Analysis of Artificial Bee Colony Optimization and Firefly algorithms

Tyler Loewen
University of Manitoba
Manitoba
Canada
loewent4@myumanitoba.ca

Tyler Loewen – loewent4@myumanitoba.ca

Abstract

Particle swarm optimization (PSO) algorithms are frequently used to solve problems in a large variety of domains from engineering, networking, to distributed systems. PSO algorithms display swarm intelligence behaviours that closely resemble groups of animals in nature such as bee colonies, fireflies, ants, and flocks of birds. In recent years, the behaviour of bee colonies and fireflies has been used to develop novel adaptations of the traditional PSO algorithms. This paper introduces the artificial bee swarm optimization (ABSO) algorithm and the basic firefly algorithm (FA) and describes their advantages and applications.

1. Introduction

Swarm intelligence algorithms have been used to solve problems in traffic routing, networking, games, robotics, and economics [1]. These algorithms are modelled after swarm-like behaviour found in schools of fish, ants, and flocks of birds, among other groups of animals. A swarm algorithm is made up of multiple individual agents that behave independently of one another, without supervision, and on a local scale, based on their surrounding environment and the interactions between other agents. This reflects the concept of a distributed system where individuals communicate and work together to complete a common goal. Collectively, these interactions lead the agents to the creation of collective intelligence and a global solution. Individual agents are not aware of the global solution but through local interactions with other a global solution can be reached.

The foundation of bee swarm and firefly algorithms is particle swarm optimizations (PSO) algorithms. Exploration and exploitation are two factors of PSO algorithms that determine the success of the algorithm [2]. The degree of exploration determines the algorithm's ability to generate new solutions in a region of unexplored search space [2]. Exploitation is "..the concentration of the algorithm's search at the vicinity of current good solutions" [2]. The balance of these two factors determines the success of the algorithm. Having too low exploration prevents the algorithm from branching out to different potential solutions and often results in premature convergence resulting in getting stuck in local optima when global optima exist [1]–[3]. Having too low exploitation reduces the algorithm's ability to converge on a potential optimal solution. The two factors are summarized by "The exploitation process applies the existing knowledge to seek better solutions, whereas the exploration process is concerned with the entire search of the space for an optimal solution.” [3]. Many PSO algorithm variations have been created that address the balancing of these two factors for applications in different domains.

In this paper, the implementation of the artificial bee swarm optimization (ABSO) algorithm, based on the behaviour of bees, is described in Section 3. Next, Section 4 discusses the firefly algorithm (FA) which is based on the behavioural patterns of fireflies in nature. Lastly, the performance and applications of the ABSO algorithm and FA are discussed in Section 5.

3. Artificial Bee Swarm Optimization Algorithm (ABSO)

3.1 Natural Behaviour of Bees

Bees are social insects that live and work together to make up an individual colony. There are three types of bees: the queen bee, drone bee, and worker bee. Each type of bee takes on a different behavioural role that contributes to the overall success of the colony.

Queen Bee

There is only one queen bee per colony. The queen bee is responsible for producing the offspring of the colony by laying thousands of unfertilized eggs over its lifetime. The queen bee’s three-to-four-year long life begins when it hatches as an unmated queen in a colony of potentially multiple other unmated queens. The unmated queens will fight each other until only one remains. If the bees in the colony do not accept this queen, they will kill her and attempt to make another queen. Although the queen bee has a stinger, it is only used for fighting rivalling queens. Once mated, the queen bee remains in the colony and continues to lay eggs, rarely leaving the hive. The original queen bee is eventually replaced by a new queen when its ability to lay eggs decreases.

Drone Bee

Bee colonies usually consist of a few hundred drone bees depending on the time of year [1]. Drone bees originating from unfertilized eggs which makes them the only male bees in a colony. They can live for up to six months [1]. Their main purpose is to mate with their queen bee, so she becomes fertile and can lay eggs. Drone bees remain inside the colony and only leave when mating. Their abdomen does not contain a stinger, unlike the queen and worker bees.

Worker Bee

Worker bees are females originating from fertilized eggs that are fed differently than queen or drone bees. They are the most populous bees in a colony. The average worker bee lives for six weeks and performs the majority of the work in the colony. Some of this work includes foraging, defending the colony, removing dead bees and debris from the hive, and controlling the temperature and humidity inside the hive through fanning. They have a barbed stinger in their abdomen that remains embedded in the target after stinging, killing the worker bee in the process.

Foraging

Foraging is a crucial aspect of the survival of a bee colony. Worker bees explore areas surrounding the colony looking for adequate supplies of resources such as pollen, nectar, and water. Once a source of nectar is found, worker bees collect it by storing it in their honey stomach for transportation back to the colony [1]. An enzyme is then released in their stomach which begins turning the nectar into honey [1]. Worker bees can forage for resources in up to a 5km radius around their hive. When the worker bees return from foraging, they empty the contents of their stomach into honeycomb cells for storage and later use. They then perform dances to communicate the information about the resource to other worker bees in the colony.

Communication

Worker bees communicate information about resources they find such as the distance, direction, and abundance of the resource to the rest of the colony through different types of dances [1]. Three different types of dances are performed depending on the distance to the resource [1]. The round dance is performed if the resource is within 100 meters of the colony [1]. No direction is communicated in the round dance [1]. The waggle dance is used for longer distances and communicates the direction of the resource [1]. The tremble dance is used when the worker bee sees a longer than the normal time required to unload its nectar into cells [1].

3.2 Implementation

The artificial bee swarm optimization (ABSO) algorithm described in [2] is a modification of the basic bee algorithm (BA) which identifies optimal parameters of solar cell models by minimizing a function given a range of parameters. The given parameters are optimized until a terminating condition is met [2]. BA variants of PSO algorithms such as the ABSO algorithm use the behaviour of bees to balance the factors of exploration and exploitation. ABSO uses two types of entities that divide labour between each other allowing for two types of tasks to be completed simultaneously [1]. This is opposed to conventional PSO algorithms that only contain a single type of entity in the swarm. The two types of entities are bees labelled as onlooker and scout bees. The scout bees are responsible for branching out and scouting for new resources which affect the exploration factor of the algorithm [2]. Onlooker bees work to improve the exploitation of the algorithm by updating and communicating the location of the currently highest quality resource to other bees [2].

The ABSO algorithm mimics the behaviour of bees by representing potential solutions to a problem as food sources that the bees forage for. Each bee has an objective function that determines the quality of the food source [2]. Before distinguishing between scout or onlooker bees, the bees leave the hive first to forage for food sources. They then return to the hive to report the quality of the food sources they have found. The bees are then divided into scout and onlooker groups depending on the quality of food sources they have found during their initial foraging (i.e., the results of their objective function) [2]. Once ranked, a fixed number of bees who found the worst quality food sources are designated as scout bees. The remaining bees are designated as onlooker bees with a portion of them designated as elite onlooker bees.

Onlooker bees are required to remember the location and quality of the food sources they discover [2]. The elite onlooker bees are responsible for performing dances to communicate to other onlooker bees where the best food sources are currently located, just as biological bees do [1], [2]. Notably, as the quality of currently known food sources increases, more bees fly to those food sources creating a swarm-like behaviour. During an iteration of the algorithm, each onlooker bee chooses an elite onlooker bee to listen to through a tournament selection process [2]. In this process, elite onlooker bees that are communicating information about a lower quality food source have a lower chance of being chosen than elite onlooker bees that are communicating a higher quality food source [2]. This causes a gradual convergence to the higher quality food sources over multiple iterations of the algorithm which supports the exploitation factor of PSO algorithms. After getting information from a chosen elite onlooker bee, each onlooker bee combines this information with its current information about the food to determine a new position. In this way, the overall trending location of the best food sources decided by the hive collectively is used to affect the current position of the bees, creating a hivemind behaviour.

The scout bees are tasked to randomly fly over the location of the food sources in search of new food sources [2]. This behaviour impacts the exploration factor of the algorithm by helping the colony diversify and branch out to find new food sources, helping to decrease the probability of becoming stuck at local minima. The movement of the scout bees is defined by a function that among other things, controls the distance the bee can move in a single iteration of the algorithm [2]. In each iteration, the distance a scout bee can move decreases linearly [2]. This helps the scout bees reach a larger radius (search space) early in the algorithm to help discover more potential solutions while also aiding a local search in later iterations by converging through reducing the distance travelled. The steps of the ABSO algorithm are as follows:

1. Randomly initialize a swarm of bees in the search space uniformly along with a vector for each solution.
2. Compute the value of the objective function for each bee
3. Rank the bees based on the results of their objective function
4. Divide and specify the onlooker and scout bees
5. Update position of the scout and onlooker bees according to their movement patterns
6. Replace a bee's position with its current position if it exceeds the search space
7. Repeat steps 2 to 6 until the maximum iterations are reached
8. The best achievement of the swarm is selected as the optimal solution

By using an initial search of food sources to determine their location and quality, onlooker bees can begin communicating information about the food sources to other bees in the swarm. The bees are then able to use the information about food sources from the onlooker bees to guide the search for new and better food sources. Once the algorithm terminates the current best food source is accepted as the global minimal solution.

4. Firefly Algorithm (FA)

4.1 Natural Behaviour of Fireflies

Fireflies are insects with more than 2,000 species in the order beetles [4], [5]. They have bioluminescence capabilities to produce light using light-emitting organs called lanterns located on the beetle's lower abdomen [5]. The lights serve two main purposes: to identify mating partners of the same species and to attract prey [4], [6], [7]. Some diurnal species cannot produce light and instead use pheromones to achieve these functions. Information is communicated between fireflies using the colour, timing, duration, and repetition of the emitted light [4], [6], [7]. By varying these properties a firefly can produce a distinct flashing signal that encodes its species and sex [5]. In most cases, the male firefly will produce light signals during flight to attract females that are located on the ground. Once the females receive the signal, they respond accordingly with their light response [5]. The strength of the light from a firefly determines how attracted other fireflies are to the light [8].

4.2 Implementation

The original firefly algorithm (FA) is an optimization algorithm created in 2008 by Xin-She Yan [7]. FA uses the flashing bioluminescence behaviour of fireflies to discover an optimal solution through particle swarm intelligence. The object function for FA is associated with the light that fireflies emit [7]. Each firefly represents a randomized solution with an intensity relative to its performance on the objective function [6]. Three rules are created to adapt the natural behaviour of fireflies to work with the algorithm:

- All fireflies are unisex so that anyone firefly will be attracted to any other firefly regardless of sex
- The attractiveness of fireflies is proportional to their light intensity, thus the less bright one will move towards the brighter one. If no firefly is brighter, it moves randomly
- The brightness of a firefly is affected or determined by the landscape of the objective function

Two main components affect the behaviour of the swarm: the intensity of emitted light and the attractiveness of the firefly [7]. The intensity of a firefly is the absolute measure of light emitted by the firefly. The attractiveness is the intensity of the light as observed by other fireflies [5]. The attractiveness of a firefly depends on two factors. First, the distance from the observing firefly to the firefly emitting light. This distance follows the inverse square law which states the diminishing of light over distance can is represented by $I = 1/d^2$ where I is the intensity and d is the Euclidean distance to the light source. Second, the absorption of light by air, which is determined by some fixed light absorption coefficient [4]. A function to determine the attractiveness of a firefly is created by combining the inverse square law formula with the light absorption coefficient [7].

Each firefly's movement is determined by the level of attractiveness to surrounding fireflies. A firefly tends to move from its current position towards the most attractive. The movement function consists of three terms: the current position of the firefly, the attraction to another firefly, and a random walk value [5]. When a firefly is not attracted to any other firefly the attraction term is 0 meaning movement only depends on the random walk term [5]. To update the position of the firefly the attraction term, the random walk term and the current position are summed. The steps of FA are summarized as follows [7]:

1. The objective function, initial population of fireflies, light intensity of each firefly, and the absorption coefficient are initialized
2. For each firefly, compare the intensity of every other firefly and move towards the firefly with a stronger intensity if one exists. Otherwise move randomly.
    a. Vary attractiveness with the distance between the two fireflies
    b. Evaluate new solutions and update light intensity
3. Rank the fireflies by intensity and find the current global best solution

FA is controlled by three parameters: the randomization parameter that is part of the random walk term, the attractiveness function, and the light absorption coefficient [5]. The light absorption coefficient controls the speed at which the algorithm converges on a potential solution and is customized to better fit the specific application of FA. When the absorption coefficient is 0, the attraction function becomes constant anywhere in the search space [5]. This is equivalent to every firefly being able to see every other firefly light from any distance, causing the fireflies to move directly to the current optimal solution [7]. When the absorption coefficient approaches ∞ the random walk term takes over causing the firefly to move in a random direction [5]. This is equivalent to fireflies emitting light in a heavy fog where the lights are barely visible, causing the fireflies to move randomly [7]. By controlling these three parameters FA can be adjusted to work with many different types of optimization problems.

5. Results and Applications

Improvements and variations of the basic BA have been used to solve problems in network routing, pattern formation on a grid, software fault tolerance, and load balancing of tasks in cloud computing environments, among other problems [1], [9]. The ABSO variation of BA performs better than pattern search, simulated annealing, and harmony search algorithms in identifying optimal parameters of the solar cell models [2].

The basic FA is well suited for solving NP-hard problems and multi-modal optimization applications [4], [5]. It has been applied to many different classes of optimization problems [5]. Some of these problems include continuous, combinatorial, constrained, multi-objective, dynamic, and noisy optimization problems as well as classification problems in machine learning, data mining, and neural networks [5]. In continuous-optimization problems like welded beam design and pressure vessel design, FA is more efficient than particle swarm optimization, genetic algorithms, simulated annealing and differential evolution [5]. FA could be improved by gradually reducing the randomization parameter in the random walk term as the algorithm reaches a solution [4], [5]. This would allow the algorithm to better converge on local optima in later iterations without leading to premature convergence in the early iterations of the algorithm.

6. Conclusions

The firefly algorithm (FA) and the artificial bee swarm (ABSO) algorithm provide advantages over traditional particle swarm optimization (PSO) algorithms, such as minimizing getting stuck at a local minimum/maximum due to premature convergence. Both algorithms provide parameters that can be tweaked to make them compatible with a wide variety of problems. These parameters include constants to adjust the balance between exploration and exploitation that affect convergence and exploration in the problem space. PSO algorithms and their variants are powerful tools that are simple to implement compared to other complex algorithms, highly adaptable to a large variety of problem types due to many adjustable parameters, and can discover optimal solutions to NP-hard problems.

References

[1] D. Karaboga and B. Akay, “A survey: Algorithms simulating bee swarm intelligence,” *Artif. Intell. Rev.*, vol. 31, no. 1–4, pp. 61–85, 2009, doi: 10.1007/s10462-009-9127-4.

[2] A. Askarzadeh and A. Rezazadeh, “Artificial bee swarm optimization algorithm for parameters identification of solar cell models,” *Appl. Energy*, vol. 102, pp. 943–949, 2013, doi: 10.1016/j.apenergy.2012.09.052.

[3] Z. Li, W. Wang, Y. Yan, and Z. Li, “PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems,” *Expert Syst. Appl.*, vol. 42, no. 22, pp. 8881–8895, 2015, doi: 10.1016/j.eswa.2015.07.043.

[4] X. S. Yang, “Firefly algorithms for multimodal optimization,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5792 LNCS, pp. 169–178, 2009, doi: 10.1007/978-3-642-04944-6_14.

[5] I. Fister, X. S. Yang, and J. Brest, “A comprehensive review of firefly algorithms,” *Swarm Evol. Comput.*, vol. 13, pp. 34–46, 2013, doi: 10.1016/j.swevo.2013.06.001.

[6] S. Arora and S. Singh, “The Firefly Optimization Algorithm: Convergence Analysis and Parameter Selection,” *Int. J. Comput. Appl.*, vol. 69, no. 3, pp. 48–52, 2013, doi: 10.5120/11826-7528.

[7] X. Yang, “Nature-Inspired Metaheuristic Algorithms,” *Nature-Inspired Metaheuristic Algorithms Second Ed.*, p. 115, 2010.

[8] P. Sivaranjani and A. Senthil Kumar, “Hybrid Particle Swarm Optimization-Firefly algorithm (HPSOFF) for combinatorial optimization of non-slicing VLSI floorplanning,” *J. Intell. Fuzzy Syst.*, vol. 32, no. 1, pp. 661–669, 2017, doi: 10.3233/JIFS-152551.

[9] L. D. Dhinesh Babu and P. Venkata Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” *Appl. Soft Comput. J.*, vol. 13, no. 5, pp. 2292–2303, 2013, doi: 10.1016/j.asoc.2013.01.025.