolf optimizer algorithm is applicable for challenging problems with unknown search spaces. It gives high performance for unconstrained as well as constrained problems.

E. **Bat Algorithm**

The bat algorithm is a bio-inspired algorithm developed by Yang in 2010[9] and has been found to be very efficient. The algorithm is a based on echolocation or bio-sonar characteristics of microbats.

Microbats typically use a type of sonar, called echolocation to detect prey, avoid obstacles and locate their roosting crevices in the dark. They can emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species. Bat algorithm has a distinct advantage over other metaheuristic algorithms. That is, it has a capability of automatically zooming into a region where promising solutions have been found. This zooming is accompanied by the automatic switch from explorative moves to local intensive exploitation. As a result, bat algorithm has a quick convergence rate, at least at early stages of the iterations, compared with other algorithms [9].

Bat Algorithm
Based on the characteristics of bat echolocation, Xin-She Yang (2010) developed the bat algorithm with the following three idealized rules:
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 belongs to set [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.

Applications of bat algorithm

Since the bat algorithm has been developed, Bat algorithms have been applied in almost every area of optimization, classifications, image processing, feature selection, scheduling, data mining and others. Yang et al. use the bat algorithm to study topological shape optimization in microelectronic applications so that materials of different thermal properties can be placed in such a way that the heat transfer is most efficient under stringent constraints. Komarasamy and Wahi studied K-means clustering using bat algorithm and they concluded that, the combination of both K-means and bat algorithm can achieve higher efficiency and thus performs better than other algorithms. Du and Liu presented a variant of bat algorithm with mutation for image matching, and they indicated that their bat-based model is more effective and feasible in imagine matching than other models such as differential evolution and genetic algorithms.

III. CONCLUSION

This paper comprise of five important nature inspired swarm intelligence algorithms. Particle swarm optimization algorithms have the simplest framework, allowing particles to search for the optimal solution directly. With the help of social interactions and related population topologies, PSO algorithms are able to avoid local minimums and search for global optimal solution more efficiently. The ant and bee colony optimization algorithms are based on metaheurisitic derived from their feedback systems pheromones and waggle dances respectively. Grey wolf optimizer algorithm is applied to unknown search space problems. Bat algorithm has been exercised for almost every optimization problem. All these five algorithms have rich applications in problem solving.

ACKNOWLEDGEMENT

We would like to take this opportunity to thank people who have been constantly supporting and encouraging us in matters, big or small. Firstly we would like to thank our Professors for their valuable guidance. Furthermore, we would like to thank anonymous referees for helping immensely to improve this paper.

REFERENCES

[1] Beni, G. Wang, J. “Swarm Intelligence in Cellular Robotic Systems”,Advanced Workshop on Robots and biological systems,Italy, June 1989.
[2] Yichen Hu, “Swarm Intelligence”.
[3] J. Kennedy, and R.C. Eberhart,” Particle swarm optimization”, IEEE International Conference on Neural Networks,1995.
[4] Dian Palupi Rini, Siti Mariyam Shamsuddin and Siti Sophiyati Yuhaniz,”Particle Swarm Optimization: Techniques, System and challenges”, International journal of computre applications, January 2011 .
[5] M. Dorgio, T. Sttzle, and M. Birattari,” Ant colony optimization”, Computational Intelligence Magazine, 2006.
[6] D.T.Pham, A. Ghanbarzadeh, E.Koc, S Otri, S. Rahim, M. Zaidi, “The Bees Algorithm – A Novel Tool For Complex Optimisation Problems”, Manufacturing Engineering Centre, Cardiff University,UK.
[7] Camazine S, Deneubourg J, Franks NR, Sneyd J, Theraula G and Bonabeau E.,” Self-organization in Biological Systems”, Princeton University Press, 2003.
[8] S.Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey Wolf Optimizer,” Advances in Engineering Software, 2014.
[9] Dr Ajay Jangra, Adima Awasthi, Vandana Bhatia, “A Study on Swarm Artificial Intelligence”,IJARCSSE,August 2013.
[10] Xin-She Yang,”Bat Algorithm: Literature Review and Applications”,School of Science and Technology,London,2013.
[11] Nada M. A. Al Salami,”Ant Colony Optimization Algorithm”,UbiCC Journal, August 2009.
[12] Rui Tang, Simon fong, Xin-She Yang, Suash Deb,” Wolf Search Algorithm with Ephmeral Memory”,IEEE,2012