International Journal of Business and Technology

Volume 2
Issue 2 *Spring 2014*

Article 1

May 2014

# Swarm Intelligence as an Optimization Technique

Alma Bregaj
*Ismail Qemali University*

Follow this and additional works at: https://knowledgecenter.ubt-uni.net/ijbte

Part of the Computer Sciences Commons, and the Engineering Commons

Recommended Citation
Bregaj, Alma (2014) "Swarm Intelligence as an Optimization Technique," *International Journal of Business and Technology*: Vol. 2 : Iss. 2 , Article 1.
DOI: 10.33107/ijbte.2014.2.2.01
Available at: https://knowledgecenter.ubt-uni.net/ijbte/vol2/iss2/1

This Article is brought to you for free and open access by the Publication and Journals at UBT Knowledge Center. It has been accepted for inclusion in International Journal of Business and Technology by an authorized editor of UBT Knowledge Center. For more information, please contact knowledge.center@ubt-uni.net.

# Swarm Intelligence as an Optimization Technique

Alma Bregaj

Departament of Computer Sciences, University "I.Qemali" of Vlora

**Abstract.** Optimization techniques inspired by swarm intelligence have become increasingly popular during the last years. Swarm intelligence is based on nature-inspired behaviours and is successfully applied to optimisation problems in a variety of fields. The advantage of these approaches over traditional techniques is their robustness and flexibility. These properties make swarm intelligence a successful design paradigm for algorithms that deal with increasingly complex problems. In this paper I am focused on the comparison between different swarm-based optimisation algorithms and I have presented some examples of real practical applications of these algorithms.
**Keywords**: Optimization, swarm Intelligence, Ant Colony Optimization, Particle Swarm Optimization, Artificial Bees Colony.

## 1 Introduction

The complex and often coordinated behavior of swarms fascinates not only biologists but also computer scientists. Bird flocking and fish schooling are impressive examples of coordinated behaviors that emerges without central control. Social insect colonies show complex problem-solving skills arising from the actions and interactions of nonsophisticated individuals. Swarm intelligence (SI) systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The agents follow very simple rules, and although there is no centralized control structure dictating how individual agents should behave, local interactions between such agents lead to the emergence of a complex global behavior.
In this paper, I have made a short analysis of the most successful methods of optimization techniques inspired by Swarm Intelligence: *Ant Colony Optimization(ACO)* which is based on the philosophy of ant colony, *Particle Swarm Optimization(PSO)* and *Artificial Bee Colony(ABC)*.

## 2. Swarm Intelligence Algorithms

The most known swarm intelligence algorithms and a short description for each of them is presented in the following lines.

### 2.1 Ant Colony Optimization (ACO)

Ant colony optimization (ACO) was one of the first techniques for approximate optimization inspired by swarm intelligence. More specifically, ACO is inspired by the foraging behavior of ant colonies. The basic principle is based on finding the shortest path from food source to anthill by smelling pheromones (chemical substances they leave on the ground during walk).
The system of obtaining food in an ant colony is managed by hundreds of individuals and covered thousands of square meters. In process of collecting food if there are two possible paths to reach a food source, as shown in Fig. 1, and they have no clue about which direction to choose, they choose it randomly. It is assumed that half of them choose the first direction and the rest choose the other one. Suggesting that all ants have same walking speed, the shorter way will receive a greater amount of pheromone per time. Next time when they will choose the shortest way by smelling more pheromone on the shorter path than the longer one. Other ants make use of pheromone concentration to determination of the shortest way, which give them the possibility to collect food quicker.

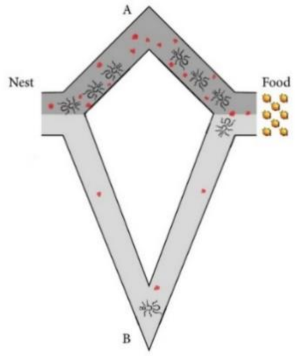


**Fig. 1.** The pheromone deposition of ants(red dots).

Understanding a natural phenomena and the design of nature-inspired algorithm are related to each other, but are two things completely different. Understanding a natural phenomenon is constrained by observations and experiments, while designing a nature-inspired algorithm is only limited by one's imagination and available technology. Figure 2 illustrates the framework that is generally used to move from a natural phenomenon to a nature-inspired algorithm.


**Fig. 2**. An illustration to the general framework that represents the transition from a natural phenomenon to a nature-inspired algorithm.

The main idea is to model the problem to be solved as a search for an optimal path in a weighted graph, and to use artificial ants to search for quality paths. In the ACO algorithm the main task for each ant is to find the shortest path between a source node and a destination node. ACO metaheuristic can be characterized by the following:

1. *A probabilistic transition rule is used to determine the moving direction of each ant.*
2. *Pheromone update mechanism indicates the problem solution quality*.

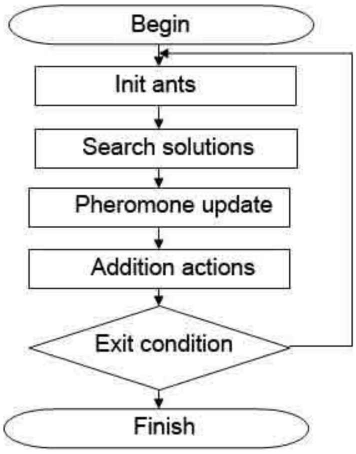


**Fig. 3.** Basic steps off ACO

### 2.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique modelled on the social behaviors observed in animals or insects, e.g., bird
flocking, fish schooling, and animal herding. It was originally proposed by James Kennedy and Russell Eberhart in 1995.

In these groups, there is a leader (individual with the best value of fitness) who guides the movement of the whole swarm. The movement an individual is based on the leader and on its own knowledge. Generally could by said that the model PSO presuppose that the behaviour of each individual is a compromise between its own and collective knowledge.

The PSO algorithm consists of just three steps, which are repeated until some stopping condition is met:

1. Evaluate the fitness of each particle




DOI: 10.33107/ijbte.2014.2.2.01

2. Update individual and global best fitnesses and positions
3. Update velocity and position of each particle

The first two steps are fairly trivial. Fitness evaluation is conducted by supplying the candidate solution to the objective function. Individual and global best fitnesses and positions are updated by comparing the newly evaluated fitnesses against the previous individual and global best fitnesses, and replacing the best fitnesses and positions as necessary.

PSO algorithms have been applied to optimization problems ranging from classical problems such as scheduling, the traveling salesman problem, neural network training, task assignment, to highly specialized applications such as reactive power and voltage control, biomedical image registration, and even music composition.

### 2.3 Artificial Bees Colony(ABC)

The bees algorithm is a population-based search algorithm inspired by the natural foraging behaviour of honey bees. In its basic version, the algorithm starts by scout bees being placed randomly in the search space. Then the fitnesses of the sites visited by the scout bees are evaluated and Bees that have the highest fitnesses are chosen as "selected bees" and sites visited by them are chosen for neighbourhood search. Then, the algorithm conducts searches in the neighbourhood of the selected sites, assigning more bees to search near to the best e sites. Searches in the neighbourhood of the best e sites are made more detailed by recruiting more bees to follow them than the other selected bees. Together with scouting, this differential recruitment is a key operation of the Bees Algorithm. The remaining bees in the population are assigned randomly around the search space scouting for new potential solutions. These steps are repeated until a stopping criterion is met. At the end of each iteration, the colony will have two parts, those that were the fittest representatives from a patch and those that have been sent out randomly. the algorithm performs a kind of neighbourhood search combined with random search and can be used for both combinatorial and functional optimisation.

The pseudo code of basic bees algorithm is:

1) Initialise population with random solutions.
2) Evaluate the fitness for all candidates.
3) While (stopping criterion not met): //Forming new population.
    a) Select sites for circumambience search.
    b) Send out bees to selected sites (more bees for better sites) and evaluate its fitness.
    c) Select bees with the highest fitness from each patch.
    d) Assign remaining bees to search randomly and evaluate their fitness.
4) End While

Termination of the algorithm is reaching the satisfactory solution or after given number of repetitions.

## 3 Comparision of swarm-based optimization algorithms

Optimization algorithms above were compared by eight following benchmark functions with hundred independent measurements.

**Table 1.** Comparision between Genetic Algorithms(GA), Ant Colony Optimization and Bees Algorithm.

| Function | GA | | ACO | | Bees Algorithm | |
|---|---|---|---|---|---|---|
| | Number of iterations | Success(%) | Number of iterations | Success(%) | Number of iterations | Success(%) |
| **Griewang** | 200000 | 100 | 50000 | 100 | 1847 | 100 |
| **Hypersphere** | 15468 | 100 | 22050 | 100 | 999 | 100 |
| **Rosenblock** | 10212 | 100 | 6842 | 100 | 1657 | 100 |
| **Martin&Gaddy** | 2844 | 100 | 1688 | 100 | 526 | 100 |
| **Branim** | 7325 | 100 | 1936 | 100 | 898 | 100 |
| **Goldstein&Price** | 5662 | 100 | 5330 | 100 | 7113 | 100 |
| **De Jong** | 10160 | 100 | 6000 | 100 | 1847 | 100 |

The last function De Jong's figured out, that the Bees Algorithm reached the optimum 207 faster than GA and 120 times faster than ACO, with a success of 100%. For the Goldstein & Price function, the Bees Algorithm could find the optimum almost 5 times faster than GA and ACO, again with 100% success. In Branin's function there was for Bees Algorithm a 15% improvement compared with ACO and 77% improvement compared with GA, also with 100% success. Rosenbrock's function in two-dimensions has with the Bees Algorithm at least twice fewer evaluations than the other methods also with 100% success. Four-dimensions Rosenbrock's function, where ACO could reach the optimum 3,5 times faster than the Bees Algorithm with success rate 100%. In Hyper Sphere model of six dimensions, the Bees Algorithm needed half of function evaluations compared with GA and one third compared with ACO. Last but not least Griewangk function is ten-dimensional and the Bees Algorithm found the optimum with 100% success and 10 times faster than GA and 25 times faster than ACO.

## 4 Conclusions

Experiments with different test cases on these algorithms show that the Bees Algorithm is more efficient because it gives the results in the biggest number of function in the shortest time with 100% success rate in all cases. But a fact is that the Bees Algorithm is less adaptive than ACO. The Bees Algorithm is used in training neural networks for pattern recognition, forming manufacturing cells, fcheduling jobs for a production machine, data clustering etc.

Some other applications of swarm intelligence are: communication networks, robotics, scheduling problems, graph theory etc. Swarm technology is particularly attractive because it is cheap, robust, and simple. Other important advantages of swarm intelligence are: adaptability, scalability, collective, robustness, and individual simplicity of agents**.**

Because of the lack of central coordination, SI systems could suffer from a stagnation situation (e.g., in ACO, stagnation occurs when all the ants eventually follow the same suboptimal path and construct the same tour). Another disavantage of SI systems is parameter tuning since many parameters of SI systems are problem-dependent.

## References

1. H. Ahmed, and J. Glasgow, “Swarm Intelligence: Concepts, Models and Applications,”, University of Queen, Kingston, Ontario, Canada K7L3N6, 2012.
2. M.Negnevitsky, “Artificial Intelligence, A Guide to Intelligence Systems”, Second Edition, 2005.
3. M.Mitchel, “An introduction to Genetic Algorithms”, 1998.
4. K. Panigrahi, Y. Shi, and M.H. Lim, “Handbook of Swarm Intelligence” , vol 7, ISBN 978-3-642-17389-9.
5. S. Das, B. K. Panigrahi, S. S. Pattnaik, “Nature-Inspired Algorithms for Multi-objective Optimization, Handbook of Research on Machine Learning Applications and Trends: Algorithms Methods and Techniques” , Hershey, New York, 2009.
6. V.S. Hernández and A. Weitzenfeld, “Ant Colony Algorithm for Swarm Systems”, IEEE, 2006.
7. M. Dorigo, E. Bonabeau, and G. Theraulaz, “Ant algorithms and stigmergy, Future Gener. Comput. Systems”, 2000.
8. Andrés, S. Lozanob. A particle swarm optimization algorithm for part–machine grouping. 2005. Robotics and Computer-Integrated Manufacturing 22 (2006) 468–474.
9. Swarm Intelligence in Optimization, Christian Blum1 and Xiaodong Li2
10. A Bee Algorithm for Multi-Agents System-Lemmens ,Steven . Karl Tuyls, Ann Nowe -2007.
11. Analysis of Particle Swarm Optimization Algorithm, Qinghai Bai, Computer Information Science, Vol. 3, No.1, Febrary 2010.
12. Bonabeau, E.—Dorigo, M.—Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford 1999.
13. Swarm Intelligence – Literature Overview, Yang Liu , Kevin M. Passino. 2000.
14. Swarm-Based Optimisation, Author: Lenka Raudenská, Quality Innovation Prosperity Xiii/1 – 2009.



https://knowledgecenter.ubt-uni.net/ijbte/vol2/iss2/1

DOI: 10.33107/ijbte.2014.2.2.01