# Chapter 4
# Bee Inspired Algorithms

**Abstract** In this chapter, we present a set of algorithms that are inspired by different honeybees behavioural patterns, i.e., artificial bee colony (ABC) algorithm, honeybees mating optimization (HBMO) algorithm, artificial beehive algorithm (ABHA), bee colony optimization (BCO) algorithm, bee colony inspired algorithm (BCiA), bee swarm optimization (BSO) algorithm, bee system (BS) algorithm, BeeHive algorithm, bees algorithm (BeA), bees life algorithm (BLA), bumblebees algorithm, honeybee social foraging (HBSF) algorithm, OptBees algorithm, simulated bee colony (SBC) algorithm, virtual bees algorithm (VBA), and wasp swarm optimization (WSO) algorithm. We first describe the general knowledge of honeybees in Sect. 4.1. Then, the fundamentals and performances of these algorithms are introduced in Sects. 4.2, 4.3, and 4.4, respectively. Finally, Sect. 4.5 summarises in this chapter.

## 4.1 Introduction

Honeybee is a typical social insect that works together in a highly structured social order to finish different kinds of jobs such as bee dance (communication), bee foraging, queen bee, task selection, collective decision making, nest site selection, mating, floral/pheromone laying, and navigation that have long since attracted the attention of human beings (Janson et al. 2005; Latty et al. 2009; Slaa and Hughes 2009; Landa and Tullock 2003). Based on those features, many models have been developed for intelligent systems and applied to solve combinatorial optimization problems. In this chapter, a set of bee inspired algorithms are collected and introduced as follows:

- Section 4.2: Artificial Bee Colony.
- Section 4.3: Honeybees Mating Optimization.
- Section 4.4.1: Artificial Beehive Algorithm.
- Section 4.4.2: Bee Colony Optimization.

B. Xing and W.-J. Gao, *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*, Intelligent Systems Reference Library 62,
DOI: 10.1007/978-3-319-03404-1_4, 

- Section 4.4.3: Bee Colony-inspired Algorithm.
- Section 4.4.4: Bee Swarm Optimization.
- Section 4.4.5: Bee System.
- Section 4.4.6: BeeHive.
- Section 4.4.7: Bees Algorithm.
- Section 4.4.8: Bees Life Algorithm.
- Section 4.4.9: Bumblebees Algorithm.
- Section 4.4.10: Honeybee Social Foraging.
- Section 4.4.11: OptBees.
- Section 4.4.12: Simulated Bee Colony.
- Section 4.4.13: Virtual Bees Algorithm.
- Section 4.4.14: Wasp Swarm Optimization.

The effectiveness of these newly developed algorithms are validated through the testing on a wide range of benchmark functions and engineering design problems, and also a detailed comparison with various traditional performance leading computational intelligence (CI) algorithms, such as particle swarm optimization (PSO), genetic algorithm (GA), differential evolution (DE), evolutionary algorithm (EA), fuzzy system (FS), ant colony optimization (ACO), and simulated annealing (SA).

### *4.1.1 Foraging Behaviour of Bees*

Among other algorithms inspired by honeybees, probably one of the most noticeable behaviours visible to us is the foraging of each individual bee. Foraging process includes two main modes of behaviour: recruitment of nectar source and abandonment of a source (Tereshko and Lee 2002). It starts with some scout bees leaving the hive in order to search food source to gather nectar. After finding food (i.e., flowers), scout bees return to the hive and inform their hive-mates about the richness of the flower (i.e., quantity and quality) and the distance of the flower to the hive (i.e., location) through a special movements called "dance", such as round dance, waggle dance, and tremble dance depending on the distance information of the source. Typically, she dances on different areas in an attempt to "advertise" food locations (by touching her antennae) and encourage more remaining bees to collect nectar from her source. After the dancing show, more foraging bees will leave the hive to collect nectar follow one of the dancing scout bees. Upon arrive, the foraging bee stores the nectar in her honey stomach and returns to the hive unloading the nectar to empty honeycomb cells. The described process continues repeatedly until the scout bees explore new areas with potential food sources.

### 4.1.2 Marriage Behaviour of Bees

Another famous behaviour of bees is the marriage (i.e., mating) behaviour. In the kingdom of honeybees, queens are specialized in egg-laying. A colony may contain one queen or more during its life cycle. Typically, mating occur in flight and 10–40 m above ground. It begins when the queen flights far away from the nest performing the mating flight during which the drones follow the queen and mate with her in the air. Normally, the queen mates with $12 \pm 7$ drones, and after the mating process, the drones die.

### 4.1.3 Dancing and Communication Behaviour of Bees

In general, honeybees (i.e., scout bees) perform a series of movements (i.e., dancing) to exchange information (such as the location, quantity and quality of food sources) and persuade their nestmates to follow them. There are two types of dances, i.e., round dance when food is very close and waggle dance. In addition, according to the speed of the dances, honeybees transmit the distance information, i.e., if dance is faster, then the food distance is smaller (Bitam et al. 2010).

## 4.2 Artificial Bee Colony Algorithm

### 4.2.1 Fundamentals of Artificial Bee Colony Algorithm

Artificial bee colony (ABC) algorithm was recently proposed in Karaboga and Basturk (2007), Karaboga (2005). The basic idea of designing ABC is to mimic the foraging behaviour (such as exploration, exploitation, recruitment and abandonment) of honeybees. Typically, ABC algorithm consists of two groups of bees: employed artificial bees (i.e., current exploiting foragers) and unemployed artificial bees (i.e., looking for a food source to exploit). The latter will be classified further in two groups: scouts who are searching the environment surrounding the nest for new food sources, and onlookers that are waiting in the nest and finding a food source through the information shared by employed artificial bees (Karaboga and Basturk 2008). In ABC, it is assumed that each food source position corresponds to a possible solution and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The main steps of the ABC algorithm are listed as below (Karaboga and Basturk 2008; Karaboga and Akay 2009b; Senthilnath et al. 2011):

- Initialization: The initial population can be defined as $P(G = 0)$ of $SN$ solutions (food source positions), where $SN$ denotes the size of employed bees or onlooker bees. Moreover, each solution $x_{ij}(i = 1, 2, \ldots, SN;\ j = 1, 2, \ldots, D)$ is a $D$-dimensional vector. Here, $D$ is the number of optimization parameters.

- Then, placing the employed bees on the food sources in the memory and updating feasible food source. In order to produce a candidate food position from the old one ($x_{ij}$) in memory, the memory by employed bees is updated via Eq. 4.1 (Karaboga and Basturk 2008; Karaboga and Akay 2009b; Senthilnath et al. 2011):

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right), \quad j \in \{1, 2, \ldots, D\}, k \in \{1, 2, \ldots, SN\} \wedge k \neq i, \tag{4.1}$$

where $v_{ij}$ is a new feasible dimension value of the food sources that is modified from its previous food sources value ($x_{ij}$) based on a comparison with the randomly selected neighbouring food source value ($x_{kj}$), and $\phi_{ij}$ is a random number between $[-1, 1]$ to adjust the production of neighbour food sources around $x_{ij}$ and represents the comparison of two food positions visually.

- Next, placing the onlooker bees on the food sources in the memory. The onlooker bee chooses a probability value associated with that food source ($p_i$) via Eq. 4.2 (Karaboga and Basturk 2008; Karaboga and Akay 2009b; Senthilnath et al. 2011):

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \tag{4.2}$$

where $fit_i$ is the fitness value of the solution $i$ which is proportional to the nectar amount of the food source in the position; $SN$ denotes the size of employed bees or onlooker bees. Clearly, the higher $fit_i$ is, the greater the probability is of selecting $x_{ij}$.

- Updating feasible food source, by onlooker bees using Eq. 4.1.
- Adjusting by sending the scout bees in order to discovering new food sources. The operation of scout bees explore a new food source can be defined by Eq. 4.3 (Karaboga and Basturk 2008; Karaboga and Akay 2009b; Senthilnath et al. 2011):

$$x_i^j = x_{\min}^j + rand[0, 1]\left(x_{\max}^j - x_{\min}^j\right), \tag{4.3}$$

where $x_{\min}$ and $x_{\max}$ is the lower and upper limit respectively of the search scope on each dimension. Here the value of each component in every $x_i$ vector should be clamped to the range $[x_{\min}, x_{\max}]$ to reduce the likelihood of scout bees leaving the search space.

- Memorizing the best food source found so far.
- Finally, checking whether the stopping criterion is met. If yes, terminating the algorithm; otherwise, restarting the main procedure of the ABC.

Summarizing the steps in standard ABC yields to (Karaboga and Akay 2009a; Karaboga and Akay 2009b; Karaboga and Basturk 2008; Senthilnath et al. 2011):

- Step 1: The employed bees will be randomly sent to the food sources and evaluating their nectar amounts. If an employed bee finds a better solution, she will update her memory; otherwise, she counts the number of the searches around the source in her memory.

- Step 2: If all employed bees complete the search process, the nectar and position information of the food sources will be shared with the onlooker bees.
- Step 3: An onlooker bee does not have any source in her memory and thus she will evaluate all the information from employed bees and choose a probably profitable food source (recruitment).
- Step 4: After arriving at the selected area, the onlooker bee searches the neighbourhood of the source and if she finds a better solution, she will update the food source position just as an employed bee does. The criterion for determination of a new food source is based on the comparison process of food source positions visually.
- Step 5: Stopping the exploitation process of the sources abandoned by the employed/onlooker bees if the new solution cannot be further improved through a predetermined number of trials limit. At this moment, the employed/onlooker bees become scout bees.
- Step 6: Sending the scouts into the search area for discovering new food sources (exploration), randomly.
- Step 7: Memorizing the best food source found so far.

These seven steps are repeated until a termination criterion (e.g., maximum cycle number) is satisfied.

### 4.2.2 Performance of ABC

In order to evaluate the ABC algorithm, Karaboga and Basturk (2007) employed five high dimensional benchmark testing functions. In comparison with other CI algorithms (e.g., EA, PSO, GA), the simulation results demonstrated that the ABC algorithm has the capability of getting out of a local minimum trap which make it a promising candidate in dealing with multivariable, multimodal function optimization tasks.

## 4.3 Honeybee Mating Optimization Algorithm

### 4.3.1 Fundamentals of Honeybee Mating Optimization Algorithm

In general, a honeybee community consists of three types of members: the queen, male honeybees (or drones), and neuter/undeveloped female honeybees (or workers). The honeybees mating optimization (HBMO) algorithm was proposed in Abbass (2001a, b) to simulate the social behaviour found among honeybees. In the original HBMO, a drone mates with a queen probabilistically through the use of an annealing function as stated via Eq. 4.4 (Niknam et al. 2011):

$$\Pr(D) = \exp\left(\frac{-\Delta(F)}{V_{queen}(t)}\right), \tag{4.4}$$

where $\Pr(D)$ denotes the probability for a drone $D$ adding the sperm to the queen's spermatheca, $\Delta(F)$ represents the absolute difference between the fitness of the drone and the queen, and $V_{queen}(t)$ stands for the velocity of the queen at time $t$. After each transition in space, the velocity and the energy of the queen decreases based on Eq. 4.5 (Niknam et al. 2011; Marinakis et al. 2008):

$$\begin{array}{l} V_{queen}(t+1) = \alpha \times V_{queen}(t) \\ E_{queen}(t+1) = \alpha \times E_{queen}(t) \end{array}, \tag{4.5}$$

where $\alpha$ denotes the decreasing factor which is a positive real number within the interval of $[0, 1]$. The amount of speed and energy reduction after each transition and each step is controlled through this parameter.

In HBMO algorithm, the broods that are generated through the mating process between the queen and a drone are calculated through Eq. 4.6 (Niknam et al. 2011):

$$\begin{array}{l} X_{Brood,j} = X_{queen} + \beta \times \left(X_{queen} - D_i\right) \\ X_{Brood,j} = \left[x_{broodj,1},\, x_{broodj,1},\, \ldots, x_{broodj,Ng}\right]_{1 \times Ng} \end{array}, \tag{4.6}$$

where $D_i$ represents the $i$th drone stored in the spermatheca of the queen, and $\beta \in [0, 1]$ refers to the mating factor.

Summarizing the steps in HBMO yields to (Horng 2010; Haddad et al. 2006; Afshar et al. 2007; Boussaïd et al. 2013):

- Step 1: Initializing parameters and population.
- Step 2: Mating flight process, where a queen (best solution) selects drones probabilistically to form the spermatheca (list of drones). A drone then selected from the list randomly for the creation of broods.
- Step 3: Creation of new broods by crossover the drone's genotypes with the queens.
- Step 4: Use of workers to conduct local search on broods (trial solutions).
- Step 5: Adaptation of worker's fitness, based on the amount of improvement achieved on broods.
- Step 6: Replacement of weaker queen by fitter broods.
- Step 7: Termination.

### 4.3.2 Performance of HBMO

In Marinakis et al. (2008), by utilizing the HBMO algorithm, the authors made an attempt to solve vehicle routing problem. Overall fourteen benchmark problems from the literature were chosen to test the performance of HBMO. Each instance of the problem set consists of between 51 and 200 nodes including the depot. The location of the nodes is defined by their Cartesian coordinates and the travel cost from node $i$ to $j$ is assumed to be associated with the corresponding Euclidean distance. Each selected problem includes the capacity constraints while the problems 6–10, 13, and 14 also have the restrictions of the maximum length of travel route and non zero service times. For the first ten problems, nodes are randomly placed over a square, while for the remaining four problems, nodes are distributed in clusters and the depot is not located in the centre. The experimental results obtained through the HBMO were compared with other twenty methods, among them, half is the most efficient metaheuristic approaches, and the other half is the most efficient nature inspired algorithms. Through the comparison, it can be observed that the HBMO ranks the 2nd and the 1st place among the ten metaheuristic methods and ten nature inspired algorithms, respectively.

## 4.4 Emerging Bee Inspired Algorithms

In addition to the aforementioned two bee inspired algorithms, i.e., ABC and HBMO, the characteristics of this interesting insect also motivate researchers to develop several other bee inspired innovative CI algorithms.

### 4.4.1 Artificial Beehive Algorithm

#### 4.4.1.1 Fundamentals of Artificial Beehive Algorithm

Artificial beehive algorithm (ABHA) was originally proposed by Muñoz et al. (2009). Based on a set of behavioural rules of each individual bee (e.g., the individual oriented model), the ABHA is used to solve continuous optimization problem. In order to implement ABHA, the following procedures need to be followed (Muñoz et al. 2009):

- Initializing parameters, such as the current position of the individual ($\theta(t)$), the current cost value ($J(\theta(t))$), the past cost value ($J(\theta(t-1))$), the abandon tendency ($0 < p_{ab} < 1$), and the homing motivation ($0 < p_h < 1$).
- Defining the bees' states. Typically, there are four states for each individual, i.e., novice state, experimented state, search state, and food source state. The details of each state are as follows (Muñoz et al. 2009):

State 1: Novice state. In this state, the bee is in the "nest" (i.e., an abstract position represented only by the state of the bee, where the information is exchanged) and does not have information about a source. It performs a random search or follow a dance if it is available. The current position of each bee can be defined via Eq. 4.7 (Muñoz et al. 2009):

$$\theta_i(t) = NaN(t), \tag{4.7}$$

where $NaN$ represents not a number.

State 2: Experimented state. In this state, the bee is still in the "nest" but it has some information about a food source. If the information means high quality (i.e., a good food source), it can be transmitted to other individuals through a dancing that denoted as a selection probability ($p_{si}$) via Eq. 4.8 (Muñoz et al. 2009):

$$\begin{cases} p_{si} = -\left(\frac{1}{\max(J_j - \min(J_j))}\right)(J_i - \max(J_i)) & \text{good information} \\ \text{random search or follow a dance} & \text{bad information} \end{cases}, \tag{4.8}$$

where $p_{si}$ is a selection probability that $i$ indicates one of the $j$th individuals with available dances.

State 3: Search state. In this state, the bee leaves its nest and looks for a better foraging source than the current. The bee's position is updated via Eq. 4.9 (Muñoz et al. 2009):

$$\theta_i(t+1) = \theta_i + SS(i)\psi(t), \tag{4.9}$$

where $SS(i)$ is the step size at the direction $\psi(t)$.

State 4: Food source state. Finally, the bee decides which source is the best.

- Calculating the probabilities that are used to balance the exploitation and exploration characteristics of the algorithm.

#### 4.4.1.2 Performance of ABHA

To test the effectiveness of ABHA, a set of well-known test functions were adopted in Muñoz et al. (2009), namely, Grienwank function, Rastrigin function, Ackley function, De Jong F2 function, Schewefel function, and Schaffer F6 function. Compared with other CI algorithms, such as PSO and its variants, computational results showed that ABHA achieved good results. Also, in a few cases it is capable with a better performance than the compared algorithms.

## 4.4.2 Bee Colony Optimization

### 4.4.2.1 Fundamentals of Bee Colony Optimization Algorithm

Bee colony optimization (BCO) algorithm was originally proposed in Teodorović and Dell'Orco (2005). In BCO, when the foragers return to the hive, a waggle dance is performed by each forager, then the other bees based on a probability follow the foragers. In order to implement BCO, the following procedures need to be followed (Bonyadi et al. 2008; Teodorović 2009b; Teodorović and Dell'Orco 2005; Teodorović et al. 2011):

- Step 1: Initialization: Assigning an empty solution to each bee within the colony.
- Step 2: For each bee, performing the forward pass mechanism, i.e., fulfilling the following subtasks. (a) Setting $k = 1$; (b) Evaluating all possible constructive movements; (c) Selecting one movement using the roulette wheel strategy, (d) Letting $k = k + 1$. If $k \leq NC$, redoing the subtask (b).
- Step 3: All bees return to the hive, i.e., starting the backward pass mechanism.
- Step 4: Evaluating (partial) objective function value carried by each bee.
- Step 5: For each bee, determining whether to carry on with its own exploration and becoming a recruiter, or turning to a follower. In BCO, a loyalty decision strategy is introduced at this step, i.e., at the beginning of each new round of forward pass, the probability of the $b$th bee is loyal to its previously obtained partial solution. This probability can be expressed via Eq. 4.10 (Teodorović et al. 2011):

$$p_b^{u+1} = e^{-\frac{O_{\max} - O_b}{u}}, \quad b = 1, 2, \ldots, B, \tag{4.10}$$

  where $O_b$ denotes the normalized value of the objective function carried by the $b$th bee, $O_b$ represents the maximum over all normalized values of partial solution to be evaluated, and the ordinary number of the forward pass is indicated by $u$.
- Step 6: For each follower bee, selecting a new solution from its recruiters using the roulette wheel strategy. In BCO, a recruiting mechanism is performed at this step. For each unemployed bee, it will decided to follow which recruiter based on a certain probability. The probability that $b$'s partial solution would be selected by any unallocated bee is defined by Eq. 4.11 (Teodorović et al. 2011):

$$p_b = \frac{O_b}{\sum_{k=1}^{R} O_k}, \quad b = 1, 2, \ldots, R, \tag{4.11}$$

  where $O_k$ denotes the normalized value for the objective function of the $k$th announced partial solution, and the number of recruiters is represented by $R$.
- Step 7: If the solutions are not finished, returning to Step 2.
- Step 8: Evaluating all solutions and looking for the best one.

- Step 9: Checking whether the termination criterion is met. If not, going back to Step 2; otherwise, outputting the best available solution.

In addition to the standard BCO detailed in this section, several enhanced versions of BCO can also be found in the literature as outlined below:

- Autonomous BCO (Zeng et al. 2010).
- Multiobjective BCO (Low et al. 2009).

#### 4.4.2.2 Performance of BCO

In order to see how the BCO algorithm performs, Teodorović and Dell'Orco (2005) tested it on a ride-matching problem in which the ridesharing is one of the popular travel demand management methodologies. The preliminary experimental results showed that the performance of BCO is very promising.

Apart from the original case study described above, the BCO algorithm has also been successfully applied to a variety of optimization problems as listed below:

- *p*-Center problem (Davidović et al. 2011).
- Scheduling optimization (Chong et al. 2006; Wong et al. 2008).
- Software maintenance (Kaur and Goyal 2011).
- Transportation (Teodorović 2008; Teodorović and Dell'Orco 2005).
- Travelling salesman problem (Bonyadi et al. 2008).

Interested readers are referred to these selected variants and representative applications, together with several excellent reviews [e.g., (Teodorović 2008; Bitam et al. 2010; Teodorović et al. 2011; Goyal 2012; Figueira and Talbi 2013; Teodorović 2009a)], for a further exploration and exploitation of the BCO algorithm.

### *4.4.3 Bee Colony-inspired Algorithm*

#### 4.4.3.1 Fundamentals of Bee Colony-Inspired Algorithm

Bee colony inspired algorithm (BCiA) was originally proposed in Häckel and Dippold (2009). In order to implement BCiA, the following control procedures need to be followed (Häckel and Dippold 2009):

- First of all, initializing both populations $P_1$ and $P_2$ by scout bees.
- Stage 1: Once the bees in $P_1$ finish the solution construction, they will communicate with their counterparts in $P_2$. At this stage, if the quality of a solution $\varphi^{eb}$ in $P_1$ is better than the worst one $\varphi^{worst'}$ found in $P_2$ and also if this solution is not yet contained in $P_2$, then the solution of $\varphi^{worst'}$ is replaced by $\varphi^{eb}$.
- Stage 2: If the algorithm reaches the second stage, in a similar way, the bees in $P_2$ will conduct the solution construction task with a subsequent feedback to

their peers in $P_1$. At this stage, if the quality of a solution $\varphi^{eb'}$ in $P_2$ is better than the worst one $\varphi^{worst}$ on found in $P_1$ and also in not yet included in $P_1$, then the solution of $\varphi^{worst}$ is replaced by $\varphi^{eb'}$.

- Afterwards, checking the solutions' age. If, after the maximum iteration numbers, the quality of a solution does not improve, this solution is removed. An old solution in $P_1$ is replaced by the solution carried by a scout bee, while a solution of $P_2$ is replaced by the best solution found in $P_1$, but not yet included in $P_2$.
- Finally, checking whether the stopping criterion is met. If not, the above mentioned procedure will be repeated; otherwise, the algorithm terminates.

#### 4.4.3.2 Performance of BCiA

In order to see how the BCiA performs, Häckel and Dippold (2009) tested it on a classic vehicle routing problem with time windows. The preliminary experimental results showed that the performance of BCiA is very promising, in particular for the smaller test entities of the benchmarks.

### *4.4.4 Bee Swarm Optimization*

#### 4.4.4.1 Fundamentals of Bee Swarm Optimization Algorithm

Bee swarm optimization (BSO) algorithm was originally proposed by Akbari et al. (2009). Typically, the proposed algorithm includes three types of bees, i.e., experienced forager, onlooker, and scout bees. Each type of bees had a distinct moving pattern which are used by the bees to adjust their flying trajectories. In order to implement BSO, the following procedures need to be followed (Akbari et al. 2009; Niknam and Golestaneh 2013; Sotelo-Figueroa et al. 2010):

- Initializing the population. The initial size of population ($\beta$) is determined manually that involves three types of bees via Eq. 4.12 (Akbari et al. 2009):

$$\beta = \xi \cup \kappa \cup \vartheta, \tag{4.12}$$

  where $\xi$ represents the sets of experienced forager bees, $\kappa$ denotes the sets of onlooker bees, and $\vartheta$ is the sets of scout bees.

  In addition, each bee ($i$) is associated with a position vector via Eq. 4.13 (Akbari et al. 2009):

$$\vec{x}(\beta, i) = (x(\beta, i_1), x(\beta, i_2), \ldots, x(\beta, i_D)), \tag{4.13}$$

  where $\vec{x}(\beta, i)$ denotes a feasible solution in an $d$-dimensional search space ($S \subset R^D$) that need to be optimized.

- Initializing parameters. A set of parameters will be initialized at this phase, such as the maximum number of iterations ($Iter_{\max}$), the number of the bees (n($\beta$)), and the initialization function via Eq. 4.14 (Akbari et al. 2009):

$$\vec{x}_0(\beta, i) = Init(i, S), \forall i \in \beta, \tag{4.14}$$

where $Init(i, S)$ represents the initialization function which associates a random position to the $i$th bee in the search space $S$.
- Creating the fitness function. The fitness function is given via Eq. 4.15 (Akbari et al. 2009):

$$\vec{x}(\beta, i) = fit(\vec{x}(\beta, i)). \tag{4.15}$$

- Updating phase.

1. The position of an experienced forager bee ($\xi$) is updated via Eq. 4.16 (Akbari et al. 2009):

$$\begin{aligned} \vec{x}_{new}(\xi, i) = \vec{x}_{old}(\xi, i) + \omega_b r_b \left(\vec{b}(\xi, i) - \vec{x}_{old}(\xi, i)\right) \\ + \omega_e r_e \left(\vec{e}(\xi, \cdot) - \vec{x}_{old}(\xi, i)\right), \end{aligned} \tag{4.16}$$

where $r_b$ and $r_e$ are random variables of uniform distribution in range of $[0, 1]$, $\omega_b$ and $\omega_e$ denote the best food source found by the $i$th bee and the elite bee, respectively, and $\vec{x}_{new}(\xi, i)$ represents the position vector of the new food source found by the experienced forager.
Overall, the whole equation can be divided into three parts, the first part in the right side represents the position vectors of the old food sources found by the experienced forager, the second parts in the right side represents the cognitive knowledge that attract the experienced forager towards the best position ever found by the bee, and the third parts in the right side represents the social knowledge that attract the experienced forage towards the best position ($\vec{e}(\xi, \cdot)$) which is found by the interesting elite bee.
2. The onlooker bees ($\kappa$) use only the social knowledge provided by experienced forager bees to adjust their moving trajectory in the next iteration. Their positions are updated via Eq. 4.17 (Akbari et al. 2009):

$$\vec{x}_{new}(\kappa, i) = \vec{x}_{old}(\kappa, i) + \omega_e r_e \left(\vec{e}(\xi, i) - \vec{x}_{old}(\kappa, i)\right), \tag{4.17}$$

where $\vec{x}_{new}(\kappa, i)$ represents the position of the new food source which is selected by the onlooker bee ($i$), $\omega_e r_e$ is the parameter that probabilistically controls the attraction of the onlooker bees towards their interesting food source area, and $\vec{e}(\xi, i)$ is the position vector of the interesting elite bee for onlooker bees ($p_{jd}$) that is determined through Eq. 4.18 (Akbari et al. 2009):

$$p_j = \frac{fit\left(\vec{x}(\xi, j)\right)}{\sum_{c=1}^{n(\xi)} fit\left(\vec{x}(\xi, c)\right)}, \tag{4.18}$$

where $fit\left(\vec{x}(\xi, i)\right)$ is the fitness value of the food source which is found by the experienced forager bee ($j$), and $n(\xi)$ is the number of experienced forager bees.

3. The positions of the scout bees ($\vartheta$) are updated via Eq. 4.19 (Akbari et al. 2009):

$$\vec{x}_{new}(\vartheta, i) = \vec{x}_{old}(\vartheta, i) + Rw(\tau, x_{old}(\vartheta, i)), \tag{4.19}$$

where $\vec{x}_{old}(\vartheta, i)$ represents the position of the abandoned food source, and $Rw$ is a random walk function that depends on the current position of the scout bee and the radius search $\tau$. Typically, the initial value of radius $\tau \in (\tau_{\min} < \tau < \tau_{\max})$ is defined as a percentage of $|X_{\max} - X_{\min}|$, where $X_{\max}$ and $X_{\min}$ are the maximum and minimum value of the search space along a dimension.

- Information selecting. It will be determined via Eq. 4.20 (Akbari et al. 2009):

$$\begin{cases} \text{if} \quad fit\left(\vec{x}(\xi, i)\right) > fit\left(\vec{b}(\xi, i)\right) \quad \text{then} \quad \vec{b}(\xi, i) = \vec{x}(\xi, i) \\ \text{if} \quad fit\left(\vec{b}(\xi, i)\right) > fit\left(\vec{e}(\xi, \cdot)\right) \quad \text{then} \quad \vec{e}(\xi, \cdot) = \vec{b}(\xi, i) \end{cases}, \tag{4.20}$$

where $\vec{b}(\xi, i)$ denotes the position of the best food source that an experienced forager ($i$) can remember, and $\vec{e}(\xi, \cdot)$ represents the position of the best food source that the elite bee can find.

#### 4.4.4.2 Performance of BSO

To evaluate the effectiveness of BSO, six analytical benchmark functions were employed in Akbari et al. (2009). Compared with other bee inspired algorithms (such as ABC), computational results showed that the proposed algorithm outperforms the others investigated in this study.

### *4.4.5 Bee System*

#### 4.4.5.1 Fundamentals of Bee System Algorithm

Bee system (BS) algorithm was originally proposed in Sato and Hagiwara (1997). In order to implement BS, the following procedures need to be followed (Lučić 2002; Sato and Hagiwara 1997; Lučić and Teodorović 2003):

- In BS, the first global search is performed by *pop_G*. The purpose of this function is to find as broad as possible to escape from a local optimum. If, for successive $G_{SC}$ generations, one chromosome is found to be the best, it will be regarded as a very good solution point around which there may exist the global best. In BS, this solution is call superior chromosome which will be kept for local search.
- Concentrated crossover: All chromosomes in $pop\_L_k$ are made couple with $SC_k$ at the beginning of local search and the crossover mechanism is performed. This concentrated crossover transfers information about the $k$th superior chromosome to all other chromosomes in the $k$th population denoted by $pop\_L_k$.
- Migration: In BS, an individual bee is randomly selected per predetermined generation $G_{mig}$ for emigrating to its neighbourhood population. Through this strategy, each population manages to search independently and cooperatively.
- Pseudo-simplex approach: In BS, for a more efficient and effective search, a pseudo-simplex mechanism is employed.

First, picking up three best so far chromosomes and name them $C_1$, $C_2$, and $C_3$ according to their corresponding fitness value.

Then, translating them into three vectors, i.e., $\mathbf{X}_1$, $\mathbf{X}_2$, and $\mathbf{X}_3$. Next, calculating the middle point $\mathbf{X}_0$ of $\mathbf{X}_1$ and $\mathbf{X}_2$ according to Eq. 4.21 (Sato and Hagiwara 1997):

$$\mathbf{X}_0 = \frac{\mathbf{X}_1 + \mathbf{X}_2}{2}. \tag{4.21}$$

Right after this, computing $\mathbf{X}_{ref}$ and $\mathbf{X}_{cont}$, respectively, based on Eqs. 4.22 and 4.23, respectively (Sato and Hagiwara 1997):

$$\mathbf{X}_{ref} = (1 + \alpha)\mathbf{X}_0 - \alpha\mathbf{X}_3, \tag{4.22}$$

$$\mathbf{X}_{cont} = (1 - \beta)\mathbf{X}_0 + \beta\mathbf{X}_3, \tag{4.23}$$

where the reflection ration and contraction ration are denoted by $\alpha$ and $\beta$, respectively.

Next, exchanging $\mathbf{X}_{ref}$ and $\mathbf{X}_{cont}$ into chromosomes and set them as $C_{ref}$ and $C_{cont}$, respectively.

Finally introducing $C_1$, $C_{ref}$, and $C_{cont}$ to the initial population where the crossover and mutation have already applied.

- Turning back to global search: The local search will be terminated once the predetermined number of generations is reached. If the best solution found so far does not meet the stopping criterion, the algorithm will repeat.

#### 4.4.5.2 Performance of BS

In order to see how the BS algorithm performs, Sato and Hagiwara (1997) tested it on nine benchmark test functions selected from the literature. Compared with other CI approaches (e.g., GA), the preliminary experimental results showed that BS outperforms GA in all cases which make it a very promising optimization algorithms in dealing with highly complex multivariate functions.

### *4.4.6 BeeHive*

#### 4.4.6.1 Fundamentals of BeeHive Algorithm

BeeHive algorithm was originally proposed in Wedde et al. (2004) and Farooq (2006) which is a novel network routing algorithm. It is inspired from the dance language and foraging behaviour of honeybees. The main characteristics of BeeHive algorithm can be concluded as follows: (Farooq 2009; Wedde and Farooq 2006; Farooq and Caro 2008):

- Step 1: During the start up stage, all nodes in the networks begin with a process named foraging region formation. The first generation of short distance bee agents are launched at this stage for propagating their nomination in their neighbourhood.
- Step 2: By comparing the information received from a short distance bee agent, a node will determined whether to resign as a representative node and join the foraging region.
- Step 3: Once the former representative node quits, the other nodes will activate an election mechanism
- Step 4: The nodes continues to announce the next generations of short distance bee agents by following pre-described steps until the network is split into disjoint foraging regions, and overlapping with the foraging zones.
- Step 5: After the Step 4 is executed, the BeeHive algorithm gets into a normal phase.
- Step 6: When a replica of a specific bee agent reaches a site, it will update the local stored routing information, and then, except of being sent back to the node where the replica comes from, it will be continuously flooded.
- Step 7: Representative nodes only generate long distance bee agents that could be received by the neighbours.
- Step 8: In BeeHive algorithm, bee agents employ priority queues mechanism for the purpose of a quick routing information dissemination.
- Step 9: Each node carries the current routing information not only for reaching nodes within its foraging zone, but also for reaching the representative nodes of foraging regions.
- Step 10: Choosing the next hop for a data packet in a stochastic manner.

- Step 11: The goodness of a neighbour $j$ of node $i$ for arriving at a destination $d$, denoted by $g_{jd}$, is defined via Eq. 4.24 (Wedde and Farooq 2006):

$$g_{jd} = \frac{\frac{1}{p_{jd}+q_{jd}}}{\sum_{k=1}^{N}\left(\frac{1}{p_{kd}+q_{kd}}\right)}, \tag{4.24}$$

where the propagation and queuing delays are denoted by $p_{jd}$ and $q_{jd}$, respectively.
- Step 12: In BeeHive algorithm, three kinds of routing tables, namely, intra foraging zone, inter foraging region, and foraging region membership are allocated to each node $i$.

#### 4.4.6.2 Performance of BeeHive Algorithm

In order to evaluate the BeeHive algorithm, the Japanese Internet backbone scenario was employed in Wedde et al. (2004). Through an extensive comparison with two other state of the art routing algorithms, BeeHive showed a quite attractive overall performance. Interested readers are referred to Farooq (2009) for a more detailed explanation about the working principles and applications of the BeeHive algorithm.

### *4.4.7 Bees Algorithm*

#### 4.4.7.1 Fundamentals of Bees Algorithm

Bees algorithm (BeA) was originally proposed in Pham et al. (2006). The basic working procedures of BeA are listed as follows (Pham et al. 2006; Karaboga and Akay 2009a; Pham and Castellani 2009; El-Abd 2012b):

- Step 1: Initializing population with random solutions. At this stage, the BA requires several parameters to be set such as number of scout bees ($n$), number of sites selected out of $n$ visited sites ($m$), number of the best sites out of $m$ selected sites ($e$), number of bees recruited for the best $e$ sites ($nep$), number of bees recruited for the other ($m-e$) selected sites ($nsp$), and initial size of patches ($ngh$) which includes the site, its neighbourhood, and the stopping criterion.
- Step 2: Assessing the fitness of the population.
- Step 3: Forming new population while stopping criterion is not met.
- Step 4: Selecting sites for neighborhood search. The bees with the highest fitness values are chosen as "selected bees" at Step 4, and accordingly, the sites visited by them are chosen for neighborhood search.
- Step 5: Recruiting bees for selected sites (more bees for best $s$ sites), and evaluating the fitness value. In Step 5, the BeA conducts search in the

neighborhood of the selected sites. More bees will be assigned to search around the best *e* sites. The bees can either be selected directed based on the value of fitness associated with the sites they are visiting, or the fitness values will be used to determine the probability of the bees being selected. Exploration of the surroundings of the best *e* sites represents more suitable solutions can be made available through recruiting more bees to follow them than the other selected bees. This differential recruitment mechanism, along with scouting strategy, is the core operation of the BeA algorithm.

- Step 6: Choosing the fittest bee from each patch. Pham et al. (2006) introduced a constraint at this stage that is for each patch, only the bees with the highest fitness value can be selected to from the next bee population. The purpose of introducing such restriction is to reduce the number of points that are going to be explored.
- Step 7: Assigning the remaining bees to do random search, and evaluating their fitness. In the bee population, the remaining bees are assigned randomly around the search space looking for new possible solution candidates.
- Step 8: Terminating the loop when stopping criterion is met. All aforementioned steps will be executed repeatedly until a stooping criterion is met. At the end of each iteration, the population of bee colony consists of two parts: representatives from each selected patch, and other scout bees performing random searches.

In addition to the standard BeA detailed in this section, several enhanced versions of BeA can also be found in the literature as outlined below:

- Binary BeA (Xu et al. 2010b).
- Distributed BeA (Jevtić et al. 2012).
- Hybrid BeA (Shafia et al. 2011; Lien and Cheng 2012).
- Multi-objective BeA (Pham and Ghanbarzadeh 2007).
- Neighbourhood enhanced BeA (Ahmad 2012).

#### 4.4.7.2 Performance of BeA

In order to see how BeA performs, Pham et al. (2006) first employed two standard benchmark functions, namely, Shekel's Foxholes function and inverted Schwefel's function with six dimensions, for testing purpose. Furthermore, eight other benchmark functions selected from the literature were introduced for validating BeA. Compared with other CI methods (e.g., GA and ACO), the overall performance of BeA is quite competitive.

Recently, Xu et al. (2010b) utilized BeA to study a group of reconfigurable mobile robots which are designed to provide daily service in hospital environments for different kinds of tasks such as guidance, cleaning, delivery, and monitoring. The fulfilment of each job requires an associated functional module that can be installed onto various robot platforms via a standard connection interface.

Since the classic BeA focuses mainly on single-objective functional optimization problems, a variant called binary BeA was proposed in Xu et al. (2010b) to deal with the multi-objective multi-constraint combinatorial optimization task. In binary BeA, a bee is describe as two binary matrixes **MR** and **RH**, standing for how to assign the $M$ tasks to the $R$ robots and the $R$ robots to the $H$ homes, respectively. The size of **MR** is $M \times R$ in which its $R$ columns represent the $R$ robots, while the $M$ missions is represented by the $M$ rows. Xu et al. (2010b) evaluated the proposed algorithm with an example problem (20 missions, 8 robots, and 4 homes) with a size of $8^{20} \times 4''^{8} = 2^{76}$ combinations. At first 12 stochastic solutions are obtained by scout bees through global search in which six elite bees survive after the non-dominated selection. The final experiments demonstrated that the proposed algorithm is a suitable candidate tool in treating workload balancing issue among a team of swarm robots.

Apart from the case study described above, the BeA has also been successfully applied to a variety of optimization problems as listed below:

- Controller design optimization (Jones and Bouffet 2008).
- Construction site layout optimization (Lien and Cheng 2012).
- Data clustering (Shafia et al. 2011).
- File search optimization (Dhurandher et al. 2011).
- Filter design optimization (Pham and Koç 2010).
- Manufacturing system optimization (Pham et al. 2007a, Ramírez et al. 2010).
- Mechanical design optimization (Pham et al. 2009).
- Robot control optimization (Xu et al. 2010b; Jevtić et al. 2012).
- Scheduling optimization (Pham et al. 2007b).

## 4.4.8 Bees Life Algorithm

### 4.4.8.1 Fundamentals of Bees Life Algorithm

Bees life algorithm (BLA) was recently proposed by Bitam and Mellouk (2013) to solve a vehicular ad hoc network (VANET) problem, in particular, the quality of service multicast routing problem (QoS-MRP). Two bees' behaviours are employed in the proposed algorithm, i.e., reproduction and foraging behaviours. In addition, the former incorporated the crossover and mutation operators, while the latter used a neighbourhood search approach. In order to implement BLA, the following procedures need to be followed (Bitam and Mellouk 2013):

- Initializing bees' population.
- The fitness function value corresponding to each candidate solution is calculated.
- Reproduction process. In this process, based on the crossover and mutation operators, two new individuals (i.e., queen bee and drone bee) are selected. The queen starts breeding broods, then those broods will be evaluated according to

the fitness function, after that, the best fittest brood will be considered as the new queen for next iteration. Also, the drone bees and worker bees are updated.
- Foraging process. In this process, the worker bees are looking for food sources. Each worker will be mimicked as one region. During the recruitment process, the best food source will be founded.
- Ranking and selecting. Only the highest fitness will be selected to form the next bee population.
- Evaluating the fitness of population (i.e., queen, drone, and worker bees).
- Termination.

#### 4.4.8.2 Performance of BLA

To evaluate the effectiveness of BLA, a series of tests based on VANET simulation scenario are conducted in Bitam and Mellouk (2013). Compared with other CI algorithms (such as GA, BeA, and HBMO), computational results showed that the proposed algorithm outperforms the others in terms of the solution quality and complexity.

### *4.4.9 Bumblebees Algorithm*

#### 4.4.9.1 Fundamentals of Bumblebees Algorithm

Bumblebees algorithm was originally proposed in Comellas and Martínez-Navarro (2009). It is a more simple and efficient version than the one introduced in Comellas and Gallegos (2005) as angels and mortals. In bumblebees algorithm, the bumblebees are employed to play the role of the mortals, while the static food cells are used to replace the angels. The food cells together with the fixed position of nest creates a simulating environment which can influence the bumblebees' behaviour and in turn helps to move the optimization process forward. In bumblebees algorithm, the following operators are defined (Comellas and Martínez-Navarro 2009):

- *The world*: In the proposed algorithm, the habitat of the colony of bumblebees is an artificial world consisting of a toroidal square grid with $n \times n$ cells. There are four possible states assigned to each individual cell, namely, empty, with food, with a bumblebee, or containing a nest.
- *Movement*: Initially, all bumblebees are located in the next. At each generation, each single bumblebee will fly out of the nest one by one and randomly move to any other positions around its current position.
- *Bumblebees' birth*: During the algorithm initialization, all bumblebees are positioned inside the nest and each bumblebee is associated with a randomly generated solution. The record of some of the best solutions found so far is kept

by the queen who will pass one of such solution to a newly born bumblebee every few generation.

- *Mutation*: Similarly to GA, the mutation mechanism is also introduced in bumblebees algorithm with a slightly modification of applying mutation on all individual bumblebees at each generation.
- *The reaper*: In order to avoid the over- or de-population, a reaper mechanism is employed in bumblebees algorithm, i.e., one unit of life is subtracted from a bumblebee's life at each generation. When the age of a bumblebee arrives at 0, this individual will be removed from the current artificial world.

#### 4.4.9.2 Performance of Bumblebees Algorithm

In order to test the performance of the bumblebees algorithm, Comellas and Martínez-Navarro (2009) tested it on the classic graph colouring problem. In comparison with other CI methods (e.g., GA), the bumblebees algorithm offered a better solution quality.

### *4.4.10 Honeybee Social Foraging Algorithm*

#### 4.4.10.1 Fundamentals of Honeybee Social Foraging Algorithm

Honeybee social foraging (HBSF) algorithm was recently proposed in Quijano and Passino (2010). In order to implement the HBSF algorithm, the following procedures need to be followed (Quijano and Passino 2010; Nakrani and Tovey 2003; Nakrani and Tovey 2004; Scholz-Reiter et al. 2008):

- Foraging profitability landscape: For $i = 1, 2, \ldots, B$, bee $i$ is denoted by $\theta^i \in \Re^2$, a position expression within 2-dimensional space. The foraging profitability landscape is represented by $J_f(\theta)$ which has a value falling within the range of $[0, 1]$. It is proportional to the profitability of nectar at a location indicated by $\theta \in \Re^2$.
- Bee roles and expeditions: Setting $x_j(k)$ as the number of bees at site $j$ at $k$. A normal option for expressing this is via Eq. 4.25 (Quijano and Passino 2010):

$$s_j(k) = \frac{a_j}{x_j(k)}, \tag{4.25}$$

where the amount of nutrients per second at the $j$th site is denoted by $a_j$. Suppose that there are $B_f(k)$ deployed forager bees. Initially letting $B_f(0) = 0$ for the reason of no foraging sites are being discovered. Then the foraging profitability assessment used by an individual deployed forager bee can be expressed via Eq. 4.26 (Quijano and Passino 2010):

$$F^i(k) = \begin{cases} 1 & \text{if } J_f\left(\theta^i(k)\right) + w_f^i(k) \geq 1 \\ J_f\left(\theta^i(k)\right) + w_f^i(k) & \text{if } 1 > J_f\left(\theta^i(k)\right) + w_f^i(k) > \varepsilon_n \, , \\ 0 & \text{if } J_f\left(\theta^i(k)\right) + w_f^i(k) \leq \varepsilon_n \end{cases} \tag{4.26}$$

where $w_f^i(k)$ stands for the profitability assessment noise.

- Dance strength determination: In HBSF, the dance strength is denoted by $L_f^i(k)$, i.e., the number of waggle runs of bee $i$ at step $k$. Setting $F_q^i(k)$ as the amount of nectar gathered under the profitability assessment $F^i(k)$. Then the total quantity of nectar influx to the hive at step $k$, denoted by $F_{tq}(k)$ can be defined via Eq. 4.27 (Quijano and Passino 2010):

$$F_{tq}(k) = \sum_{i=1}^{B} F_q^i(k) = \alpha \sum_{i=1}^{B} F^i(k) = \alpha F_t(k). \tag{4.27}$$

- Explorer allocation and forager recruitment: In HBSF, the explorer allocation process is designed to simultaneously happen with the recruitment of observer bees to forage sites. Then the probability that the dance of bee $i$ will be followed by an observer bee is defined via Eq. 4.28 (Quijano and Passino 2010):

$$p_i(k) = \frac{L_f^i(k)}{\sum_{i=1}^{B_f(k)} L_f^i(k)}. \tag{4.28}$$

#### 4.4.10.2 Performance of HBSF

To evaluate the effectiveness of the HBSF algorithm, an engineering application, namely, dynamic resource allocation for multizone temperature control problem was adopted by Quijano and Passino (2010). Computational results showed that the proposed algorithm is able to achieve an ideal free distribution situation which could maximize the uniform temperature allocation.

### *4.4.11 OptBees*

#### 4.4.11.1 Fundamentals of OptBees Algorithm

OptBees algorithm was originally proposed in Maia et al. (2012) that is based on the processes of collective decision-making by bee colonies to solve multimodal continuous optimization problem. In order to implement the OptBees algorithm, the following procedures need to be followed (Maia et al. 2012):

- Determination of the recruiter bees: Typically, the probability of being a recruiter bee is associated with each bee in the swarm is given via Eq. 4.29 (Maia et al. 2012):

$$p_i = \left(\frac{p_{\max} - p_{\min}}{Q_{\max} - Q_{\min}}\right) \cdot (Q_i - Q_{\min}) + p_{\min}, \tag{4.29}$$

  where $p_{\min}$ and $p_{\max}$ are the minimum and maximum probabilities of a bee be a recruiter, respectively, $Q_i$ represents the quality of the site explored by bee $i$, and $Q_{\min}$ and $Q_{\max}$ represent the minimum and maximum site qualities at the current iteration. After these procedures, the number non-recruiter bees in the swarm is defined via Eq. 4.30 (Maia et al. 2012):

$$M = N - r, \tag{4.30}$$

  where $N$ is the total number of bees, $r$ denotes the number of recruiter bees, and $M$ represents the number of non-recruiter bees.
- Determination of the recruited and scout bees: The number of recruited bees ($n$) is calculated by Eq. 4.31 (Maia et al. 2012):

$$n = [p_{rec} \cdot M], \tag{4.31}$$

  where $p_{rec}$ is the percentage of non-recruiter bees that will be actually recruited, $M$ is the number of non-recruiter bees, and $[\cdot]$ denotes the nearest integer function. So, the number of scout bees ($S$) is given by Eq. 4.32 (Maia et al. 2012):

$$S = M - n, \tag{4.32}$$

- Recruitment process: Due to two-dimensional search spaces, two equal probabilities' level are employed in this process via Eq. 4.33 (Maia et al. 2012):

$$\begin{cases} x_i = x_i + \alpha \cdot \mathbf{U} \otimes (y - x_i) \\ x_i = x_i + u \cdot \alpha \cdot (y - x_i) \end{cases}, \tag{4.33}$$

  where $\alpha$ is the recruitment rate, is the recruited bee, $x_i$ is the recruiter bee, $y$ is a random number with uniform distribution in the interval $[0, 1]$, $\mathbf{U}$ is a vector whose elements are random numbers with uniform distribution in the interval $[0, 1]$, and $\otimes$ denotes the element-wise product.

- Exploration process: In this process, the scout bees are moved to a random point (i.e., a new region in the search space).

#### 4.4.11.2 Performance of OptBees

To evaluate the effectiveness of the OptBees algorithm, five minimization problems were adopted by Maia et al. (2012). Computational results showed that the proposed algorithm is capable of generating and maintaining the diversity and consequently obtaining multiple local optima solutions without losing the ability of global optimization.

## 4.4.12 Simulated Bee Colony Algorithm

### 4.4.12.1 Fundamentals of Simulated Bee Colony Algorithm

Simulated bee colony (SBC) algorithm was originally proposed in McCaffrey and Dierking (2009) to extract rule sets from clustered categorical data. In SBC, each bee is viewed as an object with a memory matrix and it is modelled as an array. In order to implement the SBC algorithm, the following procedures need to be followed (McCaffrey and Dierking 2009):

- Step 1: Initialization.
- Step 2: Repeating. First, placing the active bees on the food sources in the memory. Then, putting the inactive bees on the food source in the memory. Next, sending the scout bees to the search area for discovering new food sources. Finally, memorizing the best food source found so far.
- Step 3: Termination.

### 4.4.12.2 Performance of SBC

To evaluate the effectiveness of SBC, the proposed algorithm was tested with six benchmark data sets in McCaffrey and Dierking (2009). Computational results showed that SBC can successfully discover the underlying rule set for all six test data sets.

## 4.4.13 Virtual Bees Algorithm

### 4.4.13.1 Fundamentals of Virtual Bees Algorithm

Virtual bees algorithm (VBA) was originally proposed by Yang (2005) and recently is used in Khan et al. (2010) for solving controller design problem. For VBA, the population of bees is associated with a memory bank, a food source. Also, all the memories communicate between bees with a waggle dance procedure. In order to implement VBA, the following procedures need to be followed (Khan et al. 2010; Yang 2005):

- Initializing the population.
- The fitness function value corresponding to each candidate solution is calculated.

- Defining a criterion for communicating the direction and distance.
- Updating the new position of each individual. The new position is updated via Eq. 4.34 (Khan et al. 2010; Yang 2005):

$$\begin{cases} x_k^{i+1} = x_k^i \cdot (1+\beta) + x_{best} \cdot \beta + \alpha \cdot (rand(i) - 0.5) \\ y_k^{i+1} = y_k^i \cdot (1+\beta) + y_{best} \cdot \beta + \alpha \cdot (rand(i) - 0.5) \end{cases}, \tag{4.34}$$

  where $\alpha$ and $\beta$ are two positive constants called randomness amplitude and speed of convergence, respectively, $x_{best}$ and $y_{best}$ are best parameters in the $i$th iteration, and $rand(i)$ is a random number in the interval $[0, 1]$.
- Ranking the candidate solutions.
- Checking termination criterion.

#### 4.4.13.2 Performance of VBA

To evaluate the effectiveness of VBA, De Jong's test function and keane's multi-peaked bumpy function were adopted in Yang (2005). Compared with other CI algorithms (such as GA), computational results showed that the proposed algorithm is capable of solving multilevel optimization problems in which many local minimums are involved.

### *4.4.14 Wasp Swarm Optimization*

#### 4.4.14.1 Fundamentals of Wasp Swarm Optimization Algorithm

Wasp swarm optimization (WSO) algorithm was originally proposed in Theraulaz et al. (1991) that is based on some behaviours found in wasp colony (Karsai and Wenzel 2000; Lucchetta et al. 2008).

The basic idea of WSO was to mimic a wasp colony behaviour, in particular according to the importance of individual wasp to the whole colony, assigning the resources to different wasp (Fan and Zhong 2012; Theraulaz et al. 1991). Therefore in WSO algorithm, resources will be allocated to individual candidate solutions and such allocation is completed in a randomly manner where the strength of each option controls its chosen probability. In Cicirello and Smith (2004), a tournament process was utilized to implement this stochastic selection process: the weakest option (for example $a$) challenges the second weakest option (for instance $b$) and the winning probability of $a$ over $b$ is determined through $p_{ab} = s_a^2 / \left(s_a^2 + s_b^2\right)$. The winner of this challenge (say $a$) will carry on to challenges the third weakest option (denoted by $c$), and wins with a probability of $p_{ac} = s_a^2 / \left(s_a^2 + s_c^2\right)$. The challenge will continue until the final winner is selected. In some situations, it is more convenient to allocate costs instead of strengths to the individual wasps, i.e., the lower the cost, the higher the strength of a wasp. In this case, the winning probability of wasp $i$ over $j$ can be defined via Eq. 4.35 (Cicirello and Smith 2004):

$$p_{ij} = \frac{s_j^2}{s_i^2 + s_j^2}, \; i, j = 1, \ldots, c. \tag{4.35}$$

#### 4.4.14.2 Performance of WSO

In Song et al. (2005), the authors utilized WSO to find a trade-off between the distribution cost and the service level in the context of dynamic vehicle routing problem with time windows (DVRPTW for short). In real-world environment, a real-time request impacts more on one vehicle route while less on the other vehicle routes. In order to take the geographical location information and time window information of the customers into account, a wasp-like agent strategy is employed in their study to determine when to re-optimize the vehicle routes. The vehicle considered in Song et al. (2005) is associated with a wasp named vehicle wasp which is used to control the status of the vehicle and it has a set of response thresholds as stated in Eq. 4.36 (Song et al. 2005):

$$\theta_v = \left\{\theta_{v,1}, \theta_{v,2}, \ldots, \theta_{v,n}\right\}, \tag{4.36}$$

where $\theta_{v,i}$ denotes the response threshold value of the vehicle wasp to a new demand $i$.

Based on the simulation data for static VRPTW found in the literature, the authors re-constructed them for DVRPTW through producing request randomly. The experimental results showed that when the dynamic property is less than 80 %, the WSO algorithm can generates a better solution. Nevertheless, when the dynamic property is greater than 80 %, the WSO algorithm failed to provide a suitable solution. At the end of the study, the authors claimed that WSO algorithm could fit the practical distribution environment well as long as the route that needs to be re-optimized does not deviate far from the original planned route.

Apart from the case study described above, the WSO algorithm has also been successfully applied to a variety of optimization problems as listed below:

- Data clustering (Runkler 2008).
- Decentralized control optimization (Theraulaz and Bonabeau 1995; Baker 1998; Karsai 1999; Anderson and Bartholdi 2000; Cicirello and Smith 2004; Wang 2009).
- Image processing (Fan and Zhong 2012).
- Logistics system optimization (Pinto et al. 2005).
- Scheduling optimization (Wang et al. 2006).
- Vehicle routing problem (Song et al. 2005).
- Maximum satisfiability problem (Pinto et al. 2006; Cao et al. 2009; Anonymous 2010).

## 4.5 Conclusions

In this chapter, we introduced a set of CI algorithms which are based on the behaviour of the honeybees. These algorithms are mainly divided into two categories, i.e., the foraging behaviour and the mating behaviour. Although they are newly introduced CI methods, we have witnessed the following rapid spreading of at least two of them, i.e., ABC and HBMO.

First, several enhanced versions of ABC can be found in the literature as outlined below:

- ABC* (Forghany et al. 2012).
- Cauchy distribution-based ABC (Rajasekhar et al. 2011).
- Chaotic ABC (Xu et al. 2010a; Alatas 2010).
- Cooperative ABC (El-Abd 2010).
- DisABC (Kashan et al. 2012).
- Discrete ABC (Tasgetiren et al. 2011; Tasgetiren et al. 2010; Karabulut and Tasgetiren 2012; Koc et al. 2012; Tasgetiren et al. 2013).
- Elitist ABC (Mezura-Montes and Velez-Koeppel 2012).
- Enhanced ABC (Tsai et al. 2009).
- Global best ABC (Gao et al. 2012; Li et al. 2012; Jadhav and Roy 2013).
- Hybrid ABC (Tien and Li 2012; Shi et al. 2010; Vitorino et al. 2012; Kang et al. 2013; Zhang et al. 2013).
- Improved ABC (Karaboğa and Çetinkaya 2011; Gao et al. 2011; Gao and Liu 2011; Cheng and Jiang 2012).
- Mean mutation operator based ABC (Sharma et al. 2012a).
- Memetic ABC (Fister et al. 2012).
- Micro ABC (Rajasekhar et al. 2012).
- Modified ABC (Anandhakumar et al. 2011; Mezura-Montes et al. 2010; Gao and Liu 2012; Sharma et al. 2012a; Gao et al. 2013).
- Multi-hive ABC (Zhang et al. 2012).
- Multiobjective ABC (Akbari et al. 2012; Chaves-González et al. 2013).
- Mutable smart ABC (Gorji-Bandpy and Mozaffari 2012).
- Opposition-based ABC (El-Abd 2012a).
- Parallel ABC (Narasimhan 2009; Subotic et al. 2010).
- Penalty guided ABC (Hsieh and Yeh 2012).
- Rosenbrock ABC (Kang et al. 2011).
- Vector evaluated ABC (Omkar et al. 2011).
- Chaotic improved HBMO (Niknam et al. 2011).
- Hybrid HBMO (Niknam et al. 2008; Niknam 2009).
- Maximum entropy based HBMO (Horng 2010).
- Modified HBMO (Olamaei et al. 2012; Niknam et al. 2012).
- Multiobjective HBMO (Niknam 2011).

Second, the ABC algorithm has also been successfully applied to a variety of optimization problems as listed below:

- Artificial neural network training (Dhahri et al. 2012).
- Automated software refactoring (Koc et al. 2012).
- Circuit design optimization (Zhang and Ye 2012).
- Composite structures optimization (Omkar et al. 2011).
- Controller design optimization (Rajasekhar et al. 2012).
- Disassembly line balancing (Kalayci and Gupta 2013).
- Expert system design (Babu et al. 2011).
- Filter design optimization (Karaboğa and Çetinkaya 2011).
- Fuel cell research (Zhang et al. 2013).
- Gene research (Forghany et al. 2012; Chaves-González et al. 2013; lez-Álvarez et al. 2013).
- Image processing (Ma et al. 2011; Akay 2013; Hancer et al. 2012; Akay and Kirmizi 2012).
- Manufacturing optimization (Samanta and Chakraborty 2011; Yildiz 2013; Ajorlou and Shams 2012).
- Portfolio optimization (Chen et al. 2012).
- Power system optimization (Hemamalini and Simon 2010; Hong 2011; Subramanian et al. 2011; Anandhakumar et al. 2011; Ayan and Kılıç 2012; Bommirani and Thenmalar 2013; Jadhav and Roy 2013).
- Redundancy allocation problem (Hsieh and Yeh 2012).
- Robot control (Xu et al. 2010a).
- Scheduling optimization (Tasgetiren et al. 2011; Sundar and Singh 2012; Tasgetiren et al. 2010; Tasgetiren et al. 2013).
- Supply chain optimization (Kumar et al. 2010).
- Thermal engine optimization (Gorji-Bandpy and Mozaffari 2012).
- Travelling salesman problem (Karabulut and Tasgetiren 2012).
- Unstable periodic orbits detection (Gao et al. 2012).

Third, the relative HBMO algorithm applications can be found below:

- Image processing (Horng 2010).
- Power system optimization (Niknam et al. 2008; Niknam 2009; Niknam et al. 2011; Niknam 2011; Olamaei et al. 2012; Niknam et al. 2012).
- Travelling salesman problem (Marinakis et al. 2011).
- Vehicle routing problem (Marinakis et al. 2008; Marinakis et al. 2010).
- Water resource management (Haddad et al. 2006; Afshar et al. 2007; Haddad et al. 2009; Haddad et al. 2010).

Interested readers are referred to them together with several excellent reviews [e.g., (Karaboga and Akay 2009c; Karaboga et al. 2012; Teodorović 2009a; Teodorović et al. 2011; Goyal 2012)] as a starting point for a further exploration and exploitation of the honeybees inspired algorithms.

## References

Abbass, H. A. (2001a, May 27–30). MBO: Marriage in honey bees optimization. A haplometrosis polygynous swarming approach. In *IEEE Proceedings of the Congress on Evolutionary Computation* (pp. 207–214). Seoul, South Korea.

Abbass, H. A. (2001b). A monogenous MBO approach to satisfiability. In *Proceeding of the International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA)*. Las Vegas, NV, USA.

Afshar, A., Haddad, O. B., Mariño, M. A., & Adams, B. J. (2007). Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. *Journal of the Franklin Institute, 344*, 452–462.

Ahmad, S. A. (2012). *A study of search neighbourhood in the bees algorithm*. Unpublished doctoral thesis, Cardiff University.

Ajorlou, S., & Shams, I. (2013). Artificial bee colony algorithm for CONWIP production control system in a multi-product multi-machine manufacturing environment. *Journal of Intelligent Manufacturing, 24*, 1145–1156. doi:10.1007/s10845-012-0646-5.

Akay, B., (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing, 13,* 3066–3091. (http://dx.doi.org/10.1016/j.asoc.2012.03.072).

Akay, B., & Kirmizi, I. (2012, June 10–15). Structural optimization of wavelet packets using swarm algorithms. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–5). Brisbane, Australia.

Akbari, R., Mohammadi, A., & Ziarati, K. (2009). A powerful bee swarm optimization algorithm. In *IEEE 13th International Multitopic Conference (INMIC)* (pp. 1–6).

Akbari, R., Hedayatzadeh, R., Ziarati, K., & Hassanizadeh, B. (2012). A multi-objective artificial bee colony algorithm. *Swarm and Evolutionary Computation, 2*, 39–52.

Alatas, B. (2010). Chaotic bee colony algorithms for global numerical optimization. *Expert Systems with Applications, 37*, 5682–5687.

Anandhakumar, R., Subramanian, S., & Ganesan, S. (2011). Modified ABC algorithm for generator maintenance scheduling. *International Journal of Computer and Electrical Engineering, 3*, 812–819.

Anderson, C., & Bartholdi, J. J. (2000, September 19–20). Centralized versus decentralized control in manufacturing: Lessons from social insects. In *International Conference on Complexity and Complex Systems in Industry* (pp. 92–105). UK: University of Warwick.

Anonymous. (2010). WAIST: Wasp inspired scheduling for real-time strategy games. *Especial Videojogos, 11*, 1–7.

Ayan, K., & KıLıÇ, U. (2012). Artificial bee colony algorithm solution for optimal reactive power flow. *Applied Soft Computing, 12*, 1477–1482.

Babu, M. S. P., Ramjee, M., Narayana, S. S. V. N. L., & Murty, S. N. V. R. (2011). Sheep and goat expert system using artificial bee colony (ABC) algorithm and particle swarm optimization (PSO) algorithm. In *IEEE 2nd International Conference on Software Engineering and Service Science (ICSESS)* (pp. 51–54).

Baker, A. D. (1998). A survey of factory control algorithms that can be implemented in a multi-agent heterarchy: Dispatching, scheduling, and pull. *Journal of Manufacturing Systems, 17*, 297–320.

Bitam, S., & Mellouk, A. (2013). Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks. *Journal of Network and Computer Applications, 36*, 981–991.

Bitam, S., Batouche, M., & Talbi, E.-G. (2010, April 19–23). A survey on bee colony algorithms. In *IEEE International Symposium on Parallel and Distributed Processing, Workshops and Phd Forum (IPDPSW)* (pp. 1–8). Atlanta, GA.

Bommirani, B., & Thenmalar, K. (2013). Optimization technique for the economic dispatch in power system operation. *International Journal of Computer and Information Technology, 2*, 158–163.

Bonyadi, M. R., Azghadi, M. R., & Shah-Hosseini, H. (2008). Population-based optimization algorithms for solving the travelling salesman problem. In F. Greco (Ed.), *Travelling salesman problem, Chapter 1* (pp. 1–34). Vienna: In-Tech.

Boussaïd, I., Lepagnot, J., & Siarry, P. (2013). A survey on optimization metaheuristics. *Information Sciences, 237*, 82–117.

Cao, Y., Yang, Y., & Wang, H. (2009). Intelligent job shop scheduling based on MAS and integrated routing wasp algorithm and scheduling wasp algorithm. *Journal of Software, 4*, 487–494.

Chaves-González, J. M., Vega-Rodríguez, M. A., & Granado-Criado, J. M. (2013). A multiobjective swarm intelligence approach based on artificial bee colony for reliable DNA sequence design. *Engineering Applications of Artificial Intelligence, 26*, 2045–2057. (http://dx.doi.org/10.1016/j.engappai.2013.04.011).

Chen, A. H. L., Liang, Y.-C., & Liu, C.-C. (2012, June 10–15). An artificial bee colony algorithm for the cardinality-constrained portfolio optimization problems. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–8). Brisbane, Australia.

Cheng, X., & Jiang, M. (2012). An improved artificial bee colony algorithm based on Gussian mutation and chaos disturbance. In Y. Tan, Y. Shi, & Z. Ji (Eds.), *ICSI 2012, Part I* (Vol. 7331, pp. 326–333)., LNCS Berlin, Heidelberg: Springer.

Chong, C. S., Low, M. Y. H., Sivakumar, A. I., & Gay, K. L. (2006). A bee colony optimization algorithm to job shop scheduling. In *IEEE Proceedings of 2006 Winter Simulation Conference* (pp. 1954–1961).

Cicirello, V. A., & Smith, S. F. (2004). Wasp-like agents for distributed factory coordination. *Autonomous Agents and Multi-Agent Systems, 8*, 237–266.

Comellas, F., & Gallegos, R. (2005). Angels & mortals: A new combinatorial optimization algorithm. *Study on Fuzziness Soft Computing, 166*, 397–405.

Comellas, F., & Martínez-Navarro, J. (2009). Bumblebees: A multiagent combinatorial optimization algorithm inspired by social insect behaviour. In *Proceedings of First ACM/SIGEVO Summit on Genetic and Evolutionary Computation (GEC)* (pp. 811–814).

Davidović, T., Ramljak, D., Šelmić, M., & Teodorović, D. (2011). Bee colony optimization for the *p*-center problem. *Computers and Operations Research, 38*, 1367–1376.

Dhahri, H., Alimi, A. M., & Abraham, A. (2012, June 10–15). Designing beta basis function neural network for optimization using artificial bee colony (ABC). In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–7). Brisbane, Australia.

Dhurandher, S. K., Misra, S., Pruthi, P., Singhal, S., Aggarwal, S., & Woungang, I. (2011). Using bee algorithm for peer-to-peer file searching in mobile ad hoc networks. *Journal of Network and Computer Applications, 34*, 1498–1508.

El-Abd, M. (2010, July 18–23). A cooperative approach to the artificial bee colony algorithm. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 124–128). Barcelona, Spain: CCIB.

El-Abd, M. (2012a, June 10–15). Generalized opposition-based artificial bee colony algorithm. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–4). Brisbane, Australia.

El-Abd, M. (2012b). Performance assessment of foraging algorithms vs. evolutionary algorithms. *Information Sciences, 182*, 243–263.

Fan, H., & Zhong, Y. (2012). A rough set approach to feature selection based on wasp swarm optimization. *Journal of Computational Information Systems, 8*, 1037–1045.

Farooq, M. (2006). *From the wisdom of the hive to intelligent routing in telecommunication networks: A step towards intelligent network management through natural engineering*. Unpublished doctoral thesis, Universität Dortmund.

Farooq, M. (2009). *Bee-inspired protocol engineering: From nature to networks*. Berlin, Heidelberg: Springer. ISBN 978-3-540-85953-6.

Farooq, M., & Caro, G. A. D. (2008). Routing protocols for next-generation networks inspired by collective behaviors of insect societies: An overview. In C. Blum & D. Merkle (Eds.), *Swarm intelligence: Introduction and application* (pp. 101–160). Berlin, Heidelberg: Springer.

Figueira, J. R., & Talbi, E.-G. (2013). Emergent nature inspired algorithms for multi-objective optimization. *Computers and Operations Research, 40*, 1521–1523.

Fister, I., Fister, I. J., Brest, J., & Žumer, V. (2012, June 10–15). Memetic artificial bee colony algorithm for large-scale global optimization. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–8). Brisbane, Australia.

Forghany, Z., Davarynejad, M., & Snaar-Jagalska, B. E. (2012, June 10–15). Gene regulatory network model identification using artificial bee colony and swarm intelligence. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–6). Brisbane, Australia.

Gao, W., & Liu, S. (2011). Improved artificial bee colony algorithm for global optimization. *Information Processing Letters, 111*, 871–882.

Gao, W.-F., & Liu, S.-Y. (2012). A modified artificial bee colony algorithm. *Computers and Operations Research, 39*, 687–697.

Gao, W.-F., Liu, S.-Y., & Jiang, F. (2011). An improved artificial bee colony algorithm for directing orbits of chaotic systems. *Applied Mathematics and Computation, 218*, 3868–3879.

Gao, F., Fei, F.-X., Deng, Y.-F., Qi, Y.-B., & Balasingham, I. (2012). A novel non-Lyapunov approach through artificial bee colony algorithm for detecting unstable periodic orbits with high orders. *Expert Systems with Applications, 39,* 12389–12397. (http://dx.doi.org/10.1016/j.eswa.2012.04.083).

Gao, W., Liu, S., & Huang, L. (2012). A global best artificial bee colony algorithm for global optimization. *Journal of Computational and Applied Mathematics,* 236, 2741–2753.

Gao, W.-F., Liu, S.-Y., & Huang, L.-L. (2013). A novel artificial bee colony algorithm with Powell's method. *Applied Soft Computing, 13,* 3763–3775. (http://dx.doi.org/10.1016/j.asoc.2013.05.012).

Gorji-Bandpy, M., & Mozaffari, A. (2012). Multiobjective optimization of irreversible thermal engine using mutable smart bee algorithm. *Applied Computational Intelligence and Soft Computing, 12*, 1–13.

Goyal, S. (2012). The applications survey: Bee colony. *Engineering Science and Technology: An International Journal, 2*, 293–297.

Häckel, S., & Dippold, P. (2009, July 8–12). The bee colony-inspired algorithm (BCiA)–a two-stage approach for solving the vehicle routing problem with time windows. In *Proceedings of GECCO'09* (pp. 25–32). Montréal, Québec, Canada.

Haddad, O. B., Afshar, A., & Mariño, M. A. (2006). Honey-bees mating optimization (HBMO) algorithm: A new heuristic approach for water resources optimization. *Water Resources Management, 20*, 661–680.

Haddad, O. B., Afshar, A., & Mariño, M. A. (2009). Optimization of non-convex water resource problems by honey-bee mating optimization (HBMO) algorithm. *Engineering Computations: International Journal for Computer-Aided Engineering and Software: Practice and Experience, 26*, 267–280.

Haddad, O. B., Mirmomeni, M., & Mariño, M. A. (2010). Optimal design of stepped spillways using the HBMO algorithm. *Civil Engineering and Environmental Systems, 27*, 81–94.

Hancer, E., Ozturk, C., & Karaboga, D. (2012, June 10–15) Artificial bee colony based image clustering method. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–5). Brisbane, Australia.

Hemamalini, S., & Simon, S. P. (2010). Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions. *Electric Power Components and Systems, 38*, 786–803.

Hong, W.-C. (2011). Electric load forecasting by seasonal recurrent SVR (support vector regression) with chaotic artificial bee colony algorithm. *Energy, 36*, 5568–5578.

Horng, M.-H. (2010). A multilevel image thresholding using the honey bee mating optimization. *Applied Mathematics and Computation, 215*, 3302–3310.

Hsieh, T.-J., & Yeh, W.-C. (2012). Penalty guided bees search for redundancy allocation problems with a mix of components in series–parallel systems. *Computers and Operations Research, 39*, 2688–2704.

Jadhav, H. T., & Roy, R. (2013). Gbest guided artificial bee colony algorithm for environmental/economic dispatch considering wind power. *Expert Systems with Applications, 40,* 6385–6399. (http://dx.doi.org/10.1016/j.eswa.2013.05.048).

Janson, S., Middendorf, M., & Beekman, M. (2005). Honeybee swarms: How do scouts guide a swarm of uninformed bees. *Animal Behaviour, 70*, 349–358.

Jevtić, A., Gutiérrez, Á., Andina, D., & Jamshidi, M. (2012). Distributed bees algorithm for task allocation in swarm of robots. *IEEE Systems Journal, 6,* 296–304. (http://dx.doi.org/10.1109/JSYST.2011.2167820"10.1109/JSYST.2011.2167820).

Jones, K. O., & Bouffet, A. (2008, June 12–13). Comparison of bees algorithm, ant colony optimisation and particle swarm optimisation for PID controller tuning. In *Proceedings of International Conference on Computer Systems and Technologies (CompSysTech)* (pp. IIIA.9-1–IIIA.9-6). Gabrovo, Bulgaria.

Kalayci, C. B., & Gupta, S. M. (2013). Artificial bee colony algorithm for solving sequence-dependent disassembly line balancing problem. *Expert Systems with Applications, 40*, 7231–7241. (http://dx.doi.org/10.1016/j.eswa.2013.06.067).

Kang, F., Li, J., & Ma, Z. (2011). Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences, 181*, 3508–3531.

Kang, F., Li, J., & Li, H. (2013). Artificial bee colony algorithm and pattern search hybridized for global optimization. *Applied Soft Computing, 13*, 1781–1791.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization*. Technical Report TR06, Computer Engineering Department, Engineering Faculty, Erciyes University.

Karaboga, D., & Akay, B. (2009a) Artificial bee colony (ABC), harmony search and bees algorithms on numerical optimization. In *Proceedings of IPROMS Conference* (pp. 1–6).

Karaboga, D., & Akay, B. (2009b). A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation, 214*, 108–132.

Karaboga, D., & Akay, B. (2009c). A survey: Algorithms simulating bee swarm intelligence. *Artificial Intelligence Review, 31*, 61–85.

Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization, 39*, 459–471.

Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing, 8*, 687–697.

Karaboğa, N., & Çetinkaya, M. B. (2011). A novel and efficient algorithm for adaptive filtering: Artificial bee colony algorithm. *Turkey Journal of Electrical Engineering and Computer Science, 19*, 175–190.

Karaboga, D., Gorkemli, B., Ozturk, C., & Karaboga, N. (2012). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*. doi: 10.1007/s10462-012-9328-0.

Karabulut, K., & Tasgetiren, M. F. (2012, June 10–15). A discrete artificial bee colony algorithm for the traveling salesman problem with time windows. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–7). Brisbane, Australia.

Karsai, I. (1999). Decentralized control of construction behavior in paper wasps: An overview of the stigmergy approach. *Artificial Life, 5*, 117–136.

Karsai, I., & Wenzel, J. W. (2000). Organization and regulation of nest construction behavior in *Metapolybia* wasps. *Journal of Insect Behavior, 13*, 111–140.

Kashan, M. H., Nahavandi, N., & Kashan, A. H. (2012). DisABC: A new artificial bee colony algorithm for binary optimization. *Applied Soft Computing, 12*, 342–352.

Kaur, A., & Goyal, S. (2011). A bee colony optimization algorithm for fault coverage based regression test suite prioritization. *International Journal of Advanced Science and Technology, 29*, 17–29.

Khan, L., Ullah, I., Saeed, T., & Lo, K. L. (2010). Virtual bees algorithm based design of damping control system for TCSC. *Australian Journal of Basic and Applied Sciences, 4*, 1–18.

Koc, E., Ersoy, N., Camlidere, Z. S., & Kilic, H. (2012). A Web-service for automated software refactoring using artificial bee colony optimization. In Y. Tan, Y. Shi, & Z. Ji (Eds.), *ICSI 2012, Part I* (Vol. 7331, pp. 318–328)., LNCS Berlin, Heidelberg: Springer.

Kumar, S. K., Tiwari, M. K., & Babiceanu, R. F. (2010). Minimisation of supply chain cost with embedded risk using computational intelligence approaches. *International Journal of Production Research, 48*, 3717–3739.

Landa, J. T., & Tullock, G. (2003). Why ants do but honeybees do not construct satellite nests. *Journal of Bioeconomics, 5*, 151–164.

Latty, T., Duncan, M., & Beekman, M. (2009). High bee traffic disrupts transfer of directional information in flying honeybee. *Animal Behaviour, 78*, 117–121.

Lez-Álvarez, D. L. G., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2013). Comparing multiobjective swarm intelligence metaheuristics for DNA motif discovery. *Engineering Applications of Artificial Intelligence, 26*, 314–326.

Li, G., Niu, P., & Xiao, X. (2012). Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. *Applied Soft Computing, 12*, 320–332.

Lien, L.-C., & Cheng, M.-Y. (2012). A hybrid swarm intelligence based particle-bee algorithm for construction site layout optimization. *Expert Systems with Applications, 39*, 9642–9650.

Low, M. Y. H., Chandramohan, M., & Choo, C. S. (2009). Application of multi-objective bee colony optimization algorithm to automated red teaming. In *Proceedings of IEEE 2009 Winter Simulation Conference* (pp. 1798–1808).

Lucchetta, P., Bernstein, C., Théry, M., Lazzari, C., & Desouhant, E. (2008). Foraging and associative learning of visual signals in a parasitic wasp. *Animal Cognition, 11*, 525–533.

Lučić, P. (2002). *Modeling transportation problems using concepts of swarm intelligence and soft computing*. Unpublished doctoral thesis, Virginia Polytechnic Instituute and State University.

Lučić, P., & Teodorović, D. (2003). Computing with bees: Attacking complex transportation engineering problems. *International Journal on Artificial Intelligence Tools, 12*, 375–394.

Ma, M., Liang, J., Guo, M., Fan, Y., & Yin, Y. (2011). SAR image segmentation based on artificial bee colony algorithm. *Applied Soft Computing, 11*, 5205–5214.

Maia, R. D., Castro, L. N. D., & Caminhas, W. M. (2012, June 10–15). Bee colonies as model for multimodal continuous optimization: The OptBees algorithm. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–8). Brisbane, Australia.

Marinakis, Y., Marinaki, M., & Dounias, G. (2008). Honey bees mating optimization algorithm for the vehicle routing problem. *Studies in Computational Intelligence (SCI), 129*, 139–148. (Berlin, Heidelberg: Springer).

Marinakis, Y., Marinaki, M., & Dounias, G. (2010). Honey bees mating optimization algorithm for large scale vehicle routing problems. *Natural Computing, 9*, 5–27.

Marinakis, Y., Marinaki, M., & Dounias, G. (2011). Honey bees mating optimization algorithm for the Euclidean traveling salesman problem. *Information Sciences, 181*, 4684–4698.

Mccaffrey, J. D., & Dierking, H. (2009). An empirical study of unsupervised rule set extraction of clustered categorical data using a simulated bee colony algorithm. In G. Governatori, J. Hall, & A. Paschke (Eds.), *RuleML 2009* (Vol. 5858, pp. 182–193)., LNCS Berlin, Heidelberg: Springer.

Mezura-Montes, E., & Velez-Koeppel, R. E. (2012, July 18–23). Elitist artificial bee colony for constrained real-parameter optimization. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 2068–2075). Barcelona, Spain: CCIB.

Mezura-Montes, E., Damián-Araoz, M., & Cetina-Domíngez, O. (2010, July 18–23). Smart flight and dynamic tolerances in the artificial bee colony for constrained optimization. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 4118–4125). Barcelona, Spain: CCIB.

Muñoz, M. A., López, J. A., & Caicedo, E. (2009). An artificial beehive algorithm for continuous optimization. *International Journal of Intelligent Systems, 24*, 1080–1093.

Nakrani, S., & Tovey, C. (2003, December 15–17). On honey bees and dynamic allocation in an Internet server colony. In *Proceedings of 2nd International Workshop on the Mathematics and Algorithms of Social Insects* (pp. 1–8). Atlanta, Georgia.

Nakrani, S., & Tovey, C. (2004). On honey bees and dynamic server allocation in internet hosting centers. *Adaptive Behavior, 12*, 223–240.

Narasimhan, H. (2009). Parallel artificial bee colony (PABC) algorithm. In *IEEE World Congress on Nature and Biologically Inspired Computing (NaBIC)* (pp. 306-311).

Niknam, T. (2009). An efficient hybrid evolutionary algorithm based on PSO and HBMO algorithms for multi-objective distribution feeder reconfiguration. *Energy Conversion and Management, 50*, 2074–2082.

Niknam, T. (2011). An efficient multi-objective HBMO algorithm for distribution feeder reconfiguration. *Expert Systems with Applications, 38*, 2878–2887.

Niknam, T., & Golestaneh, F. (2013). Enhanced bee swarm optimization algorithm for dynamic economic dispatch. *IEEE Systems Journal, 7*, 754–762. doi:10.1109/JSYST.2012.2191831.

Niknam, T., Olamaie, J., & Khorshidi, R. (2008). A hybrid algorithm based on HBMO and fuzzy set for multi-objective distribution feeder reconfiguration. *World Applied Sciences Journal, 4*, 308–315.

Niknam, T., Mojarrad, H. D., Meymand, H. Z., & Firouzi, B. B. (2011). A new honey bee mating optimization algorithm for non-smooth economic dispatch. *Energy, 36*, 896–908.

Niknam, T., Fard, A. K., & Seifi, A. (2012). Distribution feeder reconfiguration considering fuel cell-wind-photovoltaic power plants. *Renewable Energy, 37*, 213–225.

Olamaei, J., Niknam, T., Badali, S., & Arefi, (2012). Distribution feeder reconfiguration for loss minimization based on modified honey bee mating optimization algorithm. *Energy Procedia, 14*, 304–311.

Omkar, S. N., Senthilnath, J., Khandelwal, R., Naik, G. N., & Gopalakrishnan, S. (2011). Artificial bee colony (ABC) for multi-objective design optimization of composite structures. *Applied Soft Computing, 11*, 489–499.

Pham, D. T., & Castellani, M. (2009). The bees algorithm: Modelling foraging behaviour to solve continuous optimization problems. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 223*, 2919–2938.

Pham, D. T., & Ghanbarzadeh, A. (2007). Multi-objective optimisation using the bees algorithm. In *Third International Virtual Conference on Intelligent Production Machines and Systems (IPROMS)* (pp. 1–5). Dunbeath, Scotland: Whittles.

Pham, D. T., & Koç, E. (2010). Design of a two-dimensional recursive filter using the bees algorithm. *International Journal of Automation and Computing, 7*, 399–402.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., & Zaidi, M. (2006). The bees algorithm—a novel tool for complex optimisation problems. In *Proceedings of Second International Virtual Conference on Intelligent production machines and systems (IPROMS)* (pp. 454–459). Oxford: Elsevier.

Pham, D. T., Afify, A. A., & Koç, E. (2007a). Manufacturing cell formation using the bees algorithm. In *Third International Virtual Conference on Intelligent Production Machines and Systems (IPROMS)* (pp. 1–6). Dunbeath, Scotland: Whittles.

Pham, D. T., Ghanbarzadeh, A., Koç, E., Otri, S., Rahim, S., & Zaidi, M. (2007b). Using the bees algorithm to schedule jobs for a machine. In *Proceedings of Eighth International Conference on Laser metrology, CMM and machine tool performance (LAMDAMAP)* (pp. 430–439). Euspen, UK: Cardiff.

Pham, D. T., Ghanbarzadeh, A., Otri, S., & Koç, E. (2009). Optimal design of mechanical components using the bees algorithm. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 223*, 1051–1056.

Pinto, P., Runkler, T. A., & Sousa, J. M. (2005). Wasp swarm optimization of logistic systems. In B. Ribeiro, R. F. Albrecht, A. Dobnikar, D. W. Pearson, & N. C. Steele (Eds.), *International Conference on Adaptive and Natural Computing Algorithms, Coimbra, Portugal* (pp. 264–267). Wien: Springer.

Pinto, P. C., Runkler, T. A., & Sousa, J. M. C. (2006, September 11–13). Agent based optimization of the MAX-SAT problem using wasp swarms. In *7th Portuguese Conference on Automatic Control (CONTROLO)* (pp. 1–6). Lisboa, Portugal: Instituto Superior Técnico.

Quijano, N., & Passino, K. M. (2010). Honey bee social foraging algorithms for resource allocation: Theory and application. *Engineering Applications of Artificial Intelligence, 23*, 845–861.

Rajasekhar, A., Pant, M., & Abraham, A. (2011). Cauchy movements for artificial bees for finding better food sources. In *Third World Congress on Nature and Biologically Inspired Computing (NaBIC)* (pp. 279–284).

Rajasekhar, A., Das, S., & Suganthan, P. N. (2012, June 10–15). Design of fractional order controller for a servohydraulic positioning system with micro artificial bee colony algorithm. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–8). Brisbane, Australia.

Ramírez, F. J., Lee, J. Y., Packianather, M. S., & Pham, D. T. (2010, November 15–26). Enhancing multi-stage deep-drawing processes through the novel use of the bees-algorithm. In *Proceedings of 6th IPROMS Virtual Conference* (pp. 1–6).

Runkler, T. A. (2008). Wasp swarm optimization of the c-means clustering model. *International Journal of Intelligent Systems, 23*, 269–285.

Samanta, S., & Chakraborty, S. (2011). Parametric optimization of some non-traditional machining processes using artificial bee colony algorithm. *Engineering Applications of Artificial Intelligence, 24*, 946–957.

Sato, T., & Hagiwara, M. (1997). Bee system: Finding solution by a concentrated search. In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 3954–3959).

Scholz-Reiter, B., Jagalski, T., & Bendul, J. C. (2008). Autonomous control of a shop floor based on bee's foraging behaviour. In H.-D. Haasis (Ed.), *Dynamics in logistics* (pp. 415–423). Berlin: Springer.

Senthilnath, J., Omkar, S. N., Mani, V., Tejovanth, N., Diwakar, P. G., & Archana, S. B. (2011). Multi-spectral satellite image classification using glowworm swarm optimization. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 47–50).

Shafia, M. A., Moghaddam, M. R., & Tavakolian, R. (2011). A hybrid algorithm for data clustering using honey bee algorithm, genetic algorithm and *k*-means method. *Journal of Advanced Computer Science and Technology Research, 1*, 110–125.

Sharma, T. K., Pant, M., & Bansal, J. C. (2012a, June 10–15). Artificial bee colony with mean mutation operator for better exploitation. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–7). Brisbane, Australia.

Sharma, T. K., Pant, M., & Bansal, J. C. (2012b, June 10–15). Some modifications to enhance the performance of artificial bee colony. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–8). Brisbane, Australia.

Shi, X., Li, Y., Li, H., Guan, R., Wang, L., & Liang, Y. (2010). An integrated algorithm based on artificial bee colony and particle swarm optimization. In *Sixth International Conference on Natural Computation (ICNC)* (pp. 2586–2590).

Slaa, E. J., & Hughes, W. O. H. (2009). Local enhancement, local inhibition, ravesdropping, and the parasitism of social insect communication. In S. Jarau & M. Hrncir (Eds.), *Ecological, behavioral, and theoretical approaches, Chapter 8* (pp. 147–164). Boca Raton, FL: CRC Press, Taylor & Francis Group, LLC. ISBN ISBN 978-1-4200-7560-1.

Song, J., Hu, J., Tian, Y., & Xu, Y. (2005, September 13–16). Re-optimization in dynamic vehicle routing problem based on wasp-like agent strategy. In *Proceedings of 8th International Conference on Intelligent Transportation Systems* (pp. 688–693). Vienna, Austria.

Sotelo-Figueroa, M. A., Baltazar-Flores, M. D. R., Carpio, J. M., & Zamudio, V. (2010). A comparation between bee swarm optimization and greedy algorithm for the knapsack problem with bee reallocation. In *Ninth Mexican International Conference on Artificial Intelligence* (pp. 22–27).

Subotic, M., Tuba, M., & Stanarevic, N. (2010). Parallelization of the artificial bee colony (ABC) algorithm. In *Recent Advances in Neural Networks, Fuzzy Systems and Evolutionary Computing* (191–196).

Subramanian, S., Anandhakumar, R., & Ganesan, S. (2011). Generator maintenance management using bio-inspired search algorithm. *International Journal of Energy Sector Management, 5*, 522–544.

Sundar, S., & Singh, A. (2012). A swarm intelligence approach to the early/tardy scheduling problem. *Swarm and Evolutionary Computation, 4*, 25–32.

Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Chen, A. H.-L. (2010, July 18–23) A discrete artificial bee colony algorithm for the permutation flow shop scheduling problem with total flowtime criterion. In *IEEE World Congress on Computational Intelligence* (pp. 137–144). Barcelona, Spain: CCIB.

Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Chen, A. H.-L. (2011). A discrete artificial bee colony algorithm for the total flowtime minimization in permutation flow shops. *Information Sciences, 181*, 3459–3475.

Tasgetiren, M. F., Pan, Q.-K., Suganthan, P. N., & Oner, A. (2013). A discrete artificial bee colony algorithm for the no-idle permutation flowshop scheduling problem with the total tardiness criterion. *Applied Mathematical Modelling, 37*, 6758–6779.

Teodorović, D. (2008). Swarm intelligence systems for transportation engineering: Principles and applications. *Transportation Research Part C, 16*, 651–667.

Teodorović, D. (2009a). Bee colony optimization (BCO). In C. P. Lim, L. C. Jain, & S. Dehuri (Eds.), *Innovations in swarm intelligence* (Vol. 248, pp. 39–60)., SCI Berlin, Heidelberg: Springer.

Teodorović, D. (2009b). Bee colony optimization (BCO). In C. P. Lim, L. C. Jain, & S. Dehuri (Eds.), *Innovations in swarm intelligence*. Berlin, Heidelberg: Springer.

Teodorović, D., & Dell'Orco, M. (2005). Bee colony optimization: A cooperative learning approach to complex transportation problems. In *16th Mini-EURO Conference on Advanced OR and AI Methods in Transportation* (pp. 51–60).

Teodorovic, D. U. Š. A. N., Davidovic, T., & Selmic, M. (2011). Bee colony optimization: The applications survey. *ACM Transactions on Computational Logic, 1529*, 3785.

Tereshko, V., & Lee, T. (2002). How information mapping patterns determine foraging behaviour of a honeybee colony. *Open Systems and Information Dynamics, 9*, 181–193.

Theraulaz, G., & Bonabeau, E. (1995). Coordination in distributed building. *Science, 269*, 686–688.

Theraulaz, G., Goss, S., Gervet, J., & Deneubourg, J. L. (1991). Task differentiation in polistes wasps colonies: A model for self-organizing groups of robots. In *First International Conference on Simulation of Adaptive Behavior* (pp. 346–355). Cambridge, MA: MIT Press.

Tien, J. P., & Li, T. H. S. (2012). Hybrid Taguchi-chaos of multilevel immune and the artificial bee colony algorithm for parameter identification of chaotic systems. *Computers and Mathematics with Applications, 64*(5), 1108–1119. doi:10.1016/j.camwa.2012.03.029.

Tsai, P.-W., Pan, J.-S., Liao, B.-Y., & Chu, S.-C. (2009). Enhanced artificial bee colony optimization. *International Journal of Innovative Computing, Information and Control, 5*, 1–12.

Vitorino, L. N., Ribeiro, S. F., & Bastos-Filho, C. J. A. (2012, June 10–15). A hybrid swarm intelligence optimizer based on particles and artificial bees for high-dimensional search spaces. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–6). Brisbane, Australia.

Wang, Z. (2009). *Cooperative construction*. Unpublished master thesis, The Ohio State University.

Wang, D.-Z., Zhang, J.-S., Wan, F., & Zhu, L. (2006, May 27–29). A dynamic task scheduling algorithm in grid environment. In *5th WSEAS International Conference on Telecommunications and Informatics* (pp. 273–275). Istanbul, Turkey.

Wedde, H. F., & Farooq, M. (2006). A comprehensive review of nature inspired routing algorithms for fixed telecommunication networks. *Journal of Systems Architecture, 52*, 461–484.

Wedde, H. F., Farooq, M., & Zhang, Y. (2004). Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. In M. Dorigo (Ed.), *ANTS 2004* (Vol. 3172, pp. 83–94)., LNCS Berlin, Heidelberg: Springer.

Wong, L.-P., Puan, C. Y., Low, M. Y. H., & Chong, C. S. (2008) Bee colony optimization algorithm with big valley landscape exploitation for job shop scheduling problems. In *Proceedings of IEEE 2008 Winter Simulation Conference* (pp. 2050–2058).

Xu, C., Duan, H., & Liu, F. (2010a). Chaotic artificial bee colony approach to uninhabited combat air vehicle (UCAV) path planning. *Aerospace Science and Technology, 14*, 535–541.

Xu, S., Ji, Z., Pham, D. T., & Yu, F. (2010b). Bio-inspired binary bees algorithm for a two-level distribution optimisation problem. *Journal of Bionic Engineering, 7*, 161–167.

Yang, X. S. (2005). Engineering optimizations via nature-inspired virtual bee algorithms. In José Mira José & R. Álvarez (Eds.), *Artificial intelligence and knowledge engineering applications: A bioinspired approach*. Berlin Heidelberg: Springer.

Yildiz, A. R. (2013). A new hybrid artificial bee colony algorithm for robust optimal design and manufacturing. *Applied Soft Computing, 13*, 2906–2912. (http://dx.doi.org/10.1016/j.asoc.2012.04.013).

Zeng, F., Decraene, J., Low, M. Y. H., Hingston, P., Cai, W., Zhou, S., & Chandramohan, M. (2010, July 18–23). Autonomous bee colony optimization for multi-objective function. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1279–1286). Barcelona, Spain: CCIB.

Zhang, H., & Ye, D. (2012). An artificial bee colony algorithm approach for routing in VLSI. In Y. Tan, Y. Shi, & Z. Ji (Eds.), *ICSI 2012, Part I* (Vol. 7331, pp. 334–341)., LNCS Berlin, Heidelberg: Springer.

Zhang, W., Wang, N., & Yang, S. (2013). Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell. *International Journal of Hydrogen Energy, 38*, 5796–5806.

Zhang, H., Zhu, Y., & Yan, X. (2102, June 10–15). Multi-hive artificial bee colony algorithm for constrained multi-objective optimization. In *IEEE World Congress on Computational Intelligence (WCCI)* (pp. 1–8). Brisbane, Australia.