



A Conceptual Comparison of Artificial Bee Colony and Particle Swarm Optimization

R. Sabin Begum¹, Dr. R.Sugumar²

Research and Development Centre, Bharathiar University, Coimbatore, India¹

Associate Professor, Department of Computer and Engineering, Velammal Institute of Technology, Chennai, India²

ABSTRACT: For the last two decades, nature inspired metaheuristic algorithms have shown their ubiquitous nature in almost every aspect, where computational intelligence is used. This paper intends to focus on the comparative study of two popular and robust bio mimic strategies used in computer engineering, namely Particle Swarm Optimization (PSO) and Artificial Bee Colony (ABC). According to the results, PSO outperforms ABC. The performance comparison of both algorithms is implemented in the form of problem specific distance functions rather than an algorithmic distance function. Also an attempt is taken to examine the claim that PSO has the same effectiveness of finding the true global optimal solution as the ABC but with significantly better computational efficiency, which means less function evaluations.

KEYWORDS: Metaheuristic Algorithms, Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO).

I. INTRODUCTION

The last few decades have witnessed the introduction of several optimization algorithms developed based on nature-inspired ideas. Some examples of such algorithms include ant colony optimization [1], evolutionary algorithm [2], particle swarm optimization [3], harmony search [4] etc. Most of these algorithms are metaheuristic-based search techniques and generally referred to as multipurpose optimization algorithms because of their applicability to a wide range of problems. In a similar context, Artificial Bee Colony algorithm (ABC) was initially published by Karaboga in 2005 as a technical report for numerical optimization problems [5].

Particle Swarm Optimization (PSO) was invented by Kennedy and Eberhart in the mid 1990s [6] while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a socio cognitive study investigating the notion of “collective intelligence” in biological populations. In PSO, a set of randomly generated solutions (initial swarm) propagates in the design space towards the optimal solution over a number of iterations (moves) based on large amount of information about the design space that is assimilated and shared by all members of the swarm. PSO is inspired by the ability of flocks of birds, schools of fish, and herds of animals to adapt to their environment, find rich sources of food, and avoid predators by implementing an “information sharing” approaches, hence, developing an evolutionary advantage.

The Artificial Bee Colony (ABC) Algorithm is a swarm based meta-heuristic algorithm that was introduced by Karaboga in 2005 for optimizing numerical problems. The ABC consists of three groups of artificial bees: employed foragers, onlookers and scouts. The employed bees comprise the first half of the colony whereas the second half consists of the onlookers [7]. The employed bees are linked to particular food sources. In other words, the number of employed bees is equal to the number of food sources for the hive. The onlookers observe the dance of the employed bees within the hive, to select a food source, whereas scouts search randomly for new food sources. Analogously in the optimization context, the number of food sources (that is the employed or onlooker bees) in ABC algorithm, is equivalent to the number of solutions in the population. Furthermore, the position of a food source signifies the position of a promising solution to the optimization problem, whereas the quality of nectar of a food source represents the fitness cost (quality) of the associated solution.



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

II. RELATED WORK

The fully informed particle swarm optimization algorithm (FIPS) developed by Marco A. Montes de Oca and Thomas Stutzle in 2008 [11] is very sensitive to changes in the population topology. The velocity update rule used in FIPS considers all the neighbors of a particle to update its velocity instead of just the best one as it is done in most variants. It has been argued that this rule induces a random behavior of the particle swarm when a fully connected topology is used. This argument could explain the often observed poor performance of the algorithm under that circumstance. But it is found to be more suitable on small search regions.

Many variants of the original particle swarm optimization (PSO) algorithm have been proposed. In many cases, the difference between two variants can be seen as an algorithmic component being present in one variant but not in the other. Marco A. Montes de Oca, Thomas Stützle, Mauro Birattari and Marco Dorigo in 2009 [12] proposes new PSO, where first they presented the results and insights obtained from a detailed empirical study of several PSO variants from a component difference point of view. In the second part, proposed a new PSO algorithm that combines a number of algorithmic components that showed distinct advantages in the experimental study concerning optimization speed and reliability and call this composite algorithm Frankenstein's PSO. Frankenstein's PSO is composed of three main algorithmic components, namely, 1) a timevarying population topology that reduces its connectivity over time, 2) the FIPS mechanism for updating a particle's velocity, and 3) a decreasing inertia weight. These components are taken from AHPSO, FIPS, and the time-decreasing inertia weight variant, respectively. The first component is included as a mechanism for improving the tradeoff between speed and quality associated with topologies of different connectivity degrees. The second component is used because the analysis showed that FIPS is the only algorithm that can outperform the others using topologies of different connectivity degree. Finally, the decreasing inertia weight component is included as a mean to balance the exploration-exploitation behaviour of the algorithm.

Particle swarm optimization (PSO) is known to suffer from stagnation once particles have prematurely converged to any particular region of the search space. George I. Evers and Mounir Ben Ghalia in 2009 [13] proposed regrouping PSO (RegPSO) which avoids the stagnation problem by automatically triggering swarm regrouping when premature convergence is detected. This mechanism liberates particles from sub-optimal solutions and enables continued progress toward the true global minimum. Particles are regrouped within a range on each dimension proportional to the degree of uncertainty implied by the maximum deviation of any particle from the globally best position. Upon detection of premature convergence, the range in which particles are to be regrouped about the global best is calculated per dimension as the minimum of (i) the original range of the search space on dimension j and (ii) the product of the regrouping factor with the maximum distance along dimension j of any particle from global best.

[Zou et al. (2011)] proposed an improved algorithm based on Artificial Bee Colony to deal with multi-objective optimization problems. ABC algorithm based on the intelligent foraging behavior of honey bee swarm used less control parameters and it was found to be very efficient in solving multimodal and multidimensional optimization problems. Proposed algorithm used the concept of Pareto dominance to determine the flight direction of a bee and it maintained non dominated solution vectors which were found in an external archive. Performance of the proposed algorithm indicated that the proposed method can be considered as a viable alternative to solve multi-objective optimization problems.

[Babayigit Bilal et al.(2012)] has presented a modified ABC algorithm (ABC clobset) for numerical optimization problems to improve the exploitation capability of ABC algorithm. This method proposed a different probability function and also a new searching mechanism. The results indicated that the performance the proposed method was much better compared to basic ABC algorithm.

III. PARTICLE SWARM OPTIMIZATION

The PSO was first designed to simulate birds seeking food which is defined as a "cornfield vector." The bird would find food through social cooperation with other birds around it (within its neighbourhood). It was then expanded to multidimensional search. Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Such methods



International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

are commonly known as metaheuristics as they make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. PSO does not use the gradient of the problem being optimized, which means PSO does not require for the optimization problem to be differentiable as is required by classic optimization methods such as gradient descent and quasi-Newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc [8].

PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae. The movements of the particles are guided by the best found positions in the search-space which are updated as better positions are found by the particles.

PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best known position in the search-space as well as the entire swarm's best known position. When improved positions are being discovered these will then it will guide the movements of the swarm. The process is repeated and satisfactory solution will be discovered.

PSO Variants: Various variants of a basic PSO algorithm are possible. New and some more sophisticated PSO variants are continually being introduced in an attempt to improve optimization performance. There is a trend in that research one can make a hybrid optimization method using PSO combined with other optimization techniques[9].

- Discrete PSO
- Constriction Coefficient
- Bare-bones PSO
- Fully informed PSO.

Applications: The first practical application of PSO was in the field of neural network training and was reported together with the algorithm itself (Kennedy and Eberhart 1995). Many more areas of application have been explored ever since, including telecommunications, control, data mining, design, combinatorial optimization, power systems, signal processing, and many others. PSO algorithms have been developed to solve:

- Constrained optimization problems
- Min-max problems
- Multi objective optimization problems
- Dynamic tracking.

Algorithm: The PSO algorithm is simple in concept, easy to implement and computationally efficient.

Main steps of the procedure are:

- 1: Initialize Population
- 2: **repeat**
- 3: Calculate fitness values of particles
- 4: Modify the best particles in the swarm
- 5: Choose the best particle
- 6: Calculate the velocities of particles
- 7: Update the particle positions
- 8: **until** requirements are met.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

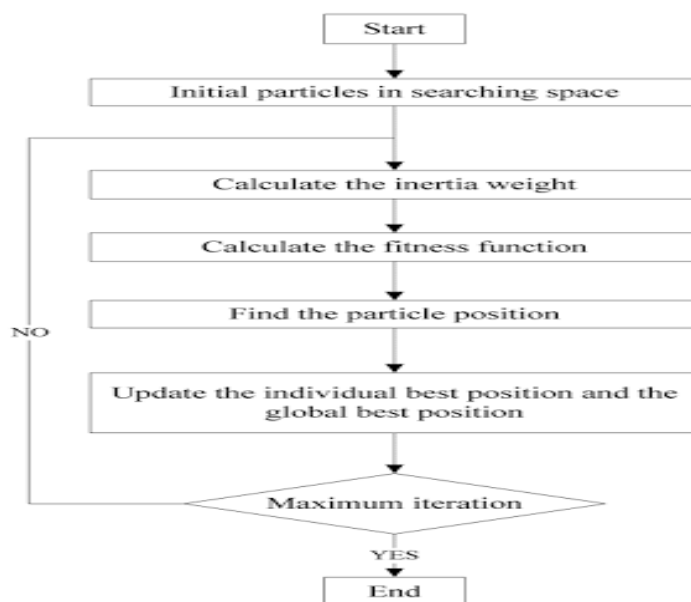


Figure 1. Flowchart of the PSO Algorithm

IV. ARTIFICIAL BEE COLONY

The Artificial Bee Colony (ABC) algorithm is a population-based numeric optimization. It is based on the simplified mathematical models of the food searching behaviours of the bee-swarms. In the ABC algorithm, any random solution of the problem corresponds to a *source of nectar*. There is one *employed bee* assigned to each nectar source. The number of the employed bees equals to the total number of food sources (i.e. the *size of population* value). The employed bee of a nectar source that has run out of nectar turns into a scout bee again. The amount of nectar in a nectar source is expressed with the objective function value of the related nectar source. Therefore, the ABC algorithm targets to locate the nectar source that has the maximum amount of nectar. Following the generation of initial nectar resources, the ABC algorithm starts to search for the solution of the numeric optimization problem using the *employedBee*, *onlooker bee*, and *scout-bee* tools. The employed bee tries to develop the nectar source to which it is assigned using the other nectar sources as well. If the employed bee finds a better nectar source, it memorizes the new nectar source to use it instead of the old one.

The search cycle of ABC consists of three rules: (i) sending the employed bees to a food source and evaluating the nectar quality; (ii) onlookers choosing the food sources after obtaining information from employed bees and calculating the nectar quality; (iii) determining the scout bees and sending them onto possible food sources. The positions of the food sources are randomly selected by the bees at the initialization stage and their nectar qualities are measured. The employed bees then share the nectar information of the sources with the bees waiting at the dance area within the hive. After sharing this information, every employed bee returns to the food source visited during the previous cycle, since the position of the food source had been memorized and then selects another food source using its visual information in the neighbourhood of the present one. At the last stage, an onlooker uses the information obtained from the employed bees at the dance area to select a food source. The probability for the food sources to be selected increases with increase in its nectar quality. Therefore, the employed bee with information of a food source with the highest nectar quality recruits the onlookers to that source. It subsequently chooses another food source in the neighbourhood of the one currently in her memory based on visual information (i.e. comparison of food source positions). A new food source is randomly generated by a scout bee to replace the one abandoned by the onlooker bees.

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

Applications of ABC:

Large numbers of realworld optimization problems have been solved by the ABC algorithm that demonstrates the utilization and effectiveness of this algorithm. In the following subsection, some areas to which ABC has applied are discussed in detail.

These areas include:

- Benchmark optimization
- Bioinformatics field
- Data Mining
- Engineering design and applications
- Scheduling

Algorithm:

The main steps of the algorithm are as below:

- 1: Initialize Population
- 2: **repeat**
- 3: Place the employed bees on their food sources
- 4: Place the onlooker bees on the food sources depending on their nectar amounts
- 5: Send the scouts to the search area for discovering new food sources
- 6: Memorize the best food source found so far
- 7: **until** requirements are met

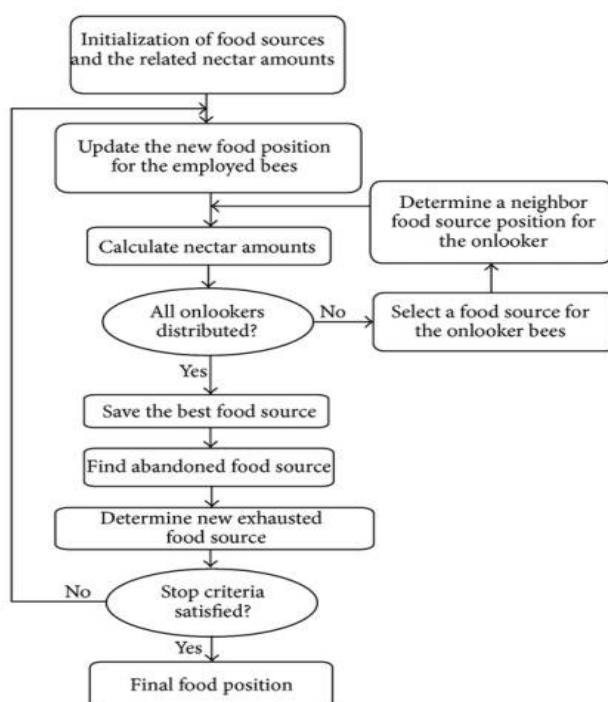


Figure 2: Flowchart of ABC algorithm

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

V. EXPERIMENTS

We have implemented the basic structure of Particle Swarm Optimization (PSO) and Artificial bee colony (ABC) [12] using MATLAB. The program was run on a PC platform in MATLAB 7.0 version. The circle function is used as our example with a population size of 50, with 300 generations and over 100 runs. In CS, value of ap was 0.25. For PSO the value of 'w', the inertia weight was .4 and the values of both confidence factors were taken as 2. Same data was collected from both ABC and PSO program. It is observed that the X-Diff values generated by PSO have started with a high value and quickly fell down to low values until most of the particles reach the same point. On the other hand the CS started with little bit low value than PSO but traversed more area and converged to the lowest point

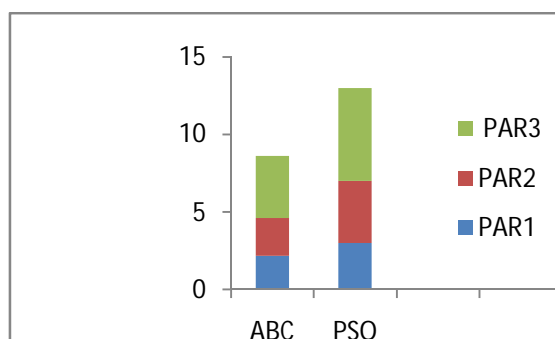


Figure 3. The difference in X values for the circle function for ABC and PSO

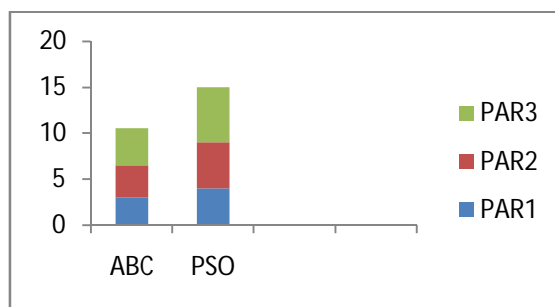


Figure 4. The difference in Y values for the circle function for ABC and PSO.

VI. CONCLUSION

The objective of this comparative study is to test the hypothesis that states, although PSO and ABC on average yield the same effectiveness (solution quality), PSO is more computationally efficient (uses less number of parameters) than the ABC. It resulted in equal effectiveness but superior efficiency for PSO over the ABC. In future, an emphasis can be placed on new type of hybridization between these two. Also more standard benchmarking test functions can be utilized to make more comparisons and they can be extended to solve various multi objective optimization problems as well.

REFERENCES

- [1] M. Dorigo, M. Birattari, and T. Stutzle. Ant colony optimization. *Computational Intelligence Magazine*, IEEE, 1(4):28–39, 2006.
- [2] P.J. Angeline, G.M. Saunders, and J.B. Pollack. An evolutionary algorithm that constructs recurrent neural networks. *Neural Networks*, IEEE Transactions on, 5(1):54-65, 1994.
- [3] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Neural Networks, 1995. Proceedings., IEEE International Conference on*, volume 4, pages 1942–1948. IEEE, 1995.



ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 5, May 2015

- [4] Z.W. Geem, J.H. Kim, and GV Loganathan. A new heuristic optimization algorithm: harmony search. *Simulation*, 76(2):60, 2001.
- [5] D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.
- [6] Kennedy, J., Eberhart, R.: Particle Swarm optimization. *Engineering and Technology*.1942–1948 (1995).
- [7] A. Singh. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem.*Applied Soft Computing*, 9(2):625–631, 2009.
- [8] Trelea, I.C. The particle swarm optimization Algorithm_: Convergence analysis and parameter selection. 85, 317–325 (2003).
- [9] R. Poli. Analysis of the publications on the applications of particle swarm optimization.*Journal of Artificial Evolution and Applications*, Article ID 685175, 10 pages, 2008.
- [10] D. Zhang, X. Guan, Y. Tang, and Y. Tang. Modified artificial bee colony algorithms for numerical optimization. In *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on*, pages 1-4. IEEE, 2011.
- [11] Marco A. Montes de Oca and Thomas Stutzle, “Convergence Behavior of the Fully Informed Particle Swarm Optimization Algorithm”, 2008.
- [12] Marco A. Montes de Oca, Thomas Stützle, Mauro Birattari and Marco Dorigo, “Frankenstein’s PSO: A Composite Particle Swarm Optimization Algorithm”, IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 13, NO. 5, OCTOBER 2009.
- [13] George I. Evers and Mounir Ben Ghalia, “Regrouping Particle Swarm Optimization: A New Global Optimization Algorithm with Improved Performance Consistency Across Benchmarks”, 2009.
- [14] Zou W, Zhu Y, Chen H, Shen H (2011), A Novel Multi-Objective Optimization Algorithm Based on Artificial Bee Colony, GECCO’11, Dublin, Ireland, ACM.
- [15] Babyigit Bilal, ResulOdzemir (2012) A modified artificial bee colony algorithm for numerical function optimization,IEEE