



A critical discussion into the core of swarm intelligence algorithms

Dávila Patrícia Ferreira Cruz¹ · Renato Dourado Maia² · Leandro Nunes De Castro¹

Received: 8 June 2018 / Revised: 24 November 2018 / Accepted: 4 February 2019 / Published online: 4 March 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

The literature is now filled with swarm intelligence algorithms developed by taking inspiration from a number of insects and other animals and phenomena, such as ants, termites, bees, fishes and cockroaches, to name just a few. Many, if not most, of these bioinspirations carry with them some common issues and features which happen at the individual level, promoting very similar collective emergent phenomena. Thus, despite using different biological metaphors as inspiration, most algorithms present a similar structure and it is possible to identify common macro-processes among them. In this context, this paper identifies a set of common features among some well-known swarm-based algorithms and how each of these approaches implement them. By doing this, we provide the community with the core features of swarm-intelligence algorithms. This diagnostic is crucial and timely to the field, because once we are able to list and explain these commonalities, we are also able to better analyze and design swarm intelligence algorithms.

Keywords Swarm intelligence · Bio-inspired algorithm · Social insect · Meta-heuristic

1 Introduction

In the last decades, many new bioinspired algorithms have been proposed in the literature. Most of them, especially the swarm intelligence (SI) based algorithms, have gained popularity due to their capability of solving complex problems. Swarm intelligence aims to design algorithms with inspiration in the collective and intelligent behavior of insects and other animals, characterized by a structured collection of agents with limited individual capabilities interacting and exhibiting a collective intelligent behavior able to solve complex problems [17, 25, 47]. Examples of inspiration for SI are the collective behavior of birds, fishes, worms and, most commonly, social insects.

Social insects have gained attention, both from biologists and computer scientists, and are a rich inspirational source for the design and development of SI, as well as to provide a better understanding of biological phenomena that govern the fascinating abilities of social insects. Individually, social insects have sophisticated cognition and the capability of showing vast behavioral repertoires. In a collective level, the insects are able to solve complex problems collaboratively in a distributed and parallel way and without the need of a central control [11, 31, 49].

A vast number of SI algorithms, mainly those based on social insects, have been proposed in the literature over the past decades. Examples of well-known swarm intelligence algorithms are the Ant Colony Optimization (ACO) [22], the Particle Swarm Optimization (PSO) [27], and the Bee Colony Optimization (BCO) [59, 60], but there are many variations of these and other more recent approaches, such as the Bat Algorithm [65] and the Firefly Algorithm [29]. Some broad and recent reviews of swarm intelligence algorithms are presented in [34, 47, 67].

The main focus of the research on SI are applications to solve complex problems, the design of new approaches, and the improvement of existing ones [47, 67]. Biological metaphors have inspired the design of new metaheuristics, but in many cases the proposals do not follow the necessary scientific rigor, and this has been causing criticisms and

✉ Dávila Patrícia Ferreira Cruz
davilapatrícia11@gmail.com

Renato Dourado Maia
renato.dourado@unimontes.br

Leandro Nunes De Castro
lnunes@mackenzie.br

¹ Graduate Program in Electrical Engineering and Computing, Mackenzie University, São Paulo, Brazil

² Computer Science Department, State University of Montes Claros, Montes Claros, MG, Brazil

damage to the area [19, 28, 57]. In this scenario, it becomes necessary a more thorough study about the motivations to propose a new algorithm, as well as what contributions they are actually bringing to the scientific and applied research communities.

Despite using different biological metaphors as inspiration, most algorithms present a similar structure and it is possible to identify common macro-processes among them. Thus, the SI algorithms, as well as their bio-inspirations, can be viewed from different perspectives and different abstraction levels by focusing on their key components and features. In this context, this paper aims to identify a set of common features among the well-known swarm-based algorithms, especially those inspired by the collective behavior of social insects, and how each of these approaches implement them. By doing this, we provide the community with the core features of swarm-intelligence algorithms, facilitating the processes of understanding and implementing such methods in a more structured way. In our point of view, this effort represents an initial step towards the restructuring of SI, as well as its strengthening as a field of research.

This paper is organized as follows. Section 2 provides a brief explanation about the swarm intelligence research field and presents an overview of some well-known SI algorithms inspired by social insect behavior, together with some recent proposals. Section 3 provides an analysis of swarm-based algorithms identifying their macro-processes and key components and Sect. 4 presents the conclusions and future trends.

2 Swarm intelligence algorithms: an overview

Natural computing aims to provide a better understanding of the world in terms of information processing, making it possible to investigate, model, abstract and apply this knowledge in different contexts [17, 19, 38]. Swarm intelligence (SI) is a subfield of natural computing and takes inspiration in the collective behavior of insects and other social animals to design problem solving algorithms. It is characterized by a structured collection of agents with limited individual capabilities interacting and exhibiting a collective intelligent behavior [16, 17, 25].

Initially, the term swarm intelligence was used by Beni and Wang [6], in cellular robotic systems, to define a collection of autonomous, non-synchronized, non-intelligent robots cooperating to achieve a global goal. Bonabeau et al. [9] extended their definition to include any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies. A swarm intelligence system consists of a collection of agents with limited individual capabilities,

but whose societies are able to present intelligent collective behaviors [12]. Most SI works are focused on studies about social insects, such as bees, ants and wasps, mainly because social insects develop complex and emergent colony-level behaviors from, relatively, simple individual behaviors. They are able to make decisions by weighing many factors, sharing information and having cognitive abilities, such as memory, which allow them to hone their decisions [30].

Several approaches inspired by social insects' behaviors have been proposed in the literature to solve different problems [34, 47]. Examples of algorithms inspired by the collective behavior of social insects include the Ant Colony Optimization (ACO) [9], the Artificial Bee Colony (ABC) [36, 37], and the Firefly Algorithm (FA) [29, 65]. Among these, several well-known swarm intelligence approaches inspired by other social organisms are presented in the literature, such as the Particle Swarm Optimization (PSO) [27], Bacterial Foraging Optimization (BFO) [48], Slime Mold Optimization (SMOA) [43], and the Bat Algorithm [65]. To build our discussion on the commonalities of swarm intelligence algorithms, we provide an overview of three well-known algorithms from the literature (Ant Colony Optimization, Bee Colony Optimization and Ant Clustering Algorithm) and two more recent approaches (Spider Swarm and the Firefly Algorithm).

2.1 Ant colony optimization

Many researches demonstrate the ability of the ants to exploit rich food sources without losing the ability to explore the environment. In nature, many ant species communicate by means of a pheromone trail when foraging [68]. The individual ants, while moving from a food source to their nest, deposit a chemical substance, called pheromone, in the path. The pheromone trail provides the other ants information about the quality and location of the food source found. This collective trail-laying and trail-following behavior, where an ant is influenced by a chemical trail left by other ants, is an example of *stigmergy* and is the core of the Ant Colony Optimization (ACO) algorithms [9, 25].

The ACO metaheuristic, proposed by Dorigo and Caro [22], is inspired by the collective behavior of real ants and manipulates a collection of artificial ants that randomly explore the search space (environment), looking for promising regions that represent high quality solutions. Most of its applications has been on the solution of combinatorial optimization problems represented as graphs. The ants evaluate the built solutions and deposit pheromone on a connection of the graph proportionally to the quality of the respective solution. In the pheromone evaporation process, the deposited pheromone decreases over time. This process is important to avoid a premature convergence toward a local optimal region and favors the exploration of the search space [24].

The ACO metaheuristic was initially applied to solve the traveling salesman problem (TSP) [23, 26] and, over the past decades, many applications have been proposed in the literature, such as scheduling [7, 13], vehicle routing [4, 52], and feature selection [58].

2.2 Artificial bee colony

Bees of the *Apis Mellifera* species fly around the hive looking for sources of food and water. The evaluation of the sources by a bee is individual and takes into account many factors, such as the amount and quality of food, the distance to the hive, risks of predators, etc. Animal behavior researches show that the evaluation of the sources is a subjective judgment and that bees work with limited information (local information). However, in a colony level, they are able to adjust the exploration effort according to the quality of the sources, as well as to abandon the unproductive sources [55]. When a bee chooses a food source, it returns to the hive and performs a series of movements, known as waggle dance, that communicate to the other bees the features of the food source found. The first studies to understand this communication mechanism of bees was presented by Von Frisch [62]. The dance is based on the subjective evaluation of the bee. This mechanism is used to transmit information about the food sources and attract other bees to explore the sources found. The adjustment of the foraging effort of the colony is related to the quality of the food source found: the higher the quality of the food source, the higher the intensity of the dance, and the higher the number of bees attracted to exploit the source [15, 42, 55]. The collective behavior of bee colonies and their foraging abilities are inspiration for solving computational problems. Several approaches inspired by the collective behavior of bee colonies are presented in the literature, such as the Artificial Bee Colony (ABC) [36], Bee Colony Optimization (BCO) [60], Artificial Beehive [45] and OptBees [41].

The Artificial Bee Colony (ABC) algorithm was originally designed to solve optimization problems in continuous environments. In ABC, the artificial colony is composed of three groups of artificial bees: employed bees; onlookers; and scouts. The employed bees are associated with specific food sources, onlooker bees watch the dance of employed bees within the hive to choose a food source, and scout bees randomly search for food sources. A food source represents a candidate solution to the problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution [3, 36]. The ABC approach has been widely studied and applied to solve real-world problems [37], for example, signal, image and video processing [1, 5], and clustering problems [35, 46]. A comprehensive survey of ABC applications is presented in Karaboga et al. [37].

2.3 Ant clustering algorithm

Some species of ants clean up their nests by forming piles of corpses (ants' cemeteries). Experiments have been performed to study and understand the organization of cemeteries by ants, as reported by Deneubourg et al. [21]. The basis of the clustering phenomenon observed in ants is the attraction between dead items. The corpses found by ants are picked up and dropped at locations where more similar items are present. The attraction increases as the cluster of corpses grow [9, 16].

This observation has also led to the design of new ant-inspired algorithms. The Ant Clustering Algorithm (ACA) is based on the cemetery formation task observed in some species of ants [18], initially applied to solve clustering problems [20], and then adapted to graph partitioning [39], text mining [51] and other applications. A colony of artificial ants perform random walks in a bi-dimensional grid, or matrix, where the input data or objects are indexed randomly. The grid represents the environment of the ants and the objects represent the corpses to be grouped. Each cell of the grid stores the object's position. Initially, the objects and artificial ants are disposed randomly on the grid. The ants have a neighborhood radius that allow them to see everything in their neighborhood and, probabilistically, pick up or drop the objects over the grid taking into account the similarities between objects. Thus, isolated objects tend to be picked up, moved around and dropped close to other similar objects [9, 16].

2.4 Spider swarm optimization

Studies of animal behavior have documented a few spider species that exhibit social behavior [40]. The spider, like other insects, can be classified based on the level of cooperative behavior in two classes: solitary spiders and social spiders. The social spiders live in a colony composed of the members (male and female spiders) and the communal web. In a colony, the females are the majority, representing approximately 80% of the members [40, 54]. The communication among the colony members can be direct or indirect. The direct communication is performed by the exchange of fluids, also known as trophallaxis, or by body contact. An example of direct communication is the mating behavior. In indirect communication the environment is used as medium. The indirect interaction consists of small vibrations of the web that are perceived by other neighboring spiders. The communication by web vibration is used to transmit information about the risks of predators, mating possibilities, neighbor features, among others [40, 54].

Inspired by the collective behavior of social spiders, Cuevas et al. [14] recently presented a swarm intelligence algorithm, named Social Spider Optimization (SSO), to solve

optimization problems. SSO manipulates a population of artificial spiders (agents) composed of two types of spiders, males and females. Each spider assumes a set of tasks according to its gender. The SSO assumes that the entire search space is a *communal web*, where all social spiders interact with each other. The spider's positions in the communal web represent the candidate solutions. Each spider in the communal web receives a weight, w_i , according to its fitness value. The higher the spider's weight, or size, the higher the quality of the solution represented by it. In the initialization phase, the population of spiders is randomly initialized and a heuristic is used to determine the number of males and females. The females present an attraction or repulsion behavior over other spiders, according to the vibration perceived in the communal web. The communication between the spiders is performed by means of small vibrations on the communal web, which depend on the weight (fitness) and distance of the spider that generated it. Thus, the vibration perceived by spider i is a result of the information transmitted by spider j .

The spider movement can be of attraction or repulsion and takes into account three elements: (1) the movement in relation to the nearest spider; (2) the movement in relation to the best spider (best fitness); and (3) a random movement. On the other hand, males are divided in two groups, dominant and non-dominant. Dominant males represent those with better fitness and are attracted to other spiders for mating. The non-dominant males are attracted to the center of the males group. In real social spider colonies, mating occurs between dominant male and female spiders. The mating consists of building new candidate solutions by combining the solutions represented by the dominant males and females. This process contributes to the exploitation of the search space in order to find better individuals.

2.5 Firefly algorithm

In nature, fireflies (*Coleoptera: Lampyridae*) produce short and rhythmic flashes that compose a specific pattern for a particular species. The fireflies use their flashing light, produced by biochemical bioluminescence processes, for two main functions: (1) attract mating partners; and (2) warn potential predators. The light emitted has an intensity in a certain distance and decreases as the distance increases. Despite the light intensity being inversely proportional to the distance, the fireflies can be visible in a large enough distance for the communication: approximately 100 m at night.

The firefly communication by the flashing of light was the inspiration to design firefly-inspired algorithms, initially applied to solve optimization problems [29, 64, 66]. The Firefly Algorithm (FA) has been applied to solve many problems, such as optimization, data mining, image processing and different engineering applications [56, 64, 69]. In FA the

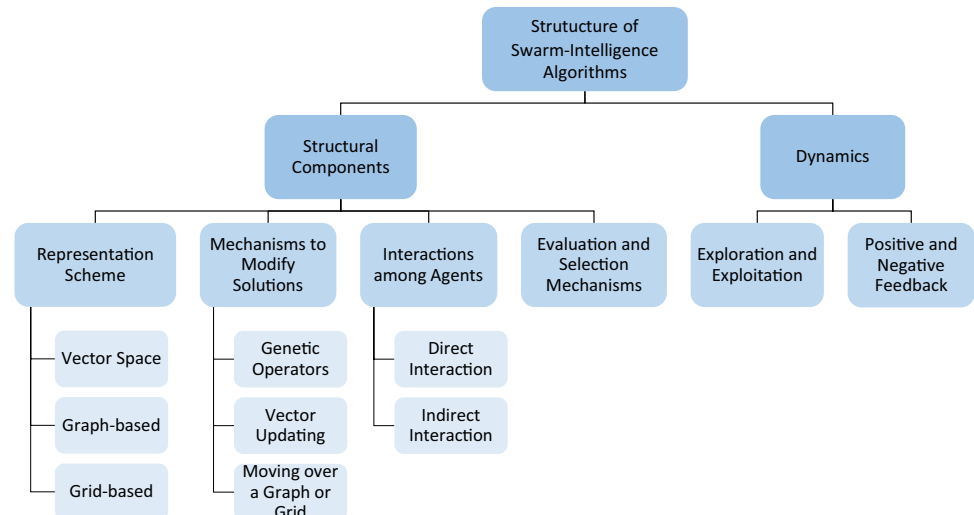
light intensity is associated with the objective function to be optimized. Three aspects, inspired by the natural metaphor, are important in the firefly algorithm: (1) the fireflies are unisex, thus they attract mating partners regardless of their gender; (2) their attractiveness is proportional to their flash light intensity (fitness value); and (3) the landscape of the fitness function can affect the light intensity [29, 64]. Each firefly location in the search space represents a solution of the optimization problem as a vector $\mathbf{x} = [x_1, \dots, x_d]$, where d is the dimension of the problem. The initial population of n fireflies is randomly generated and each firefly emits a flash of light to attract other fireflies. The light intensity I_i of firefly \mathbf{x}_i is determined by a fitness function $f(\mathbf{x}_i)$, $i = 1, 2, \dots, n$. The attractiveness varies with the distance r_{ij} between firefly i and firefly j . Thus, a firefly i is attracted to another more attractive (brighter) firefly j ; that is, one with higher fitness. Fister et al. [29] present a comprehensive review of the firefly algorithms and classify them according to some common aspects, such as the encoding schemes of fireflies, the population (swarm) setup, fitness function and type of firefly movements.

3 Structural components and dynamics of swarm intelligence algorithms

A vast number of nature-inspired algorithms, mainly those based on social insects, have been proposed in the literature over the past 2 decades. The swarm-based algorithms, normally used to solve complex optimization problems, are population-based algorithms [50] that apply an iterative process to construct and improve solutions. Despite using different biological metaphors as inspirations, most algorithms present a similar structure and it is possible to identify macro-processes in common among them.

This section provides an analysis of insect-inspired algorithms by taking into account elements related to their *structural components* and *dynamics*, as summarized in Fig. 1. As archetypes, we take the five approaches revised previously (ACO, BCO, SSO, FA and ACO) and argue that the proposal can be extended to most, if not all, insect-inspired methods available in the literature. We analyze the algorithms detaching their structural commonalities and the goal is to identify macro-processes and key components so as to propose an analytical and design framework for such methods. The algorithms are described taking into account the solution representation schemes, the mechanisms to generate and modify solutions, and the mechanisms to evaluate and select the solutions created. We analyze two important components of the dynamics of SI algorithms: the exploration/exploitation process; and the negative/positive feedback. Figure 1 presents the structure of insect inspired algorithms based on their structural components and dynamics.

Fig. 1 Structural components and dynamics of insect-inspired algorithms



3.1 Representation scheme

SI algorithms rely on the principles of multi-agent systems [53, 63]. The candidate solutions can be represented by an agent (position of the agent in the search space); a group of agents, where each agent represents part of the solution; or be built by the agents that move about a search space building the solution along the iterations.

The most common types of problems tackled by insect-inspired algorithms are optimization, data analysis, and image and signal processing [34, 47]. To solve them, the first step is to define how a candidate solution, which corresponds to one or more insects of the colony, or the problem to be solved, is going to be represented. There are basically three main types of representation or encoding schemes for candidate solutions or problems:

- **Vector space** In this type of representation, each candidate solution is represented by a string of values (e.g. bits, integers or floating-point numbers), which map, for instance, the position of a candidate solution in the search space of a continuous optimization problem ($\mathbf{x} = [x_1, x_2, \dots, x_n]$, $x_i \in \mathfrak{R}$, $\forall i = 1, 2, \dots, n$), or a given sequence of points to be moved through in a combinatorial optimization task ($\mathbf{y} = [y_1, y_2, \dots, y_m]$, $y_j \in \mathbb{Z}$, $y_j \neq y_i$, $\forall j \neq i$, $i, j = 1, 2, \dots, m$). These schemes were mainly borrowed from evolutionary algorithms [2] and are by far the most commonly found in the literature.
- **Graph-based** Many combinatorial optimization problems, such as the Traveling Salesman Problem (TSP) and the like (e.g. vehicle routing and scheduling), can be represented as a graph $G = \langle V, E \rangle$, which is characterized by a set of nodes $V = \{v_0, v_1, \dots, v_N\}$ and edges $E = \{(v_i, v_j): i \neq j\}$ connecting the nodes. For instance, in the TSP each node represents a city and each edge corre-

sponds to a path connecting a city (node) to another city (node). In such representation the agents move from one node to another until all nodes have been traversed and a full candidate solution to the problem (path connecting all cities) is proposed. Note that this representation is significantly different from the previous one, because the agents neither are candidate solutions, nor represent the problem themselves. Instead, they build a solution by moving about the graph and defining a path connecting the nodes. Therefore, the graph represents the problem, and the agents (insects) build the solution over the graph representation.

- **Grid-based** Some specific types of problem, for instance data clustering tasks, can be represented as a grid in which each position of the grid may host one or more data objects, and clusters of objects represent clusters of data. In this case, a grid is a regular tessellation of cells over a, usually, 2D surface divided into a series of contiguous cells. The agents (insects) are responsible for generating a solution to the problem by moving about the grid modifying the positions of the objects, until a satisfactory distribution of the objects in the grid is found. Similarly to the previous case, the agents themselves do not represent the candidate solutions, but, instead, they build a candidate solution to the problem by finding a suitable configuration of objects over the grid.

In the swarm intelligence context, Dorigo and collaborators [22, 25], proposed the ACO algorithm, and were pioneers in introducing a graph-based representation for solving the TSP and other combinatorial optimization problems using swarm intelligence. In this case, the problem is represented as a graph where each node corresponds to a city, and each edge is a path from one city to another. The artificial ants build the solutions by visiting all cities sequentially.

With a completely different approach, the ABC, SSO and FA use the vector space representation scheme for solving continuous optimization problems. Given a D -dimensional continuous optimization problem, a candidate solution is given by $\mathbf{x} = [x_1, x_2, \dots, x_D]$, $x_i \in \mathcal{R}, \forall i$. Thus, to solve continuous optimization problems the solutions are represented by a vector that contains the position of the agent (e.g., artificial bee, spider or firefly) in the search space. On the other hand, the ACA uses the grid-based representation scheme for solving data clustering problems. In ACA the agents build the solutions incrementally by moving the objects over a two-dimensional grid.

3.2 Mechanisms to modify solutions

Once a representation scheme (candidate solutions) is defined, the next step is to determine how these candidate solutions are going to be manipulated to generate new candidate solutions and, thus, promote a movement in the search space. This is another crucial step, because the more efficient the solution modification method, the more efficient the algorithm tend to be. Also, whilst the representation is defined by looking almost exclusively at the problem to be solved, the way solutions are manipulated is intimately related with the biological metaphor and the encoding scheme chosen. Therefore, at least in principle, it is the biological inspiration that is going to guide the search over the space.

In most insect-inspired algorithms, new solutions can be generated randomly or by using a heuristic that combines existing solutions to obtain a novel one. In this direction, the most common biological metaphors and, thus, mechanisms, to modify solutions are:

- *Genetic operators* These are borrowed from evolutionary algorithms and basically include *crossover* (specific recombination of two or more candidate solutions) and *mutation* (a random variation in a single existing candidate solution). There are many ways in which crossover and mutation can be implemented and these depend on the metaphor, encoding scheme, and problem to be solved. Some examples will be given in the following for the algorithms reviewed.
- *Vector updating* The use of a vector-space representation allows specific types of modification schemes, but two of them are the most common ones. For floating-point representations, a typical updating method is performed based on the following equation: $\mathbf{x}(k) = \mathbf{x}(k-1) + \Delta\mathbf{x}(k)$, where $\Delta\mathbf{x}(k)$ is the value to be added (or subtracted) to the current value of \mathbf{x} (candidate solution), and k represents the time step. For integer strings, where a candidate solution is a permu-

tation of integers, there are several specific operators that can be used for generating new permutations from the existing ones.

- *Moving over a graph or a grid* The moving of an agent over a graph or a grid is a way to modify the solution in a graph-based or a grid-based representation, respectively. The agents move through adjacent states of the problem by building solutions on a graph G or a grid. In a graph, at each iteration, each agent moves from a node y_i to a node y_j with a probability p or following a specific heuristic. In a grid, at each iteration, each agent moves from a cell c_i to a cell c_j with a probability p or following a specific heuristic.

In ACO the candidate solutions are built incrementally by the ants' movements. The artificial ants move through adjacent states of the problem by building paths on a weighted graph G , which represent the candidate solutions. At each iteration, each ant moves from a node y_i to a node y_j . This movement depends basically on two parameters: the amount of pheromone on the edge; and the visibility of the following nodes. By moving over the graph the ants gradually build and modify the candidate solutions. In ABC, SSA and FA, despite the metaphor, the new candidate solutions are built in one of two ways: random movement of agents in the search space (random generation); or by moving an agent toward another that represents a better solution or following a specific heuristic (guided movement). Generally, the first case is applied to build new solutions and to explore the search space. The guided movement, in general, is implemented by means of vector updating, which modify existing solutions and contribute to the exploitation of promising regions found by some agents. In ACA the manipulation of candidate solutions are performed by the ants' movement over a two-dimensional grid. At each iteration, each ant moves randomly from a cell c_i to a cell c_j .

3.3 Interaction among agents

In nature, interactions represent the communication channel among insects and between them and the environment. The communication by means of interactions allows the insects to acquire information about the environment and the colony. The rules specifying these interactions are performed based on purely local information, without information about the entire colony. Thus, insects work with restrict information of their neighborhood, receiving social information by means of interactions, without having knowledge about the entire colony [8]. The interactions allow the insects to share and obtain information about the environmental and colony conditions. Interactions among insects can be direct or indirect [44]:

- *Indirect interaction* Indirect interactions are the communication among insects mediated by the environment, known as *stigmergy* [61]. Some individuals modify the environment and others perceive this modification, adjusting their behaviors accordingly [16].
- *Direct interaction* Consists of a local communication where no modification of the environment is required. The information exchanged by direct interactions can be of different types, such as physical contact, fluid exchange or visual and acoustic signs.

Besides that, the insect can interact directly with its environment. In this case, differently from stigmergy, the insects interact directly with the environment (perhaps not explored yet) to acquire new information. In SI algorithms this interaction occurs by means of random exploration of the search space looking for new good candidate solutions.

In ACO, stigmergy is implemented by the amount of pheromone in the graph's edges and its perception by other ants. In ACA the stigmergy communication is implemented by means of the modification of the grid. In the spider-inspired algorithms, the communication is mediated by the vibration on the web, which represents the spiders' environment. In both cases, the stigmergic communication transmits the social information about the quality of the solution found. The bee and firefly algorithms are inspired by direct interaction, more specifically, the transfer of messages by waggle dance bees and flying-mating behavior of fireflies, respectively. From an algorithmic perspective, the interaction among the individuals represent the way the information represented in a solution is shared or combined to create new solutions or modify an existing solution, as presented above.

The indirect interactions are implemented in ACO by means of updating the pheromone level of each edge and the movement of the agent over the graph. The agents modify the amount of pheromone on the graph's edges and this affects the probability of other agents choosing a path. In ABC, SSO and FA, the direct interactions among insects that conduct to attraction or recruitment of other insects are implemented by updating the solutions represented by these insects (vector updating). Besides that, in SSO the agent also interacts directly in the mating behavior, implemented by the recombination of solutions represented by two or more agents.

3.4 Evaluation and selection of solutions

The evaluation and selection mechanisms aim to assess the quality of candidate solutions and to select those more adapted to the problem.

The quality of a candidate solution, \mathbf{x} , is normally expressed by a fitness function, $f(\mathbf{x})$, which evaluates the adaptability of this candidate solution to the problem at

hand. In ACO algorithms, the candidate solution is a tour traveled by an ant, and the quality of this solution is related to the cost of the tour. In ABC, SSO and FA, the candidate solution is represented by the agent's position in the search space.

The selection mechanisms act on the candidate solutions and aim to select those more adapted to the problem based on a given criterion. The selected candidate solutions are submitted to mechanisms that modify them, for instance, by combining two candidate solutions to generate a new one (recombination operator), or a solution (or part of a solution) is modified following some heuristic (vector updating and moving of the agents).

In ACO, the solutions are built by means of a probabilistic heuristic used to move the agent taking into account the amount of pheromone on each graph's edge and the visibility of the other nodes on the graph. The selection mechanism in ACO, thus, helps ants to move through edges that form shorter routes. In ABC, the selection is based on the quality of the food source explored by the bees. In SSO, the quality of the solutions is related to a weight that represents the spider's size (fitness value) and, in FA, the quality of the solutions is related to the light intensity (fitness value) emitted by the fireflies.

Despite of different bio-inspirations, the essential idea of these mechanisms is to evaluate the suitability of the candidate solutions by means of a fitness function. Thus, the evaluation and selection of solutions represented by the agents is specific and appropriate for each problem.

Table 1 provides a summary of the structural components and dynamics of the swarm intelligence algorithms reviewed.

3.5 Exploration and exploitation

The algorithms reviewed are classified as search algorithms, where agents represent candidate solutions in a search space. According to Hill et al. [33], the search process involves a trade-off between exploiting known opportunities and exploring the search space. Global search allows the exploration of the entire search space and looks for new and better solutions for the problem. It can be random or guided by using information about the environment (search space). On the other hand, local search is used to exploit promising regions of the search space to improve the solutions found. The search space exploration and exploitation result in finding (or generating) new solutions or improving known solutions, respectively. The trade-off between exploration and exploitation is essential for an effective optimization metaheuristic.

In ACO, the exploration is made by means of iterative and probabilistic procedures to construct and modify solutions. These procedures consist of the formation and updating of

Table 1 Summary of the structural components of ACO, ABC, SSO and FA algorithms

Algorithm	Solution representation	Representation scheme	Mechanisms to modify solutions	Interaction among agents	Evaluation mechanisms
ACO	Tour taken by an ant (path traveled by the agent)	Graph-based	Moving over a graph	Indirect	Cost of route
ABC	Food source explored by bee (position of agent)	Vector space	Vector updating	Direct	Quality of region explored (agent position)
SSO	Position of spider on the communal web (position of agent)	Vector space	Vector updating and recombination operators	Direct and indirect	Size of spider (agent position)
FA	Region explored by firefly (position of agent)	Vector space	Vector updating	Direct	Light intensity (agent position)
ACA	Clustering performed by agents	Grid-based	Moving over a grid	Indirect	Clustering quality

the pheromone trail by the ants. Exploitation is performed by the recruitment of ants that tend to choose those paths with more pheromone, and the recruitment is performed indirectly by means of the pheromone trail.

In general, in algorithms inspired by the foraging behavior of honey bees, such as ABC, exploitation and exploration are balanced by means of the local search and global search performed by artificial bees. Exploration consists of bees randomly moving over the space, looking for new good solutions. The exploration of the search space is performed by the random fly of the drones. Exploitation relies on the recruitment processes when the bees are recruited to exploit a high-quality region assessed by other bees.

In the SSO algorithm, exploitation occurs by the attraction of spiders for mating processes, i.e., a spider is attracted by others that represent a better solution by means of the web vibration. Exploration consists of the random movement of spiders in the net. In FA, the attraction of fireflies by flashing lights emitted by other fireflies contribute to the search space exploitation. The attraction is proportional to the intensity of flashing lights that represent the quality of solutions. In ACA, the exploration consists of the ant randomly moving

over the grid looking for corpses, and exploitation occurs by means of ants' attraction for regions with greater accumulation of dead ants (data).

Despite the different metaphors that implement these mechanisms, in general they fall into the same computational procedure. In all cases presented here, the implementation of exploration and exploitation falls within one of the procedures to modify solutions, as presented in Sect. 3.2. In the SI algorithms presented here, exploration consists of a random search for new solutions, whilst exploitation consists of a guided, local search that moves the agents towards good solutions (or good regions) over the search space. The random search moves the agents along the search space and consequently modifies the candidate solutions according to their representation schemes, as shown in Sect. 3.2. Table 2 summarizes the representation schemes, exploration and exploitation mechanisms of the algorithms reviewed.

3.6 Positive and negative feedback

The concept of feedback is important in many biological and artificial systems. Positive feedback tends to amplify

Table 2 Representation, exploration and exploitation of some insect-inspired algorithms

Biological metaphor	Representation of solution	Exploration (random generation of a new solution)	Exploitation (recruitment—modification of an existent solution)
ACO	A route in a graph $G=(V, A)$	Random movement over the graph	Following paths with high concentration of pheromone and high visibility
ABC	Vector-space $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$	Random movement over the space	Attraction of bees to high quality regions of the search space
SSO	Vector-space $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$	Random movement over the space	Attraction of spider to high quality regions of the communal web
FA	Vector-space $x_i = [x_{i1}, x_{i2}, \dots, x_{id}]$	Random movement over the space	Movement of fireflies toward other brighter fireflies
ACA	Indexed items spread on a $n \times n$ grid	Random movement over the grid	Attraction of groups of items over the ant that is carrying an item

the response to an input stimulus, promoting the creation of structures. On the other hand, negative feedback works as a counterbalance that damps the response to input stimuli [32]. Feedback (positive and negative) is one of the basic components of self-organizing (SO) phenomena that, in turn, appears as major components of a wide range of collective behaviors presented by social animals and insects [10, 31].

In ACO, positive feedback is established by the reinforcement of pheromone trails by ants and negative feedback occurs by pheromone trail evaporation and a limited number of ants, which can lead to the abandonment of a food source or a maximum number of ants exploiting a food source, respectively. In ABC, positive feedback is represented by the attraction of bees based on the quality of food source and the negative feedback is represented by the abandonment of the food source when it does not bring more benefits. In SSO and FA, the positive and negative feedback can be related to the attraction and repulsion behaviors of individuals, respectively. In ACA positive feedback is established by the deposit of corpses that contribute to the formation of small clusters and negative feedback occurs by the reduction of ants available. In all cases, a limited number of agents results in a negative feedback mechanism.

Considering the metaphors presented, positive and negative feedback are better represented in the classical metaphors ACO, ACA and ABC. In these cases, the recruitment process by pheromone trail or waggle dance allows the insects to assess the food sources quality (fitness), attracting more insects and reinforcing exploitation. Despite of similar metaphors, the foraging behavior of bees and ants have an important difference: the type of interactions. The ants use indirect interactions that lead to the construction of a shared structure (pheromone trail), and bees perform direct interactions by observing the dance and follow the information shared. The first case is less flexible and its negative feedback acts more slowly, which can lead ants to neglect new sources that came later. The second case represents a more flexible behavior, i.e., bees can rapidly adjust the colony effort to exploit different food sources or abandon them. This feature is presented also in bees and ant-inspired algorithms. On the other hand, SSO and FA do not present an explicit description of negative and positive feedback as presented by ACO and ABC.

3.7 Some notes on the structure of swarm intelligence algorithms

As presented, swarm intelligence algorithms, more specifically those inspired by the collective behavior of social insects, share a similar structure when viewed from the same perspective and abstraction level. Despite using different biological metaphors as inspirations, these commonalities can be mapped and are important to identify the key

components and features that contribute to the emergence of collective behaviors.

In all cases presented here, the trade-off between exploitation and exploration is an important component for the dynamics of the algorithms. In general, exploitation consists of the use of existing social information (shared by other insects) and exploration consists of the collection of new information by means of exploratory behaviors. The candidate solutions can often be generated by the movement of agents through the search space. The movement can be random, contributing to the search space exploration and finding of new promising regions, or guided, by using a heuristic specific to the problem. The guided movement of agents contributes to the exploitation of regions in order to find better solutions.

The definition of the representation (encoding) scheme and mechanisms to the manipulation of solutions are crucial steps to more efficient algorithms. The more efficient the solution modification method, the more efficient the algorithm tends to be. These mechanisms are directly associated with the representation scheme used to represent the candidate solutions. Usually, algorithms with the same representation scheme have similar mechanisms to modify solutions. For example, algorithms with vector space representation, such as SSO and FA, implement similar operators to combine two or more solutions, and most of them are typical evolutionary (crossover and mutation) operators.

In population-based algorithms, the agents perform a set of tasks simultaneously. Each agent has local information about the region explored and, differently from the biological metaphors, the global information is available for use in some stage of the iterative process. The evaluation process in most swarm intelligence algorithms is performed by taking into account the global information.

Table 3 summarizes the analyzed aspects taking into account the biological metaphors.

As presented here, SI algorithms share a common structure independently of their swarm inspirations and some components are essential for an effective swarm-inspired metaheuristic. On the other hand, in some cases, besides of the different swarm-inspirations, these components basically implement the same computational procedures, as presented above. For example, the exploration process in all cases consists of the random movement of agents looking for new solutions, independently if this agent is a spider, ant, bee or firefly. Furthermore, the attraction of agents for more productive regions in the exploitation process is implemented by the movement of agents toward other agents that represent better solutions, independently if this attraction is based on the vibration of web or flash light, for example.

Table 3 Swarm-based algorithms components according to the biological metaphor

Biological Metaphor	Agents	Exploration	Exploitation	Evaluation system	Interactions type	Positive feedback	Negative feedback
Foraging behavior of ants	Ants	Random exploration of the environment	Exploration of food source indicated by more reinforced pheromone trail	Cost of the route	Indirect (pheromone trail)	Reinforcement of pheromone trail	Pheromone evaporation
Foraging behavior of bees	Recruiter and non-recruiter bees	Random exploration of the environment	Exploration of selected food source (recruiter and recruited)	Quality of the food source	Direct (waggle dance)	Bees attraction	Abandonment of food source
Communication of spiders	Male and female spiders	Random movement on the communal net	Attraction and crossover of male and female spiders	Spider's size	Indirect (vibration of communal web)	Spider attraction	Repulsion behavior
Communication of fireflies	Unisex fireflies	Random movement of firefly	Attractiveness of fireflies by light intensity	Light intensity	Direct (flashing light)	Firefly attraction	Repulsion behavior
Cemetery Formation	Ants	Random exploration of the environment	Attractiveness of ants by the accumulation of dead ants	Quality of clustering	Indirect (modification of environment)	Accumulation of dead ants and pheromone deposit	Pheromone evaporation

4 Discussion and future researches

Swarm intelligence is an active research field and has attracted the attention of many researches, especially engineers and computer scientists. Recent works in SI include the use of different biological metaphors as inspiration and the improvement of current approaches to the application in several real problems. A vast number of nature-inspired algorithms, mainly those based on social insects or social animals, have been proposed in the literature over the past 2 decades. Despite the many biological metaphors used as inspirations, it is possible to identify common components and similar structures among them, according to the point of view and the abstraction level. Many, if not most, bioinspirations carry with them some common issues and features, which happen in an individual level promoting very similar collective emergent phenomena.

Swarm-based algorithms share a common structure independently of their swarm inspirations. For example, any effective swarm-based algorithm has to achieve an appropriate balance between the exploitation of good regions and the exploration of unvisited search space regions. The trade-off between positive and negative feedback is important to the convergence of the algorithm. In addition, swarm-based algorithms must conform to the self-organizing principles. As a consequence, many of the algorithms proposed in the literature are basically reproducing the same computational procedures independently of the bioinspiration chosen.

The criticism about the SI research area has increased in recent years, but little effort has been expended toward the restructuration of the area and better communication between biology and computing. There are many types of swarms in nature, but just a few can be called intelligent. An effective metaphor to swarm intelligence must satisfy the principles of swarm intelligence behaviors, such as local rules without relation to the global patterns, interactions among self-organized agents and emergent collective behaviors. A more careful and rigorous view about the importance of the metaphors for the design of new algorithms is required. Following the biological principles, despite of this common structure, the metaphor can provide specific details to contribute to the design of new algorithms that really contribute to the advancement of the swarm intelligence area.

As discussed in this paper, swarm intelligence algorithms as well as bioinspired algorithms, can be viewed from different perspectives and different levels of abstraction by identifying their essential components and characteristics. In this context, as a continuity of this research it will be proposed a biological and computational framework with the objective of guiding researchers in the

process of analysis and synthesis of swarm intelligence algorithms. The framework will contribute to the analysis and development of these algorithms in a structured and well-founded way and it represents an important step towards the strengthening of swarm intelligence as a line of research.

Acknowledgements The authors thank Capes, CNPq, Fapesp and MackPesquisa for the financial support. The authors also thank Intel as the sponsor of LCoN as a Center of Excellence in Artificial Intelligence.

References

- Akay B, Karaboga D (2015) A survey on the applications of artificial bee colony in signal, image, and video processing. *SIViP* 9(4):967–990
- Bäck T, Fogel DB, Michalewicz Z (1997) Handbook of evolutionary computation. Release 97(1)
- Basturk B, Karaboga D (2006) An artificial bee colony (ABC) algorithm for numeric function optimization. *IEEE Swarm Intell Symp* 8(1):687–697
- Bell JE, McMullen PR (2004) Ant colony optimization techniques for the vehicle routing problem. *Adv Eng Inform* 18(1):41–48
- Benala TR, Villa SH, Jampala SD, Konathala B (2009) A novel approach to image edge enhancement using artificial bee colony optimization algorithm for hybridized smoothing filters. World Congress on Nature and Biologically Inspired Computing, 2009. NaBIC 2009
- Beni G, Wang J (1993) Swarm intelligence in cellular robotic systems. In: *Robots and biological systems: towards a new bionics?*, vol 102. Springer, Berlin, pp 703–712. https://doi.org/10.1007/978-3-642-58069-7_38
- Blum C, Sampels M (2004) An ant colony optimization algorithm for shop scheduling problems. *J Math Model Algorithms* 3(3):285–308
- Bonabeau E (1998) Social insect colonies as complex adaptive systems. *Ecosystems* 1(5):437–443
- Bonabeau E, Dorigo M, Theraulaz G (1999) *Swarm intelligence: from natural to artificial systems*, vol 1. Oxford University Press, Oxford
- Bonabeau E, Theraulaz G, Deneubourg J-L, Aron S, Camazine S (1997) Self-organization in social insects. *Trends Ecol Evol* 12(5):188–193
- Bouffanais R (2016) A biologically inspired approach to collective behaviors. In: *Design and control of swarm dynamics*. Springer, Singapore, pp 5–15
- Camazine S, Deneubourg J-L, Franks NR, Sneyd J, Theraulaz G, Bonabeau E (2003) *Self-organization in biological systems*. Princeton University Press, Princeton
- Chen W-N, Zhang J (2009) An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements. *IEEE Trans Syst Man Cybern Part C (Appl Rev)* 39(1):29–43
- Cuevas E, Cienfuegos M, Zaldívar D, Pérez-Cisneros M (2013) A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Syst Appl* 40(16):6374–6384
- Das P (1998) Success and the dancing bee. *Nature*. <https://doi.org/10.1038/news980813-8>
- De Castro LN (2006) Fundamentals of natural computing: basic concepts, algorithms, and applications. Chapman and Hall/CRC, São Paulo
- De Castro LN (2007) Fundamentals of natural computing: an overview. *Phys Life Rev* 4(1):1–36
- De Castro LN, Vizine AL, Hruschka ER, Gudwin RR (2005) Towards improving clustering ants: an adaptive ant clustering algorithm. *Informatica* 29(2):143–154
- De Castro LN, Xavier RS, Pasti R, Maia RD, Szabo A, Ferrari DG (2011) The grand challenges in natural computing research: the quest for a new science. *IJNCR* 2:17–30
- Deneubourg JL, Goss S, Franks N, Sendova-Franks A, Detrain C, Chrétiens L (1991) The dynamics of collective sorting: robot-like ants and ant-like robots. In: *Proceedings of the first international conference on simulation of adaptive behaviour: from animals to animats*. MIT Press, Cambridge, pp 356–365
- Deneubourg JL, Goss S, Franks N, Sendova-Franks A, Detrain C, Chretien L (1992) The dynamics of collective sorting: Robot-like ants and ant-like robots. In: *From animals to animats: proceedings of the first international conference on simulation of adaptive behavior*, pp 353–363
- Dorigo M, Caro GD (1999) Ant colony optimization: a new meta-heuristic. In: *Proceedings of the 1999 congress on evolutionary computation*, pp 1470–1477
- Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1(1):53–66
- Dorigo M, Stützle T (2003) The ant colony optimization metaheuristic: algorithms, applications, and advances. In: Glover F, Kochenberger GA (eds) *Handbook of metaheuristics*, vol 57. Springer, New York
- Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. *IEEE Comput Intell Mag* 1(4):28–39
- Dorigo M, Maniezzo V, Colomni A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern Part B (Cybernetics)* 26(1):39–41
- Eberhart R, Kennedy J (1995) A new optimizer using particle swarm theory. In: *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*. IEEE, pp 39–43
- Fister Jr I, Mlakar U, Brest J, Fister I (2016) A new population-based nature-inspired algorithm every month: is the current era coming to the end. In: *Proceedings of the 3rd student computer science research conference*. University of Primorska Press, pp 33–37
- Fister I, Yang X-S, Brest J (2013) A comprehensive review of firefly algorithms. *Swarm Evol Comput* 13:34–46
- Gadau J, Fewell J (2009) *Organization of insect societies: from genome to sociocomplexity*. Harvard University Press, Cambridge
- Garnier S, Gautrais J, Theraulaz G (2007) The biological principles of swarm intelligence. *Swarm Intell* 1(1):3–31
- Gordon DM (2016) The evolution of the algorithms for collective behavior. *Cell Syst* 3:514–520
- Hills TT, Todd PM, Lazer D, Redish AD, Couzin ID (2015) Exploration versus exploitation in space, mind, and society. *Trends Cogn Sci* 19(1):46–54
- Hussain K, Salleh MN, Cheng S, Shi Y (2018) Metaheuristic research: a comprehensive survey. *Artif Intell Rev* 13:1–43
- Ji J, Pang W, Zheng Y, Zhe Wang ZM (2015) A novel artificial bee colony based clustering algorithm for categorical data. *PLoS One* 10(5):1–17
- Karaboga D (2005) An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Kayseri
- Karaboga D, Gorkemli B, Ozturk K, Karaboga N (2014) A comprehensive survey: artificial bee colony (ABC) algorithm and applications. *Artif Intell Rev* 42(1):21–57
- Kari L, Rozenberg G (2008) The many facets of natural computing. *Commun ACM* 51(10):72–83

39. Kuntz P, Snyers D, Layzell P (1999) A stochastic heuristic for visualising graph clusters in a bi-dimensional space prior to partitioning. *J Heuristics* 5(3):327–351
40. Lubin Y, Bilde T (2007) The evolution of sociality in spiders. *Adv Study Behav* 37:83–145
41. Maia RD, De Castro LN, Caminhas WM (2013) Collective decision-making by bee colonies as model for optimization—the OptBees Algorithm. *Appl Math Sci* 7(87):4327–4351
42. Michener CD (1969) Comparative social behavior of bees. *Annu Rev Entomol* 14(1):299–342
43. Monismith D, Mayfield B (2008) Slime mold as a model for numerical optimization. In: *IEEE swarm intelligence symposium*. IEEE, pp 1–8
44. Moussaid M, Garnier S, Theraulaz G, Helbing D (2009) Collective information processing and pattern formation in swarms, flocks, and crowds. *Top Cogn Sci* 1(3):469–497
45. Muñoz MA, López JA, Caicedo E (2009) An artificial bee-hive algorithm for continuous optimization. *Int J Intell Syst* 24(11):1080–1093
46. Ozturk C, Hancer E, Karaboga D (2015) Dynamic clustering with improved binary artificial bee colony algorithm. *Appl Soft Comput* 28:69–80. <https://doi.org/10.1016/j.asoc.2014.11.040>
47. Parpinelli RS, Lopes HS (2011) New inspirations in swarm intelligence: a survey. *Int J Bio-Inspired Comput* 3(1):1–16
48. Passino K (2002) Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst Mag* 22:52–67
49. Perry CJ, Barron AB, Chittka L (2017) The frontiers of insect cognition. *Curr Opin Behav Sci* 16:111–118
50. Prugel-Bennett A (2010) Benefits of a population: five mechanisms that advantage population-based algorithms. *IEEE Trans Evol Comput* 14(4):500–517
51. Ramos V, Merelo JJ (2002) Self-organized stigmergic document maps: environments as a mechanism for context learning. In: *Proceedings of the 1st Spanish conference on evolutionary and bio-inspired algorithms*, Mérida, pp 284–293
52. Rizzoli AE, Montemanni R, Lucibello E, Gambardella LM (2007) Ant colony optimization for real-world vehicle routing problems. *Swarm Intell* 1(2):135–151
53. Russell S, Norvig P (2004) *Inteligência artificial*. Elsevier, Amsterdam
54. Salomon M, Sponarski C, Larocque A, Avilés L (2010) Social organization of the colonial spider *Leucauge* sp. in the Neotropics: vertical stratification within colonies. *J Arachnol* 38(3):446–451
55. Seeley TD, Camazine S, Sneyd J (1991) Collective decision-making in honey bees: how colonies choose among nectar sources. *Behav Ecol Sociobiol* 28(4):277–290
56. Senthilnath J, Omkar S, Mani V (2011) Clustering using firefly algorithm: performance study. *Swarm Evol Comput* 1(3):164–171. <https://doi.org/10.1016/j.swevo.2011.06.003>
57. Sörensen K (2015) Metaheuristics—the metaphor exposed. *Int Trans Oper Res* 22(1):3–18
58. Tabakhi S, Moradi P, Akhlaghian F (2014) An unsupervised feature selection algorithm based on ant colony optimization. *Eng Appl Artif Intell* 32:112–123
59. Teodorovic D (2009) Bee colony optimization (BCO). *Innov Swarm Intell* 248:39–60
60. Teodorovic D, Dell’Orco M (2005) Bee colony optimization—a cooperative learning approach to complex transportation problems. *Advanced OR and AI methods in transportation*, pp 51–60
61. Theraulaz G, Bonabeau E (1999) A brief history of stigmergy. *Artif Life* 5(2):97–116
62. Von Frisch K (1967) *The dance language and orientation of bees*. Belknap Press, Cambridge
63. Weiss G (1999) *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, Cambridge
64. Yang X-S (2009) Firefly algorithms for multimodal optimization. In: *International symposium on stochastic algorithms*. Springer, pp 169–178
65. Yang X-S (2010) A new metaheuristic bat-inspired algorithm. In: *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pp 65–74
66. Yang X-S (2010) Firefly algorithm, Levy flights and global optimization. In: *Research and development in intelligent systems XXVI*, pp 209–218
67. Yang X-S, Deb S, Fong S, He X, Zhao Y (2016) Swarm intelligence: today and tomorrow. In: *Third international conference on soft computing and machine intelligence (ISCM)*. IEEE, pp 219–223
68. Yew JY, Chung H (2015) Insect pheromones: an overview of function, form, and discovery. *Prog Lipid Res* 59:88–105
69. Zhang Y-D, Wu L (2012) A novel method for rigid image registration based on firefly algorithm. *Int J Res Rev Soft Intell Comput (IJRRSIC)* 2(2):141–146

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.