

An Analysis of Foraging and Echolocation Behavior of Swarm Intelligence Algorithms in Optimization: ACO, BCO and BA

Tanzila Islam, Md Ezharul Islam, Mohammad Raihan Ruhin

Department of Computer Science and Engineering, Jahangirnagar University, Dhaka, Bangladesh

Email: tanzilamohita@gmail.com, mdezharul@juniv.edu, raihanruhin@gmail.com

How to cite this paper: Islam, T., Islam, M.E. and Ruhin, M.R. (2018) An Analysis of Foraging and Echolocation Behavior of Swarm Intelligence Algorithms in Optimization: ACO, BCO and BA. *International Journal of Intelligence Science*, 8, 1-27. <https://doi.org/10.4236/ijis.2018.81001>

Received: December 19, 2017

Accepted: January 28, 2018

Published: January 31, 2018

Copyright © 2018 by authors and

Scientific Research Publishing Inc.

This work is licensed under the Creative

Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Optimization techniques are stimulated by Swarm Intelligence wherever the target is to get a decent competency of a problem. The knowledge of the behavior of animals or insects has a variety of models in Swarm Intelligence. Swarm Intelligence has become a potential technique for evolving many robust optimization problems. Researchers have developed various algorithms by modeling the behaviors of the different swarm of animals or insects. This paper explores three existing meta-heuristic methods named as Ant Colony Optimization (ACO), Bee Colony Optimization (BCO) and Bat Algorithm (BA). Ant Colony Optimization was stimulated by the nature of ants. Bee Colony Optimization was inspired by the plundering behavior of honey bees. Bat Algorithm was emerged on the echolocation characteristics of micro bats. This study analyzes the problem-solving behavior of groups of relatively simple agents wherein local interactions among agents, are either directly or indirectly through the environment. The scope of this paper is to explore the characteristics of swarm intelligence as well as its advantages, limitations and application areas, and subsequently, to explore the behavior of ants, bees and micro bats along with its most popular variants. Furthermore, the behavioral comparison of these three techniques has been analyzed and tried to point out which technique is better for optimization among them in Swarm Intelligence. From this, the paper can help to understand the most appropriate technique for optimization according to their behavior.

Keywords

Optimization, Swarm Intelligence, Colony, Foraging, Echolocation

1. Introduction

A swarm is an extensive number of homogenous. Swarm based algorithms have

as of late risen as a bunch of nature driven, population based algorithm that are equipped for delivering minimal effort, quick, and strong answers for some complex issues. Swarm Intelligence (SI) can be defined as the joined mentality of decentralized or self-sorted out frameworks in regular or simulated. The motivation originates from generally organic framework. Swarm knowledge is a characteristic calculation since it is made by following the development and working conduct of common creatures and creepy crawlies. A flock of birds is an example of a swarm of birds. Bees' swarming around their hive is another example of a swarm whose individual agents are bees. On the off chance that we nearly watch a solitary ant or honey bee, we will comprehend that they are not so keen but rather their settlements are. Swarm knowledge can help people to solve complex frameworks, from truck steering to military robots. A colony can solve any issue, for example, Ant Colony Optimization algorithm is used for finding the shortest path in the network routing problem, and Particle Swarm Intelligence is used in optical network optimization. As an individual, the swarm may be little fakers, yet as colonies they react rapidly and adequately to their environment. There are two types of social interactions among swarm individuals, namely direct interaction and indirect interaction. Direct interactions are the obvious interaction through visual or audio contact, for example, birds interact with each other with sound. Indirect interaction is called Stigmergy [1], where agents interact with the environment. A pheromone trail of ants is an example of indirect interaction.

In SI, the agents form a local solution. The global solution is developed based on the local solution. The global solution then gradually optimizes to an optimal solution. So, Swarm Intelligence is a bottom-up type of problem solving technique. SI has discovered immense applicability in fields like Robotics, Artificial Intelligence, staff scheduling, process optimization, entertainment, telecommunications, software engineering, routing, software testing, networking etc. Ant Colony Optimization (ACO) is a system where blind ants work in synchronization to each other to find the best food sources. They coordinate with each other by using pheromone evaporation mechanism. Bee Colony Optimization (BCO) is a specialization to Swarm Intelligence (SI) where the agents are honey bees. They communicate and exchange important information with each other regarding rich food source's location by "Waggle Dance". Bat algorithm (BA) is a meta-heuristic optimization algorithm inspired by the echolocation behavior of micro bats. It uses frequency tuning technique to overcome the deficiency.

The main goal of Swarm Intelligence is to increase the performance and solve complex issues. The amazing efficiency of natural swarm systems motivated many researchers to discover how swarms solve complex problems in nature. Amid the most recent thirty years, a great number of papers have been distributed in gatherings and diaries which are identified with Swarm Intelligence. The objective of this paper is to provide a broaden idea about the most three swarm technologies and critically analyze their behavior. Three of the most eminent swarm intelligence

techniques for getting reasonable solutions for optimizing issues in a sensible measure of computation time are Ant Colony Optimization (ACO), Bee Colony Optimization (BCO) and Bat Algorithm (BA). In order to investigate this objective, this paper will also analyze these technologies popular variants along with their applications and find out better technique for optimization.

The remainder of this paper is structured as follows: Section 2 presents a brief description about Swarm Intelligence, some Swarm Intelligence Examples, characteristics of Swarm Intelligence and their applications, advantages, disadvantages. Sections 3, 4, 5 provide an overview of three swarm intelligence techniques (Ant Colony Optimization, Bee Colony Optimization and Bat Algorithm) respectively with their behavior, pseudo-code, and flowchart and also discuss their variants with application. The last section summarizes the comparative study of these swarm intelligence techniques and points out better technique for optimization among them in swarm intelligence.

2. Swarm Intelligence

The articulation “Swarm Intelligence” was first presented by Gerardo Beni in 1988, with regards to autonomous robotic systems where simple agents organize themselves through closest neighbor association [2]. Swarm Intelligence (SI) is an emerging area in the field of computer science that plans and concentrates effective computational strategies for solving complex issues in a way that is enlivened by the conduct of genuine swarms or insect colonies. SI technique has been used for both combinatorial and continuous optimization problems in static and in distributed settings. Swarm is a group of homogeneous individual agents interact among them and with the environment without centralization. Agents are simple with limited capabilities, but by interacting with the other agents of their own kind they achieve the task necessary for their survival like the ants’ forage for the food (**Figure 1**), bees communicate with others by waggle dance (**Figure 2**), flock of birds flying together (**Figure 3**). The agents follow



Figure 1. The ants found food source and eating food together. (Photographer: Hameem).



Figure 2. Bees communicate with others by waggle dance.



Figure 3. Flock of birds flying together (*Photographer: Meherun*).

very simple rules. Simple local behavior of an agent's in swarm leads to global intelligent behavior. In the nature there exist many natural swarms. Some of them are: Insect colonies like ants, bees, flock of birds.

There are several swarm intelligence models. These are: Ant Colony Optimization, Particle Swarm Optimization, Genetic Algorithm, Artificial Bee Colony, Bat Algorithm, Bacterial Foraging, Cat Swarm Optimization, Artificial Immune System, Cuckoo Search Algorithm, Shuffled Frog Leaping, Glowworm Swarm Optimization, Monkey Search, Firefly Algorithm, Roach Infestation Optimization etc.

2.1. Characteristics of Swarm Intelligence

Every single social insect demonstrates amazing collective problem-solving capabilities. Properties related with their group behavior like self-organization, robustness, flexibility, Stigmergy etc. are seen as characteristics that artificial

systems for optimization, control or task execution should exhibit [3]. In the most recent decade, various efforts have been made to take social insects for instance and develop algorithms motivated by their entirely self-organized conduct. These approaches can be subsumed under the concept of “Swarm Intelligence”. There are some common characteristics of swarm intelligence. These are:

- 1) Attraction: Individuals are attracted to each other.
- 2) Focus on Same Direction: When they come nearer in space, they begin to focus in the same direction.
- 3) Avoid Collision: They maintain a strategic distance from impact by moving far from each other. They keep certain separation from each other.
- 4) Stigmergy [1]:
 - a) Agents in a roundabout way cooperate by means of natural adjustment, the marvel known as Stigmergy.
 - b) Stigmergy is essentially the setting mindfulness.
 - c) Stigmergy decouples agents’ cooperation.
- 5) Self-Organization: Individuals connect with nearby, close neighbors and trust just a couple of them. This control is known as self-organization [4]. The bases of self-organization are:
 - a) Positive feedback (amplification): It advances better arrangements by designating to them more agents. e.g. recruitment and reinforcement.
 - b) Negative feedback (for offset and adjustment): It might avoid that all individuals unite to a similar conduct or to a similar state, e.g. limited number of available foragers.
 - c) Amplification of fluctuations (randomness, errors, random walks).
 - d) Multiple social interactions.
 - e) There is a continuous strain between positive feedback and negative feedback and this is the thing that really occurs in most known self-organization marvels, e.g. cell automata, markets, complex systems, and so on.
- 6) Strong Robustness: Swarm intelligence algorithm crowd control are dispersed, there is no focal control. Therefore, their workplace is in a wide range, one or some individual issues can not affect the gathering, strong robustness.
- 7) Simple: Execution of every individual operation is basic and simple to implement.
- 8) Better scalability: The measure of data of every individual detecting is limited.
- 9) Distribution: They have potentially parallelism and distributed features. Agents choose their actions and after that complete them.
- 10) Cooperation:
 - a) Agents collaborate to develop a non-deterministic, complex aggregate conduct.
 - b) Agents collaborate keeping in mind the end goal to understand complex undertakings.

11) Emergence: Complicated intelligent behavior rises up out of basic operators following straightforward principles.

a) Weak Emergence: Can follow the agent conduct from new properties.

b) Strong Emergence: Agent behavior isn't straightforwardly traceable from new conduct.

12) Imitates Nature: Artificial swarm is composed by imitating the natural swarm behavior. Here is some common Swarm behavior:

a) Foraging: To search for the food.

b) To construct the nest.

c) To move in the environment.

2.2. Advantages of Swarm Intelligence

Swarm Intelligence is the emergent collective intelligence of groups where they do various remarkable things together. There are some biological advantages of swarm systems for survival of the species in their natural evolution. These are:

- Flexible: The swarm systems are flexible because agents can be easily added or removed without influencing the structure. The colony responds to internal disturbances and external challenges. The control mechanism is not dependent on swarm size.
- Robust: The systems are robust because agents are simple in design and can complete tasks if some agents fail. Failure of a single agent has a little impact on the system performance.
- Scalable: The systems are scalable because the same control architecture can be applied to a few agents or millions of agents.
- Decentralized: The swarm systems are decentralized. There is no central control in the colony.
- Self-organized: The swarm systems are self-organized. The solutions are emergent rather than pre-defined.
- Adaptation: The swarm systems are able to easily adapt to new situations.
- Speed: Changes in the system can be engendered quick.
- Modularity: Agents act freely of other system layers.
- Parallelism: Agents' operations are inalienably parallel.
- Stability and Adaptability: Swarms are relied upon to adjust ecological variations without quickly changing modes since mode changing costs vitality.
- Quality: Aside from essential computation capacity, a swarm ought to have the capacity to reaction to quality components, for example, food and security.

2.3. Limitations of Swarm Intelligence

The capability of swarm intelligence is in reality quickly developing and extensive. It offers an option for solving complex problems. That being stated, SI still have a few limitations. These are:

- Behavior: The swarm systems are difficult to predict the behavior from the

individual rules.

- Knowledge: The functions of colony couldn't be comprehended with the information of working of an agent.
- Sensitivity: Even a little change in the basic tenets outcomes in various gathering level conduct.
- Action: Agent behavior looks like clamor as activity of decision is stochastic.
- Non-optimal: The swarm systems are highly redundant and they have no central control, they tend to be ineffective. Assigning resources is not efficient and duplication of efforts is always unbridled.

2.4. Application Areas of Swarm Intelligence

Swarm Intelligence principles have been successfully applied in a variety of problem domains and applications. Some successful application areas in SI are routing, clustering, optimization, scheduling, etc. as shown in **Figure 4**.

2.4.1. Routing

This depends on the rule that backward ants utilize the useful information assembled by the forward ants on their excursion from source to destination. The AntHocNet routing algorithm is used for MANETs (mobile ad hoc networks) [5].

2.4.2. Clustering

A cluster is a collection of agents which are similar and are dissimilar to the agents in other clusters. For e.g. to clean up the ants' nests a cluster is formed [6].

2.4.3. Scheduling

The emphasis is on the relative position of the activity as opposed to its immediate predecessor or its immediate successor in the schedule and global pheromone evaluation rule is followed. For e.g. ACO algorithm can be used to schedule large scale work flows [7].

2.4.4. Optimization

An optimization problem is the problem of finding the best solution from all the feasible solutions. For e.g. optical network optimization is an application of particle swarm intelligence.

3. Ant Colony Optimization

Ant colony optimization (ACO) is one of the first [8] techniques for optimization

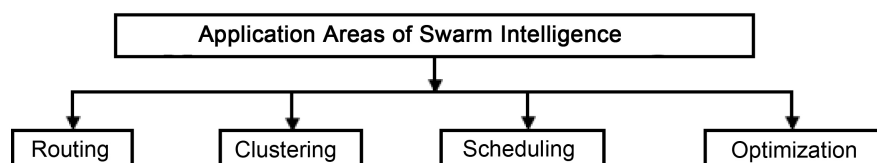


Figure 4. Application areas of Swarm Intelligence.

inspired by the Stigmergy and foraging behavior of ant colonies. Ants live in a group and their main goal is group survival rather than an individual survival. Each ant can lay pheromone to communicate with others and each ant is also able to follow the route marked with pheromone laid by other ants. The indirect communication between the ants empowers them to discover short ways between their nest and food sources. This characteristic of real ant colonies is abused in ACO algorithms to solve hard combinatorial enhancement issues like Consecutive requesting issue, booking issue, chart shading, vehicle steering.

Ant Colony Optimization (ACO) was first introduced as a powerful search and a heuristic approach for the solution of combinatorial optimization problems by M. Dorigo and his colleagues in 1991 [9]. In 1992 [10], M. Dorigo proposed an ACO algorithm in his PhD proposition which was expected to scan for an ideal way in a graph, in view of the conduct of ants looking for a way between their settlement and a wellspring of nourishment. After two years, in 1994 [11], A. Coloni, M. Dorigo *et al.* applied Ant System to the job shop scheduling problem. In 1996 [12], Ant System was first applied to the Travelling Salesman problem by M. Dorigo *et al.* In the next year [13], M. Dorigo and Gambardella modified the system to improve the performance of Ant System and to apply it to other optimization problems. They have adopted a different kind of rule named “pseudo-random proportional rule” which helps to balance between the exploitation and exploration of the search process was shown in Ant Colony System (ACS) algorithm. Botee and Bonabeau discussed the application of ACO and ACS for the traveling salesman problem in 1998 [14]. They also suggested a new approach of simple ACO for TSP. In their approach, they evolved some important parameters like number of ants, importance of trail etc. using Genetic Algorithm. They concluded that the automated parameters using GA produce better solution. In 1999, Dorigo *et al.* dealt with the recent works related to ACO. They discussed various combinatorial optimization problems such as Traveling Salesman Problem, Quadratic Assignment Problem, Job Scheduling Problem and so on. They concluded that by discussing many potential advantages of ACO algorithm over other algorithms. M. Dorigo and Stuzelle discussed some of the essential properties of ACO in 2001. They proposed a simplified ant based algorithm called S-ACO which applied for finding the shortest path in the graph. In 2005 [15], M. Dorigo and C. Blum overviewed some recent efforts to develop a theory of ant colony optimization. They also discussed convergence, presented connections between ACO algorithms and the stochastic gradient ascent and cross-entropy methods within the framework of model-based search, and lastly examined the influence of search bias on the working of ACO algorithms.

3.1. Foraging Behavior of Ants

Ants live in colonies and are “practically visually impaired”, they lay pheromone in transit from the home when they go looking for food source. On achieving the

food point the ant gathers the food and returns a similar way the compound pheromone is laid. This route is attracted by the various ants. More ants following turn out to be more appealing for the other ants. ACO is the first algorithm captivated by the look for optimal path through the pheromone correspondence in light of the conduct of ants to find the shortest route in investigating the food. This strategy is called as Stigmergy [1]. Foraging behavior of ants is the best case for clarifying the capacity of ant colonies. Foraging behavior of ants is as per the following:

- 1) Individual ants go looking for food; they wander randomly around colonies looking for food source.
- 2) Ants cannot specifically communicate with each other; indirect communication is called as Stigmergy.
- 3) At the point when the ants find their food source they promptly return close to the home on its way back it leaves a substance called as pheromone. These pheromones are unpredictable in nature they continue dissipating. They use pheromone network to ensure no ant overtake another. So, they never have traffic jams or bump into each.
- 4) Ants are fit for detecting this pheromone and the route is pulled in by different ants, they proceed onward a similar track. What's more, every ant leaves their substances and thickness the track so that if some other ants are in the source then they can follow the pheromone thickness and discover their food source.
- 5) In the event that other ant has discovered most brief ways for a similar food source, then that shortest path can be followed by many other ants and this route turns out to be more appealing as increment in the convergence of pheromone.
- 6) Their speed remains constant. No matter how many ants are on the trail.
- 7) On the off chance that there is any obstacle in the route then it will move arbitrarily to start with in any case, later they will locate the shortest path.

Figure 5 explains the behavior of ants.

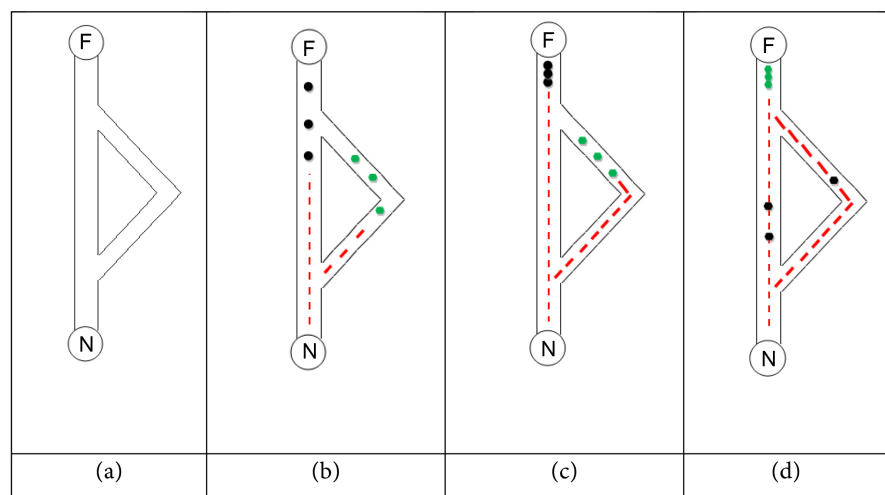


Figure 5. Ant colonies behavior for finding the shortest food source.

- 1) All ants are in the nest. There is no pheromone in the environment.
- 2) The foraging starts. Half of the ants choose the shortest path and the rest of the part chooses the longest path to the food source.
- 3) The ants that have taken the short path have arrived earlier at the food source. Therefore, when returning, the probability that they again choose the shortest path is higher.
- 4) The pheromone trail on the shortest path receives, in probability, a stronger reinforcement, and the probability of taking this path grows. Finally, due to the evaporation of the pheromone on the longest path, the whole colony will, in probability, use the shortest path.

3.2. Basic Algorithm of Ant Colony Optimization

The basic steps of the Ant Colony Optimization can be summarized in the following:

```

Begin
     $t := 0;$ 
     $P(t) := \text{InitPopulation}();$ 
     $\text{Evaluate}(P(t));$ 
     $P_{\text{pheromone}}(t) := \text{InitPheromone}(P(t));$ 
     $P_{\text{rior}} := \text{InitPriorKnowledge}();$ 
    While (stop criteria unsatisfied)
         $P(t+1) = \text{FindPath}(P(t), P_{\text{pheromone}}(t), P_{\text{rior}});$ 
         $\text{Evaluate}(P(t+1));$ 
         $P_{\text{pheromone}}(t+1) = \text{UpdatePheromone}(P(t+1), P_{\text{pheromone}}(t), P_{\text{rior}});$ 
         $t = t + 1;$ 
    End
End

```

Algorithm 3.1. Basic Algorithm of ACO.

Here, t is the number of iterations, and $P(t)$ represents the t^{th} generation. $P_{\text{pheromone}}(t)$ and P_{rior} are the pheromone matrix of the t^{th} generation and prior knowledge information matrix, respectively. They are used to guide ants' path finding behavior. Therefore, it contains two fundamental operators, path finding and pheromone updating, which aim at guiding the population searching by the integration of static prior knowledge and dynamic pheromones which are formed by every individual's step [16].

Figure 6 represents the basic flowchart of ACO [17].

3.3. Ant Colony Optimization Variations

There are many variations for ACO such as Ant System (AS), Ant Colony System (ACS), Max-Min Ant System (MMAS), Elitist Ant System (EAS), Continuous Orthogonal Ant Colony (COAC), Recursive Ant Colony Optimization (RACO) and Rank-based Ant System (AS_{rank}). The main features of these algorithms are discussed below:

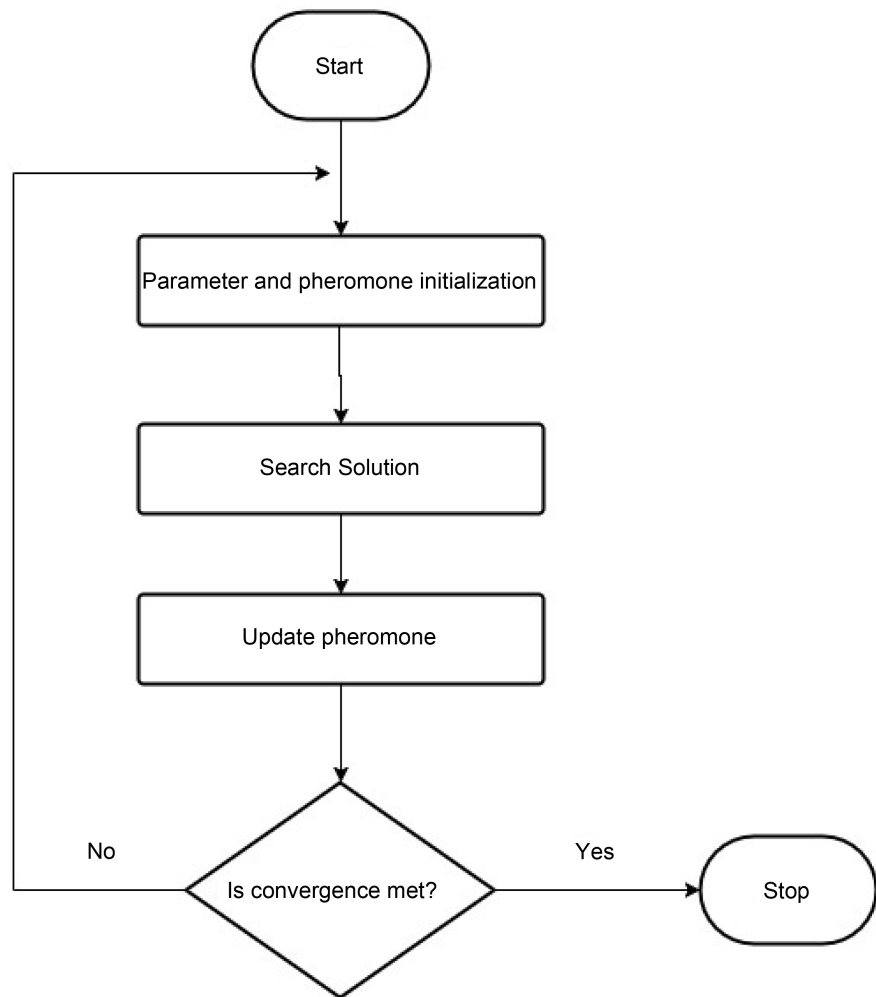


Figure 6. Basic flowchart of ACO.

3.3.1. Ant System (AS)

- 1) Ant system (AS) was the first ACO algorithm [18].
- 2) The pheromone values are updated by all the ants.
- 3) Reinforced Ant System (RAS) is the same as Ant system, except that the global best solution is reinforced each iteration.
- 4) Three different versions of AS (Dorigo, M., Maniezzo, V., Colorni, A. (1991a)) were developed: they were called ant-density, ant quantity and ant-cycle.
- 5) **Applications:** TSP, Water Distribution System Design (WDSD), Graph Coloring, Vehicle Routing Problem (VRP), Quadratic Assignment Problem (QAP), Bus Network Design, Bus driver scheduling.

3.3.2. Elite Ant System (EAS)

- 1) In EAS, along with other ants, the global best solution (the best ant) deposits pheromone at every iteration [19].
- 2) Individual ants do not automatically leave pheromone.
- 3) Thus, the search is even more centralized around the global best solution.

- 4) **Applications:** Post-Enrolment Course Timetabling Problem.

3.3.3. Rank-Based Ant System (AS_{Rank})

- 1) In the Rank-Based Ant System (RBAS), all solutions are ranked according to their length [19].
- 2) The amount of pheromone deposited is then weighted for each solution, such that the more optimal solutions deposit more pheromone than the less optimal solutions.
- 3) **Applications:** TSP, WDSD.

3.3.4. Max-Min Ant System (MMAS)

- 1) To exploit the best solutions found during iteration or during the run of the algorithm, after each iteration only one single ant adds pheromone.
- 2) Only the global best ant allows secreting pheromone on its trail with an extra constraint for pheromone value.
- 3) Avoids premature stagnation.
- 4) To avoid stagnation of the search the range of possible pheromone trails on each solution component is limited to an interval.
- 5) Additionally, we deliberately initialize the pheromone trails to T_{max} , achieving in this way a higher exploration of solutions at the start of the algorithm [21].
- 6) **Applications:** TSP, QAP, protein folding, WDSD.

3.3.5. Ant Colony System (ACS)

- 1) State Transition Rule: Provides a direct way to balance between exploration of new edges and exploitation of a priori and accumulated knowledge about the problem.
- 2) Global Updating Rule: Is applied only to edges which belong to the best ant tour.
- 3) Local Pheromone Update Rule: While ants construct a solution, a local pheromone updating rule is applied [22].
- 4) **Applications:** TSP, VRP, WDSD.

3.3.6. Continuous Orthogonal Ant Colony (COAC)

- 1) Developed by using the orthogonal design method.
- 2) The main steps in continuous orthogonal ant colony (COAC) algorithm are the orthogonal exploration and the global modulation [23].
- 3) COAC can solve unimodal functions with higher accuracy.
- 4) **Applications:** Feed forward neural network training.

3.3.7. Recursive Ant Colony Optimization (RACO)

- 1) The Recursive Ant Colony System (RACS) algorithm applies a partitioning scheme to the problem in a manner analogous to the recursive merge sort based on the divide and conquers technique.
- 2) The RACS algorithm is advantageous for larger problems where a convergent path is not easily found in limited time by using solely the Ant Colony algo-

rithms [24].

3) It can also avoid stagnation behavior by breaking down the large problem into smaller and exploring shortest path for each sub-problem. Finally, it is combined to generate an optimal path. In each sub-problem, the iteration is performed recursively to obtain the shortest path in that sub problem.

4) **Applications:** Estimation of Parameters of a Function.

3.4. Summarizing ACO Variations

There are two broad categories of optimization problems. One is discrete optimization problem and the other is continuous optimization problem. The AS, AS_{Rank}, ACS, RACO, EAS and MMAS algorithms fall into the category of discrete optimization problems while the CIAC and COAC algorithms are designed for continuous optimization problems. The variations mostly differ in the global updating rule. **Table 1** summarizes some of the ACO variations along with their authors, invention year and updating rule.

4. Bee Colony Optimization

It is perhaps worth mentioning here that in the midst of the most recent decade, the researchers have been studying the behavior of bees in an attempt to utilize the swarm intelligence concept and build up various artificial systems. Here are some optimization algorithms inspired by bees' behavior that appeared during

Table 1. Summarizing ACO variations.

| Algorithm | Authors | Year | Tour Construct | Updating Rule | References |
|--------------------|---------------------------|------|--------------------------------|--|------------|
| AS | Dorigo <i>et al.</i> | 1991 | random proportional rule | update pheromone at every iteration by all ants | [18] [19] |
| EAS | Dorigo <i>et al.</i> | 1992 | random proportional rule | deposit pheromone at every iteration by the best ant | [19] |
| AS _{rank} | Bullnheimer <i>et al.</i> | 1997 | random proportional rule | deposit pheromone according to the rank of each ant | [20] |
| MMAS | Stutzle & Hoos | 2000 | random proportional rule | after each iteration deposit pheromone by only one single ant either the best ant; interval $[T_{min}, T_{max}]$ | [21] |
| ACS | Dorigo & Gambardella | 1996 | pseudorandom proportional rule | deposit pheromone by the best ant | [22] |
| COAC | Xiao-Min Hu | 2007 | random proportional rule | deposition of pheromone is proportional to the visiting time of ants | [23] |
| RACO | Karp | 1997 | random proportional rule | the pheromone is updated only for those iteration where the ants move | [23] |

the last decade: Bee System, Bee Colony Optimization (BCO), Marriage in Honey-Bees Optimization (MBO), Bee Hive, Honey Bees, Artificial Bee Colony (ABC), Bee System Optimization (BSO), Bees Algorithm, Honey Bee Marriage Optimization (HBMO), Fast Marriage in Honey Bees Optimization (MHBO), Virtual Bee Algorithm (VBA), and Modified MBO.

Bee Colony Optimization (BCO) is one of the first algorithms that utilization fundamental standards of aggregate honey bee insight for dealing with combinatorial optimization problems [24]. It is a meta-heuristic inspired by the foraging behavior of honey bees. The essential thought behind BCO is to construct the multi operator framework which ready to solve different combinatorial optimization issues. The honey bee system is a standard case of composed collaboration, very much planned association, coordination, work division, synchronous task execution, specific people, and well-weave correspondence.

Artificial bee colony usually consists of a small number of individuals. They investigate through the search space looking for the feasible solutions. So as to locate the best possible solutions, autonomous artificial bees work together and exchange information. Utilizing collective knowledge and information sharing, simulated honey bees focus on the all the more encouraging ranges and gradually surrender arrangements from the less encouraging ones. Step by step, artificial bees altogether produce and enhance their solutions. The BCO search is running in iterations until some predefined stopping criterion is fulfilled.

Every single simulated honey bee is situated in the hive toward the start of the search procedure. Amid the search procedure, honey bees communicate directly. Each artificial bee makes a progression of nearby moves, and along these lines incrementally develops a solution of the problem. Honey bees are adding solution components to the present halfway arrangement until the point that they make at least one or more feasible solutions. When flying through space, artificial bees perform forward advance or in reverse advance. Amid a forward advance, honey bees make various partial solutions. They do this by means of a blend of individual investigation and aggregate understanding from the past. From that point forward, they play out a backward advance, *i.e.* they come back to the hive. In the hive, all honey bees take an interest in a basic leadership process. The search process is composed of iterations. Every emphasis closes when one or more feasible solutions are made.

In 2005, Karaboga D. examined the foraging behavior of honey bee swarm and proposed another algorithm simulating this behavior for solving multi-dimensional and multi-modal optimization problems, called Artificial Bee Colony. In the model, there are three groups of bees. These are: employed bees, onlookers and scouts. In the ABC algorithm, the main portion of the colony comprises of the utilized honey bees and the second half incorporates the onlookers. The algorithm is tried on three surely understood test capacities. From the simulation results, it is inferred that the proposed algorithm can be utilized for solving unimodal and multi-modal numerical optimization problems. Dur-

ing the period of 2001-2003, Lucic and Teodorovic tested the Bee Colony Optimization approach in the case of Travelling Salesman Problem (TSP).

4.1. Foraging Behavior of Bees

Artificial honey bee swarms comprise of three fundamental parts. These are food sources, employed foragers and unemployed foragers. The model additionally characterizes two driving methods of the behavior: the recruitment to a nectar source and the abandonment of a source.

1) Food Sources: The value of a food source relies upon numerous factors, for example, its closeness to the nest, its lavishness or focus of its vitality, and the simplicity of separating this energy. For straightforwardness, the “profitability” of a food source can be spoken to with a solitary amount [26].

2) Employed Foragers: They are related with a present food source, or in abuse, they bring with them information about this source, particularly the distance, area, and productivity, to impart this to a specific likelihood with other mates.

3) Unemployed Foragers: They are continually at pay special mind to a food source to abuse. There are two types of unemployed foragers. These are: scouts, searching the environment surrounding the nest for new food sources and on-lookers waiting in the nest and building up a food source through the information shared by employed foragers.

In order to understand the basic behavior characteristics of foragers better, let us examine **Figure 7**. Accept that there are two discovered food sources: A and B. At the earliest reference point, a potential forager will begin as an unemployed forager. That honey bee will have no information about the food sources around the nest. There are two conceivable [27] alternatives for such a bee:

1) It can be a scout and begins searching around the nest unexpectedly for a food because of some interior inspiration or conceivable outer piece of information (S in **Figure 7**).

2) It can be a recruit after watching the waggle dances and begins searching for a food source (R in **Figure 7**). After finding the food source, the bee uses its own particular capability to remember the location and after that quickly starts abusing it. As an “employed forager” it will take a heap of nectar from the source and come back to the hive and empty the nectar to a food store.

After emptying the food, the bee has the following three options:

1) It turns into an uncommitted follower after abandoning the food source (UF).

2) It dances and afterwards recruits nest mates before coming back to a similar food source (EF1).

3) It keeps on foraging at the food source without recruiting other bees (EF2).

The tests affirmed that new honey bees start foraging at a rate corresponding to the contrast between the eventual total number of bees and the number of present foraging. On account of honey bees, the essential properties on which

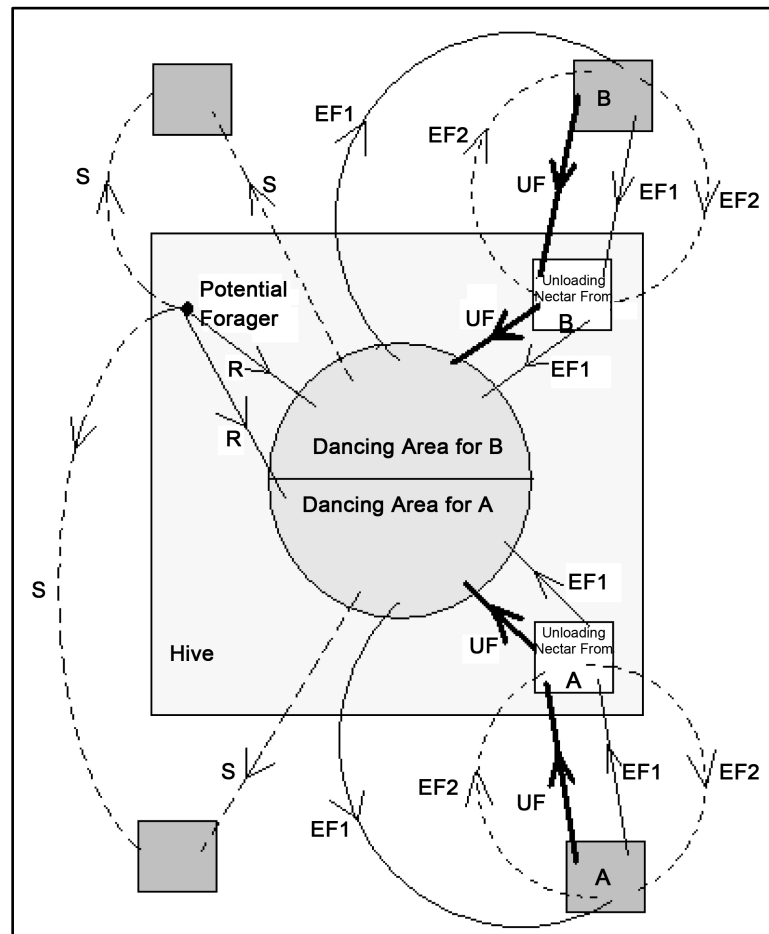


Figure 7. The foraging behavior of honey bees [27].

self-organization depend are as follows:

- 1) Positive Feedback: As the nectar amount of food sources increases, the number of onlookers visiting them increases, as well.
- 2) Negative Feedback: The exploitation process of poor food sources is ceased by bees.
- 3) Fluctuations: The scouts do a random search process for finding new food sources.
- 4) Multiple Interactions: Information exchange through waggle dance.

4.2. Basic Bee Colony Optimization Algorithm

Like Dynamic Programming, the BCO additionally solves combinatorial optimization problems in stages. Each of the characterized stages includes one optimizing variable. Give us a chance to signify by $ST = \{st_1, st_2, \dots, st_m\}$ a limited arrangement of pre-chosen stages, where m is the number of stages. By B we signify the number of bees to take an interest in the search process, and by I the total number of iterations. The arrangement of fractional arrangements at organize st_j is signified by S_j ($j = 1, 2, \dots, m$).

The following is the basic algorithm [28] of the Bee Colony Optimization:

- 1) *Initialization.* Determine the number of bees B , and the number of iterations I . Select the set of stages $ST = \{st_1, st_2, \dots, st_m\}$. Find any feasible solution x of the problem. This solution is the initial best solution.
- 2) Set $i = 1$. Until $i = I$, repeat the following steps.
- 3) Set $j = 1$. Until $j = m$, repeat the following steps.
Forward pass. Allow bees to fly from the hive and to choose B partial solutions from the set of partial solutions S_j at stage st_j .
Backward pass. Send all bees back to the hive. Allow bees to exchange information about quality of the partial solutions created and to decide whether to abandon the created partial solution and become again uncommitted follower, continue to expand the same partial solution without recruiting the nest mates, or dance and thus recruit the nest mates before returning to the created partial solution. Set, $j = j + 1$.
- 4) If the best solution x_i obtained during the i^{th} iteration is better than the best-known solution, update the best-known solution ($x = x_i$).
- 5) Set, $i = i + 1$.

Algorithm 4.1. Basic algorithm of BCO.

On the other hand, forward and backward passes could be performed until the point that some other ceasing condition is fulfilled. The conceivable ceasing conditions could be, for instance, the most extreme aggregate number of forward/backward passes, or the greatest aggregate number of forward/backward goes between two target work esteem enhancements.

Figure 8 represents the basic flowchart [29] of BCO.

4.3. Bee Colony Optimization Variations

There are some popular variations of BCO such as MABC, PABC, IABC, CABC etc. All of these are inspired by honey bees. The main features of these variations are given below:

4.3.1. Modified ABC (MABC)

- 1) Introducing constrained handling procedure to the original ABC to tackle constrained optimization problems.
- 2) The workers used Deb's rules of handling constrained strategy in the ABC selection process, instead of greedy selection procedure.
- 3) The performance was better as compared to PSO and DE (Differential Evolution).

4.3.2. Parallelized ABC (PABC)

- 1) Divides artificial agents into independent subpopulations; Ripple communication strategy used for exchange of information.
- 2) The performance of PABC-RC in terms of the convergence behavior, accuracy and speed as tested on the benchmark functions showed that PABC-RC increased the accuracy and speed of convergence of finding the near best solution over ABC by 53% and 9%, respectively.

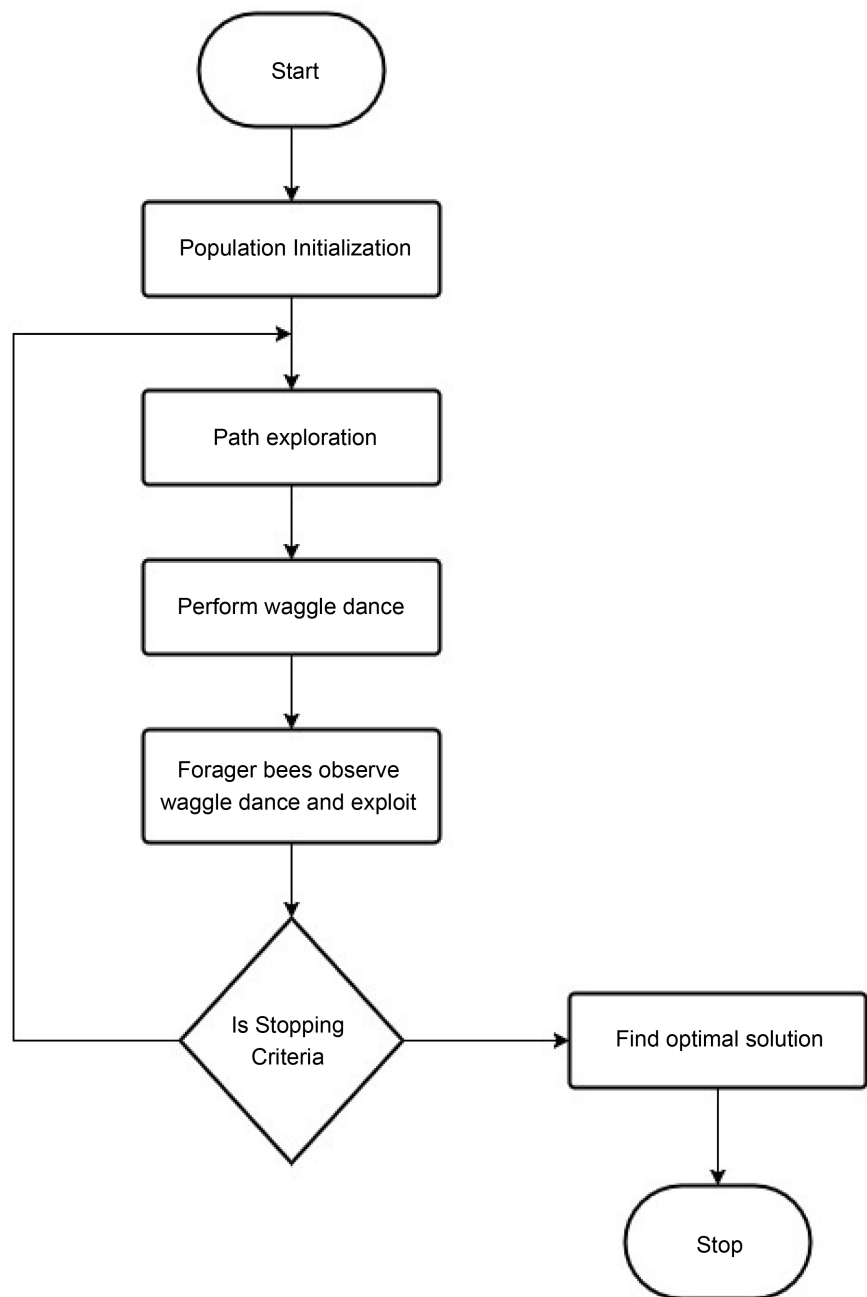


Figure 8. Flowchart of BCO.

4.3.3. MABC

- 1) Control parameters used:
 - a) Modification Rate (MR)
 - b) Scaling Factor (SF)
 - c) Limit
- 2) MR was used to enhance the convergence rate of ABC.
- 3) SF was used to verify the magnitude of changes when generating a neighboring solution.
- 4) When compared to standard ABC and other state of art techniques MABC

performed better.

4.3.4. Interactive ABC

- 1) Universal gravitation force is used to modify movements of onlooker bees.
- 2) The idea was used to enhance the exploitation capability of the original ABC.
- 3) Performance of the IABC with varied number of employed bees was tested on five numerical benchmark functions and the results were compared with the original ABC and PSO, where the IABC performed better.

4.3.5. Cooperative ABC

- 1) Produced final solution uses information from all the populations.
- 2) It results in improved solution quality & convergence speed.
- 3) As the CABC algorithm can be efficiently used for multivariable, multimodal function optimization, it can be applied to solve clustering problems.
- 4) The results of comparison with other algorithms like ACO, PSO illustrated that the proposed CABC optimization algorithm can be considered as a viable and efficient method to solve optimization problems.

4.4. Summarizing BCO Variations

Table 2 summarizes the different popular variations of BCO algorithm listing the special domain for which they were first designed and their authors.

5. Bat Algorithm

Bat algorithm (BA) was inspired by the echolocation behavior of micro bats. Bats use echolocation to detect prey, hunts, and avoid obstacles. Echolocation is a type of sonar which helps bat to detect the distance of the prey. Bats can also identify the shape, position, and angle of the prey with the help of echolocation. BA has many advantages compared with other existing swarm intelligence algorithms. They have less control parameters, excellent global optimization ability, and implementation simplicity which have shown excellent performances in

Table 2. Summarizing BCO Variations.

| Algorithm | Authors | Year | Application | References |
|--------------------------------|---------------------------|------|---|------------|
| Modified ABC (MABC) | Karaboga | 2007 | Constrained optimization problems | [30] |
| Parallelized ABC (PABC) | Luo <i>et al.</i> | 2010 | Numerical Optimization problems | [31] |
| MABC | Akay and Karaboga | 2010 | Real Parameter Optimization | [32] |
| Interactive ABC | Tsai <i>et al.</i> | 2009 | Numerical Optimization problem | [33] |
| Cooperative ABC (CABC) | Wenping Zou <i>et al.</i> | 2010 | Multivariable-multimodal function optimization, Clustering problems | [34] |

some classes of optimization problems. Due to its simplicity, convergence speed, and population feature, it has been intensively used to solve academic and real-life problems, like multi objective optimization, engineering optimization, cluster analysis, scheduling problems, structure optimization, image processing, manufacturing designing, and various other problems.

It has been demonstrated that BA is a magnificent and capable algorithm for worldwide optimization issues, discrete enhancement issues, and obliged advancement issues. However, because of the absence of good harmony amongst investigation and abuse in essential BA, the algorithm once in a while comes up short at finding worldwide ideal and is effectively caught into neighborhood optima. Numerous endeavors have been made to enhance the execution of BA [35].

Bat algorithm was developed by Xin-She Yang in 2010. He developed the bat algorithm with the following three idealized rules:

- 1) All bats utilize echolocation to sense distance, and they additionally know the distinction between food/prey and foundation obstructions in some magical way.
- 2) Bat flies with a frequency f_{\min} from the current position x_i at a velocity v_i , but varying loudness and frequency.
- 3) The loudness varies from a minimum value (A_{\min}) to maximum value (A_0).

In 2013, Yang utilized the bat algorithm to think about topological shape optimization in microelectronic applications with the goal that materials of various warm properties can be set such that the warmth exchange is most productive under stringent imperatives. It can likewise be connected to do parameter estimation as an opposite issue. In the event that an opposite issue can be legitimately figured, at that point bat algorithm can give preferred outcomes over minimum squares techniques and regularization strategies.

5.1. Echolocation Behavior of Bats

Bats are divided into two major groups, *i.e.*, mega bats and microbats. They have three habitats as they feed through the whole year, *i.e.*, roost, foraging habitats, and commuting habitats. The foraging habitats are used to find food, while the commuting habitats are used to travel between roosts. In general, the foraging behavior of bats can be divided into three phases. These are search phase, pursuit phase and capture phase. They are very flexible in their foraging behavior, such as catching insects on the wing, picking insects from vegetation and pouncing on prey close to the ground [36]. At each phase, they show different behaviors based on a perceptual system, *i.e.*, bats echolocation. Behaviors of bats are as follows:

- 1) Most microbats are insectivores. They use a type of sonar named echolocation to detect prey, avoid obstacles, and locate their roosting crevices in the dark.
- 2) They emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects.

3) Most bats use short, frequency modulated signals to sweep through about an octave, while others more often use constant-frequency signals for echolocation as shown in **Figure 9**.

4) Microbats use the time delay from the emission and detection of the echo, the time difference between their two years, and the loudness variations of the echoes to build up three-dimensional scenario of the surrounding, as shown in **Figure 10**.

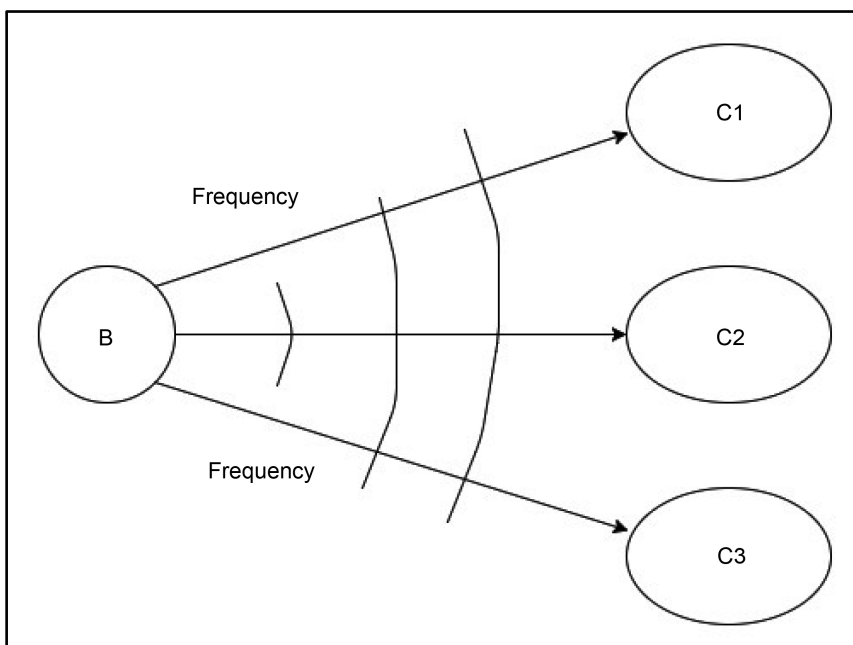


Figure 9. Bat send sound signal with frequency, F .

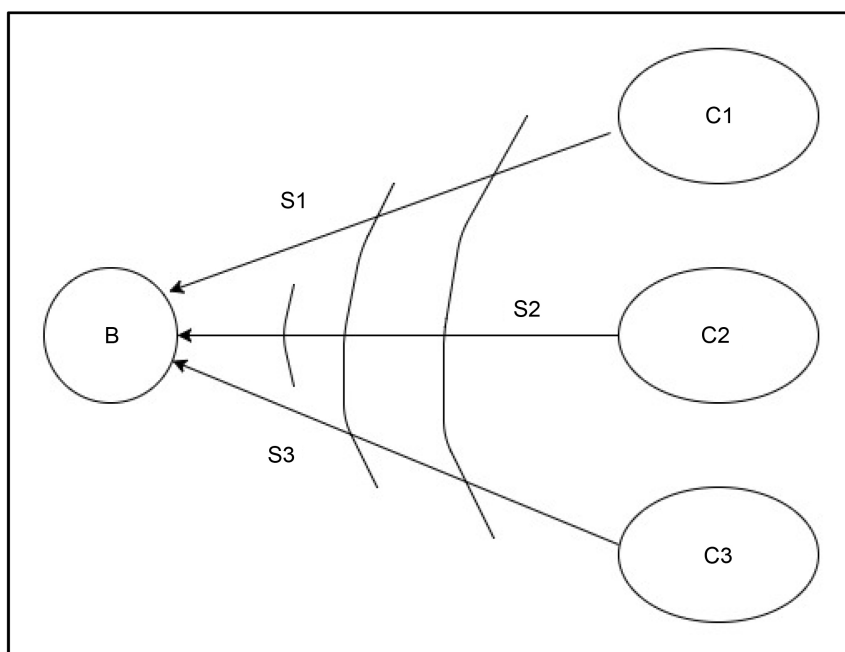


Figure 10. Echo signal used to calculate the distance, S .

5) They can recognize the distance and orientation of the objective, the sort of prey, and even the moving rate of the prey, for example, little creepy crawlies.

5.2. Basic Algorithm of Bat Algorithm

The basic steps of the Bat Algorithm [37] can be summarized in the following:

```

Initialize the bat population  $x_i$  ( $i = 1, 2, \dots, n$ ) and  $v_i$ 
Initialize frequency  $f_i$ , pulse rates  $r_i$ , and the loudness  $A_i$ 
While ( $t < \text{Max numbers of iterations}$ )
    Generate new solutions by adjusting frequency
    and updating velocities and locations/solutions
    if( $\text{rand} > r_i$ )
        Select a solution among the best solutions
        Generate a local solution around the selected best solution
    end if
    Generate a new solution by flying randomly
    if( $\text{rand} < A_i \ \& \ f(x_i) < f(x^*)$ )
        Accept the new solutions
        Increase  $r_i$  and reduce  $A_i$ 
    end if
    Rank the bats and find the current best  $x^*$ 
end while
  
```

Algorithm 5.1. Basic algorithm of Bat Algorithm.

Figure 11 represents the basic flowchart [38] of Bat Algorithm.

5.3. Bat Algorithm Variations

5.3.1. Fuzzy Logic Bat Algorithm (FLBA)

Khan *et al.* displayed a variation in 2011 by bringing fuzzy logic into the bat algorithm; they called their variant fuzzy bat algorithm.

Application: Dynamical Parameter Adaptation [39].

5.3.2. Multi Objective Bat Algorithm (MOBA)

In the same year, Yang stretched out BA to manage objective optimization, which has shown its effectiveness for settling a couple of plan benchmarks in engineering.

5.3.3. K-Means Bat Algorithm (KMBA)

Komarasamy and Wahi (2012) introduced a combination of K-means and bat algorithm (KMBA) for efficient clustering.

5.3.4. Chaotic Bat Algorithm (CBA)

Lin *et al.* (2012) introduced a chaotic bat algorithm using Lévy flights and chaotic maps to do parameter estimation in powerful organic frameworks.

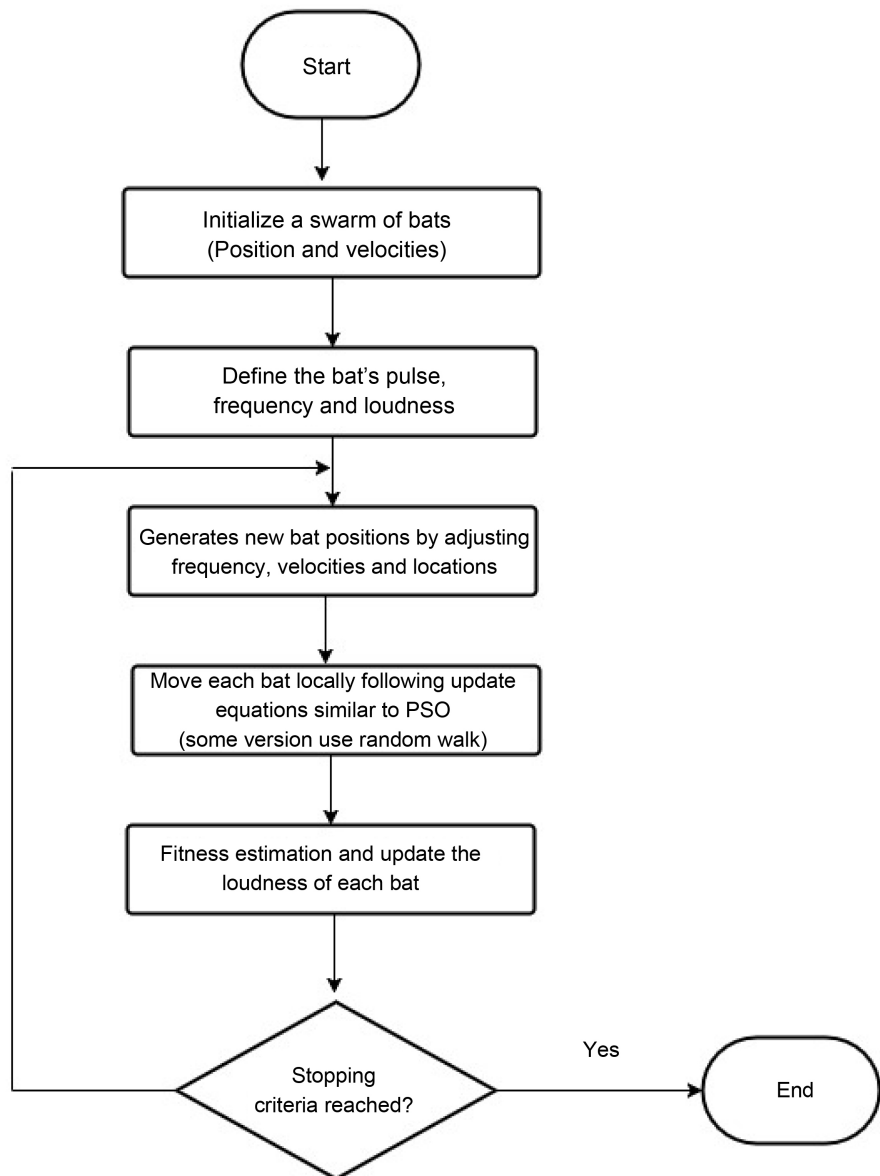


Figure 11. Flowchart of bat algorithm.

5.3.5. Binary Bat Algorithm (BBA)

Nakamura *et al.* (2012) built up a discrete adaptation of bat algorithm to solve classifications and feature selection problems.

Application: Solve 0 - 1 knapsack problem [40].

5.3.6. Differential Operator and Lévy Flights Bat Algorithm (DLBA)

After one year, in 2013, Xie *et al.* introduced a variant of bat algorithm utilizing differential operator and Lévy flights to solve function optimization problems.

5.3.7. Improved Bat Algorithm (IBA)

Jamil *et al.* (2013) expanded the bat algorithm with a decent combination of Lévy flights and subtle variations of loudness and pulse emission rates. They tried the IBA versus more than 70 distinctive test functions and proved to be

very efficient.

There are different upgrades and variations of bat algorithm [41]. For instance, Zhang and Wang (2012) utilized change to upgrade the decent variety of solutions and afterward for image matching. Furthermore, Wang and Guo (2013) hybridized bat algorithm with harmony search and have created a hybrid bat algorithm for numerical optimization of function benchmarks.

Then again, Fister Jr *et al.* (2013) built up a hybrid bat algorithm utilizing differential evolution as a local search part of bat algorithm, while Fister *et al.* (2013) incorporate quaternions into bat algorithm and exhibited a quaternion bat algorithm (QBA) for computational geometry and expansive scale optimization problems with extensive rotations. It can be expected that more variations are still under dynamic research.

6. Behavioral Analysis of ACO, BCO and BA

The objective of this paper was to analyze the behavior of the most three popular SI models, namely, ACO, BCO and BA. Despite these models are principally similar in their inspirational origin, and are based on nature-inspired properties, they are fundamentally different in the following aspects.

Table 3 shows the behavioral analysis of ACO, BCO and BA.

Table 3. Behavioral analysis of ACO, BCO and BA.

| Parameter | ACO | BCO | BA |
|--------------------------------------|---|--|--|
| Inspiration | Foraging behavior of ants. | Foraging behavior of bees. | Echolocation behavior of microbats. |
| Agents | Ant | Bee | Micro bat |
| Communication | Indirect | Direct | Direct |
| Efficiency | Less efficient than BCO | More efficient than ACO | Accurate and Efficient |
| Selection of Final Path | Depends on the amount of Pheromone on trail. | Waggle dance enables the colony to evaluate the relative merit of different patches according to both the quality of the food they provide and the amount of energy needed to harvest it. More bees are sent to promising solutions. | They use a type of sonar called echolocation to detect prey, avoid obstacles, and locate their roosting crevices in the dark. |
| Objective Function Defined by | pheromone update | waggle dance | pulse rate and emission velocity |
| Computational Time | More as compared to BCO. | Less as compared to ACO. | Less as compared to ACO and BCO. |
| Adaptive | More adaptive in nature. | Less adaptive in nature. | Less adaptive in nature. |
| Applications | Scheduling problems, Assignment problems, vehicle routing, TSP, SPAM Detection, Image Processing etc. | Job shop Scheduling problems, Flow Shop Scheduling problems, Open shop Scheduling Problems, TSP, SPAM Detection, Data Mining etc. | Exergy modelling, Exergy changes in a gas turbine, Optimal capacitor placement for loss reduction in distribution systems, Mutation for image matching, Clustering, Data Mining etc. |

7. Conclusions

Swarm Intelligence is a source of inspiration from nature. In particular, it has many features which include self-organization, robustness, cooperation, scalability, flexibility etc. These capabilities suggest a wide variety of applications that can be solved by SI principles.

After analyzing the behavior of the three algorithms, it is seen that ACO is more appropriate for applications than BCO and BA where search space is less while BCO is appropriate for larger search space. In computer science, the ant colony algorithm is the most probable process for solving computational problems which can be used for finding the shortest path. The ants drop pheromones each time when they bring food, so that shorter paths will probably be more grounded. Some ants are still randomly searching for nearer food sources. When the food source is short, the direction is no longer settled with pheromones and gradually decreases. The ant-colony works on a very self-organized system. They can rapidly reach an established performance. On the other hand, bee repeated cycles for the search of processes to discover the best solution. Sometimes they depend on its environment. Bats confront more difficulty to deal with. Indeed, even with the presentation of many bat algorithm versions, still there are some open issues in parameter control, optimality of the arrangement, longer execution time and untimely union confronting the usage of Bat Algorithm and researchers are attempting to enhance them in alternate points of view.

References

- [1] Merkle, D. and Middendorf, M. (2002) Modeling the Dynamics of Ant Colony Optimization. *Evolutionary Computation*, **10**, 235-262. <https://doi.org/10.1162/106365602760234090>
- [2] Beni, G. (1998) The Concept of Cellular Robotic Systems. *International Symposium on Intelligent Control*, Arlington, 24-26 August 1988, 57-62.
- [3] Karaboga, D. (2005) An Idea Based on Honey Bee Swarm for Numerical Optimization.
- [4] Bonabeau, E., Theraulaz, G. and Dorigo, M. (1999) *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, New York.
- [5] Di Caro, G., Ducatelle, F. and Gambardella, L.M. (2004) AntHocNet: An Adaptive Nature-Inspired Algorithm for Routing in Mobile Ad Hoc Networks. *European Transactions on Telecommunications*, **16**, 443-455. <https://doi.org/10.1002/ett.1062>
- [6] Kanade, P.M. and Hall, L.O. (2003) Fuzzy Ants as a Clustering Concept. *22nd International Conference of the North American Fuzzy Information Processing Society*, Chicago, 24-26 July 2003, 227-232.
- [7] Chen, W.-N. and Zhang, J. (2009) An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem with Various QoS Requirements. *IEEE Transactions on Systems*, **39**, 29-43.
- [8] Dorigo, M. (2004) Ant Colony Optimization. *Scholarpedia*, **2**, 1461. <https://doi.org/10.4249/scholarpedia.1461>
- [9] Colorni, A., Dorigo, M. and Maniezzo, V. (1991) Distributed Optimization by Ant Colonies. Elsevier Publishing, Amsterdam, 134-142.

- [10] Dorigo, M. (1992) Optimization, Learning and Natural Algorithms.
- [11] Colorni, A., Dorigo, M., Maniezzo, V. and Trubian, M. (1994) Ant System for Job-Shop Scheduling. *Belgian Journal of Operations Research, Statistics and Computer Science*, **34**, 39-53.
- [12] Gambardella, L.M. and Dorigo, M. (1996) Solving Symmetric and Asymmetric TSPs by Ant Colonies. *Proceedings of IEEE International Conference on Evolutionary Computation*, Nagoya, 20-22 May 1996, 622-627.
<https://doi.org/10.1109/ICEC.1996.542672>
- [13] Dorigo, M. and Gambardella, L.M. (1997) Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, **1**, 53-66. <https://doi.org/10.1109/4235.585892>
- [14] Gambardella, L.M. and Taillard, É.D. (1998) Ant Colonies for the QAP.
- [15] Dorigo, M. and Blum, C. (2005) Ant Colony Optimization Theory: A Survey. *Theoretical Computer Science*, **344**, 243-278.
<https://doi.org/10.1016/j.tcs.2005.05.020>
- [16] Tao, F., Zhang, L. and Laili, Y. (2014) Configurable Intelligent Optimization Algorithm: Design and Practice in Manufacturing. Springer Publishing Company, Berlin.
- [17] Information Resources Management Association (2016) Nature-Inspired Computing: Concepts, Methodologies, Tools, and Applications. IGI Global.
- [18] Dorigo, M., Birattari, M. and Stützle, T. (2006) Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique. *IEEE Computational Intelligence Magazine*, **1**, 28-39. <https://doi.org/10.1109/MCI.2006.329691>
- [19] Tuba, M. and Jovanovic, R. (2009) An Analysis of Different Variations of Ant Colony Optimization to the Minimum Weight Vertex Cover Problem. *Transactions on Information Science and Applications*, **6**, 936-945.
- [20] Bullnheimer, B., Hartl, R.F. and Strauß, C. (1997) A New Rank-Based Version of the Ant System Computational Study. *Central European Journal for Operations Research and Economics*, **7**, 25-38.
- [21] Stützle, T. and Hoos, H.H. (2000) MAX-MIN Ant System. *Future Generation Computer Systems*, **16**, 889-914. [https://doi.org/10.1016/S0167-739X\(00\)00043-1](https://doi.org/10.1016/S0167-739X(00)00043-1)
- [22] Dorigo, M. and Gambardella, L. (1996) A Cooperative Learning Approach to the Traveling Salesman Problem. *IEEE Transactions on Evolutionary Computation*, **1**, 53-66. <https://doi.org/10.1109/4235.585892>
- [23] Hu, X.M., Zhang, J. and Li, Y. (2008) Orthogonal Methods Based Ant Colony Search for Solving Continuous Optimization Problems. *Journal of Computer Science and Technology*, **23**, 2-18. <https://doi.org/10.1007/s11390-008-9111-5>
- [24] Gupta, D.K., Arora, Y., Singh, U.K. and Gupta, J.P. (2012) Recursive Ant Colony Optimization for Estimation of Parameters of a Function. *International Conference on Recent Advances in Information Technology*, Dhanbad, 15-17 March 2012, 448-454. <https://doi.org/10.1109/RAIT.2012.6194620>
- [25] Teodorovic, D. and Dell'orco, M. (2005) Bee Colony Optimization—A Cooperative Learning Approach to Complex Transportation Problems. *Proceedings of the 16th Mini-EURO Conference on Advanced OR and AI Methods in Transportation*, Poznan, 13-16 September 2005, 51-60.
- [26] Seeley, T.D. (1995) The Wisdom of the Hive. Harvard University Press, Cambridge.
- [27] Karaboga, D. (2005) An Idea Based on Honey Bee Swarm for Numerical Optimization.

- [28] Żak, J., Hadas, Y. and Rossi, R. (2017) Advanced Concepts, Methodologies and Technologies for Transportation and Logistics. Springer, Berlin.
- [29] Melin, P., Castillo, O. and Kacprzyk, J. (2015) Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization. Springer, Berlin. <https://doi.org/10.1007/978-3-319-17747-2>
- [30] Karaboga, D. and Basturk, B. (2007) Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems. *Foundations of Fuzzy Logic and Soft Computing*, Cancun, 18-21 June 2007, 789-798. https://doi.org/10.1007/978-3-540-72950-1_77
- [31] Luo, R., Pan, T., Tsai, P. and Pan, J. (2010) Parallelized Artificial Bee Colony with Ripple-Communication Strategy. *4th International Conference International Genetic and Evolutionary Computing*, Shenzhen, 13-15 December 2010, 350-353.
- [32] Akay, B. and Karaboga, D. (2010) A Modified Artificial Bee Colony Algorithm for Real-Parameter Optimization. *Information Science*, **192**, 120-142. <https://doi.org/10.1016/j.ins.2010.07.015>
- [33] Tsai, P.-W., Pan, J.-S., Liao, B.-Y. and Chu, S.-C. (2009) Enhanced Artificial Bee Colony Optimization. *International Journal of Innovative Computing, Information and Control*, **5**, 5081-5092.
- [34] Zou, W., Zhu, Y., Chen, H. and Sui, X. (2010) A Clustering Approach Using Cooperative Artificial Bee Colony Algorithm. *Discrete Dynamics in Nature and Society*, **2010**, Article ID: 459796. <https://doi.org/10.1155/2010/459796>
- [35] Parvanov, A.P. (2016) Handbook on Computational Intelligence.
- [36] Yang, X.-S. (2010) Nature-Inspired Metaheuristic Algorithms. Luniver Press.
- [37] Oscar, C. and Patricia, M. (2014) Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics: Theory and Applications. Springer, Berlin.
- [38] Shandilya, S.K., Shandilya, S., Deep, K. and Nagar, A.K. (2017) Handbook of Research on Soft Computing and Nature-Inspired Algorithms. IGI Global. <https://doi.org/10.4018/978-1-5225-2128-0>
- [39] Pérez, J.H., et al. (2014) A New Bat Algorithm with Fuzzy Logic for Dynamical Parameter Adaptation and Its Applicability to Fuzzy Control Design. Springer, Cham.
- [40] Rizk-Allah, R.M. and Hassanien, A.E. (2017) New Binary Bat Algorithm for Solving 0-1 Knapsack Problem. *Complex & Intelligent Systems*, 1-23. <https://doi.org/10.1007/s40747-017-0050-z>
- [41] Yang, X.-S. (2013) Bat Algorithm: Literature Review and Applications. *International Journal of Bio-Inspired Computation*, **5**, 141-149. <https://doi.org/10.1504/IJBIC.2013.055093>