International Journal of Innovative Computing, Information and Control
Volume **16**, Number **6**, December **2020**

ICIC International ©2020 ISSN 1349-4198
pp. 2063–2081

STINGLESS BEE ALGORITHM FOR NUMERICAL OPTIMIZATION PROBLEMS

Endra Joelianto[1,*], Amrizal Nainggolan[2] and Yosi Agustina Hidayat[3]

[1]Instrumentation and Control Research Group, Faculty of Industrial Technology
[2]Instrumentation and Control Master Program, Faculty of Industrial Technology
[3]Industrial Systems and Techno-Economics Research Group, Faculty of Industrial Technology
Institut Teknologi Bandung
Jalan Ganesha 10, Bandung 40132, West Java, Indonesia
[*]Corresponding author: ejoel@tf.itb.ac.id; rizal.ngl@gmail.com; yosi@mail.ti.itb.ac.id

Received April 2020; revised October 2020

Abstract. *Nature-inspired algorithms by mimicking nature strategies applicable to sort out hard optimization problems have received much attention by way of more efficient and comprehensive search mechanisms. The well-known artificial bee colony (ABC) is generated by imitating foraging behavior of honey bees for food sources. Many bee colonies exist in nature with different foraging behaviors as colony optimum mechanisms. This paper considers the stingless bee algorithm (SBA) to enrich swarm intelligence algorithm varieties from bee colonies, to explore the distinct foraging behaviors of stingless bee colony into search-based algorithm, and to know the capability of the SBA in numerical optimization problems. The developed SBA is applied in solving various numerical optimization problems results in high efficiency in acquiring a near-optimal solution. To acquire the algorithm performances, the proposed SBA is measured up with the ABC. The numerical results present SBA excels in average function evaluation to a solution (AES) and sum of errors (SE) criteria. In contrast, ABC exhibits better performances in success rate (SR) and final error (FE) criteria. The performance evaluations show bee colonies naturally develop optimal strategies in response to their environment and understanding of various foraging behaviors is practicable for constructing optimal bee colony algorithms.*

Keywords: Foraging behaviors, Swarm intelligence, Stingless bee algorithm, Numerical optimization, Artificial bee colony

1. **Introduction.** The problem of numerical optimization has received considerable attention by many researchers for the discussion and development of suitable algorithms. The awareness is widespread in many fields ranging from engineering, economics, industry and so on [1, 2, 3, 4, 5]. Optimization problems can be divided into numerical optimization problems (continuous) and combinatorial optimization problems (discrete) based on the resulted solutions [6, 7]. In contrast, the optimization problem-solving algorithm comprises two groups, i.e., deterministic and meta-heuristic (stochastic). If an algorithm works in a deterministic way in the absence of a random process, it is classified by a deterministic group. Otherwise, if a random process exists in the process, it is classified into a meta-heuristic group [6, 7].

In particular, numerical optimization can be described as the exploration and formulation of algorithms to minimize or maximize complex nonlinear functions with variables having nonlinear constraints. It is also categorized as part of numerical and mathematical

DOI: 10.24507/ijicic.16.06.2063

analysis. The use of optimization in practical problems is going up; for this reason, almost all problems in engineering, science, economics and finance, and health are resolved through numerical optimization. To illustrate, the application of numerical optimization in engineering aims to pervasive problems in rigid body dynamics [8], silicon-on-insulator finfet performance improvement [9], active filter design [10], handset antennas [11], microwave structures space mapping design [12], optimal sensor deployment [13], resource allocation optimization strategy in multi-radio and multi-channel wireless sensor networks (WSNs) [14], optimal control [15], model predictive control (MPC) for helicopter models [16] and multiple target tracking [17].

Furthermore, optimization methods have been widely put into operation in computational systems biology [18]. Economists have been modeling dynamic decisions over time using the control theory since 1970 [19], understanding labor market behavior using dynamic search models [20]. Macroeconomists construct dynamic stochastic general equilibrium models that describe the overall behavior of economic dynamics as a result of optimization of decisions that depend on consumers, investors, workers, and the government [21]. Other applications can be found, to illustrate, in health care [22], precision farming [23], traffic lights cycles and green period ratios [24], prediction of traffic flow in the short term [25], scheduling in the storm nodes in edge computing environments [26], and feature selection in pre-processing of data [27]. With the increase of capacity and faster computation speed of computers, many real problems were straightforwardly solved by means of numerical optimization. This fosters intensive development of numerical optimization algorithms.

Swarm intelligence and evolution-based algorithms are parts of the meta-heuristic algorithm that recently gained researchers' attention. The algorithm tends to be more reliable at solving optimization problems in the real world than the conventional (deterministic) optimization techniques in dealing with complexity and high uncertainties. Swarm intelligence is an attempt to create a distributed problem-solving algorithm based on the inspiration of the mutual behaviors of insect colonies or other social animals [28]. A number of the best known evolution-based or swarm intelligence-based algorithms were reviewed in [29], in which artificial bee colony algorithm known as ABC initiated by Karaboga [30] gained very high popularity. The ABC algorithm has been known as a formidable population based meta heuristic for numerical optimization and appeared to match with other swarm intelligence based algorithms for numerical optimization [4, 5].

ABC is a swarm-based algorithm recently introduced which mimics the behavior of the honeybee [30, 31] based on the mode of communication, various nest locations, mating, reproduction, task allocation, waggle dance, pheromone position, and movement for problem solving purposes by varying algorithm consistent with problem requirements. The ABC algorithm is reliable in finding optimal solutions by means of iteratively searches with excellent accuracy [32]. However, the ABC algorithm still exhibits a weakness in computing time or long iterations. Since it was first introduced, ABC was extensively researched, developed, improved and varied [33, 34, 35, 36, 37, 38]. Some ABC variants include ABC for optimization problems with constraints [39], modified ABC for optimization problems with constraints [40], combinatorial ABC [41] and discrete ABC [42] for discrete optimization, quick ABC [43], modified cooperative learning ABC [44], and others [45, 46, 47, 48] were developed in order to gain faster convergence speed.

The paper extents the stingless bee algorithm (SBA) for the application in numerical optimization problems inspired by several efficient and effective characteristics of stingless bee foraging behaviors in nature. The algorithm is aimed to exhibit a better efficiency in finding solutions that are close to optimal. Then, to measure its performance, the algorithm performance is compared with the ABC algorithm. The paper is organized as

follows. Section 2 describes the related work of the algorithm inspired by bee foraging behavior and briefly summarizes the ABC algorithm for a brief overview. Section 3 briefly summarizes the ABC algorithm. Section 4 describes the foraging behavior of real stingless bees and the development of SBA inspired by their natural behavior. Section 5 shows the experimental setup of the SBA algorithm to the well-known benchmark functions in numerical optimization and discusses the performance of the proposed SBA algorithm compared to ABC. As a final point, Section 6 summarizes the results and provides direction for future work.

2. **Related Work.** Two groups of bees that demonstrate a high level of sociality are honey bees (Apini) and stingless bees (Meliponini). At least 11 species of honey bees exist in one genus, and several hundred species of stingless bee are spread in more than 36 genera [49]. Among those species, at least 22 species of stingless bees and 3 species of honey bees can be found in West Sumatra, Indonesia. A dependency between the colony needs and individual behaviors and vice versa exists, especially by the members of society (foragers), in order to maintain contact with environments. Individual foraging decisions are also influenced by their behavioral history. These lead to a unique foraging behavior related to material interest, pursuit quantity, and longevity [50, 51, 52]. Stingless bees exhibit behavior diversity to respond food sources, such as odor marks, social assistance, and foremost foragers [53, 54, 55, 56, 57, 58, 59]. In addition, vibration signals were observed as part of the communication for foraging related to the food source [60, 61].

The foraging behaviors of three Sumatran stingless bee species studied in [62] inspired the development of a firstly published article on stingless bee algorithm (SBA) [63], to solve a combinatorial optimization problem from master thesis [64] based on stingless bee behaviors described in [62]. The SBA is considered to be capable of accelerating the convergence process to the optimum point than other bee inspired algorithms [63]. Moreover, the proposed algorithm needs to be improved for combinatorial applications and for common numerical optimization problems. For real-time applications that require high optimization speeds, in which the length of iterations increases when the number of evaluation functions increases, obtaining a faster near-optimal solution is necessary. This faster near-optimal solution is desirable than the actual but longer optimal solution. Therefore, in this paper, the considered SBA is intended to reduce fewer effective algorithms by a system breakthrough that requires an improved convergence and speed performance for the near-optimal solutions.

The acquired SBA, in this paper, is also aimed to enlarge the variety of swarm intelligence based algorithm from bee colonies, to explore knowledge on stingless bee behaviors from nature, to investigate the distinct foraging behaviors of stingless bee colony into search based algorithm, and to know the capability of the SBA in solving numerical optimization problems. Then, the improved SBA is tested to solve various types of numerical optimization problems consisting of 50 well-known benchmark functions in finding a near-optimal solution without discounting the quality of solutions. In order to gain the performances of the algorithm, the SBA is then measured up with the ABC. The numerical results present the performances of SBA related to ABC. It is of interest to compare the proposed SBA with ABC and all its variants in order to gain full comparison. However, the objectives of the paper are limited to place the attention only for the natural stingless bee behavior investigation based on [62] for the search algorithm. Therefore, the comparison is only be made to the ABC algorithm with no extension and/or improvement for comparable performance evaluation.

3. **Artificial Bee Colony (ABC).** The initial version of ABC [30] will be described in this paper because the SBA presented in this paper is also the first version for solving the optimization of numerical problems. Four main parts are present in the ABC algorithm. They are the initialization or random solution generator, onlooker bee phase, employed bee phase, and scout bee phase. The key steps of the algorithm are in this way:

1. Initialize
2. **repeat**
3. Employed bee phase
4. Onlooker bee phase
5. Scout bee phase
6. **until** requirements are met

Food sources correspond to solution candidates of the optimization problem. First, any optimization problems need to be converted to minimization problems. At the initialization step, the food sources are produced randomly by (1) where $j \in \{1, 2, \ldots, D\}$. Each solution candidate x_i $(i = 1, 2, \ldots, SN)$ is a vector of D-dimensional where SN denotes the number of available food sources or the size of employed bees. Next, $x^j_{\max}$ and $x^j_{\min}$ are upper bound (ub) and lower bound (lb) that describe the boundary constraints of the optimization problem [32, 39].

$$x^j_i = x^j_{\min} + rand[0, 1]\left(x^j_{\max} - x^j_{\min}\right) \tag{1}$$

Employed bee and onlooker bee phases are the variation operators, and they modify current solutions as described in (2). x_k denotes randomly selected food source other than x_i, and $j \in \{1, 2, \ldots, D\}$ indicates a randomly selected index. If the new solution candidate (v_{ij}) exhibits better quality, the food source is then switched to the new solution candidate ($x_{ij} = v_{ij}$) [32, 39].

$$v_{ij} = x_{ij} + rand[-1, 1](x_{ij} - x_{kj}) \tag{2}$$

The difference between the employed bee and onlooker bee is that the employed bee modifies each food source once in every cycle, while the onlooker bee is assigned to go for the food source to be exploited based on their probability. The probability for the chosen food source is defined in (3), and the fitness scaling function is defined in (4). f represents the objective function of the minimization problem [30, 37].

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{3}$$

$$fit_i = \begin{cases} \dfrac{1}{1 + f(x_i)} & \text{if } f(x_i) \geq 0 \\ 1 + |f(x_i)| & \text{else} \end{cases} \tag{4}$$

The variation operator will take place at each cycle step for every employed bee and onlooker bee. If any food sources that are unable to improve the fitness value for L times exist, the specific food source will be left behind, and the colony sends scout bees to find another potential food source. Scout bee mechanism is defined in (1).

4. Stingless Bee Foraging Behaviors Based Algorithm.

4.1. Stingless bee behavior.

The stingless bee habit cycles/behaviors begin with being a naive food seeker and turn into a scout if genuine information exists, or unemployed if the information coming from other bees triggers to start searching foods from a source that has been known. Bees that are found and are committed to exploit a food source either from scouting or from unemployed bees are referred as employed bees. At the end

of the day, the bees will temporarily become unemployed bees. The next day, a bee can visit the previous food source, be recruited by other bees or seeks out another food source (scout). Bees that find a potential food source will pass on the information to other bees to gain recruitment to the food source [50]. The photograph of stingless bees with nest can be seen in the literature, such as [65].

In honey bee, bees that find a location of rich food sources will convey the existence of the source to other bees in the nest with a dance (waggle dance). This dance will trigger unemployed bees who see it to go to the source to do the exploitation [66, 67]. In the stingless bee, the information is transferred to other bees in various ways, such as touch, movement, odors, and sound [68]. Chemicals produced from food sources or excreted by employed bees are the most important source of information on the location and exploitation of a food source and as a recruitment tool [69]. Some species of stingless bees place odor at some points on solids around the food source and also along the route back to the nest. This odor becomes the main signal involved in the recruitment communication process and strongly attracts unemployed bees to find the source which will improve the efficiency of the exploitation process.

In the exploitation phase, recruited bees will proceed to food sources to exploit. In the location/site, in addition to do the exploitation, bees will find the source in the surrounding which exhibits better quality and quantity. In Trigona Amalthea, bees prefer to visit the location occupied by their nest-mates at the first time, but the next will stay away from nest-mates [70]. Bees that exploit a food source will release odors or chemicals as a marker of the food source along with the food source quality information. These bees learn to associate the quality of food sources in a positive or negative way based on experience in exploiting previous food sources. Odor marker works as a repellent if the quality of the food source is lower; otherwise it acts as an attractant if the quality is better [71]. The bees' decision to choose an exploitation site is based on the presence of odors, instead of directly performing testing [72].

4.2. **Stingless bee algorithm.** In evolutionary algorithms, search capability is determined by exploration and exploitation. Exploration pays attention to broad-first search, while exploitation emphasizes in-depth-first search. Therefore, performances of evolutionary algorithms are greatly influenced by the ability to compromise exploration and exploitation throughout the optimization process [73]. In nature, the onlooker of honey-bee phase puts more attention to the exploitation or the deep-first search in which honey bee workers who exploit a food source tend to continue exploitation until a few days after the food at the site runs out [62, 66]. This makes the original search equation in ABC poor at exploitation [73, 74]. This weakness can be overcome by the ability of the stingless bee to utilize local information in the area of exploitation that is with the odor [72]. In stingless bees, compared to honey bees, the bees can quickly turn exploitation strategies into exploration. This seamless change in the stingless bee is useful for improving the efficiency of food search as well as being a deficiency in making use of locations with abundant food sources [67]. Bees will tend to stop exploiting the location before the food that is exhausted. Insufficiency food sources intrinsically will trigger experienced bees to find new food sources (scouts). These bees can search for new sources or visit old sources again to investigate [50]. Thus it forms a cycle of colony behaviors of stingless bee as shown in Figure 1.

Basic similarities exist between ABC and SBA and some differences will be elaborated in this section. SBA exhibits the same main algorithm steps as ABC. The main different between the two algorithms is *the variation operator in the exploitation phase* which in turn provides improvement over the ABC algorithm by including new control parameters

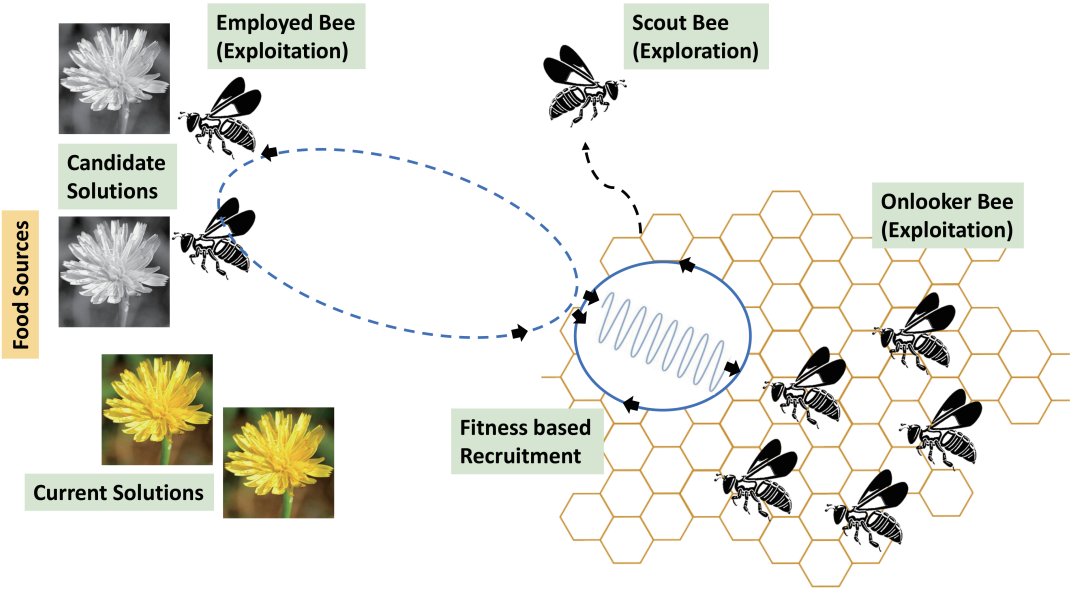

FIGURE 1. Stingless bees foraging behaviors

involved. The overall comparison is presented in Figure 2. The left flowchart denotes the standard ABC algorithm while the right flowchart is the SBA flowchart. As the ABC algorithm exists for years, the SBA flowchart is formed by referring to the ABC algorithm. The main steps from the stingless bee behaviors adopted in the SBA are described as modified and new procedures, and these procedures do not exist in the ABC algorithm. The modified procedures describe the procedures which exhibit similar characteristics to the ABC procedure, but they are modified according to the characteristics of stingless bees. Conversely, the new procedures represent behaviors of stingless bees which are different from honey bees and added to the standard algorithm of ABC.

In the following, the main steps of the SBA are described as:

1. Initialize
2. **repeat**
3. Employed bee phase (Exploitation)
4. Onlooker bee phase (Exploitation)
5. Scout bee phase (Exploration)
6. **until** requirements are met

4.2.1. *Initialization.* In both ABC and SBA, the location of the site or food source represents the candidate solutions of the optimization problem. First, any optimization problems are converted to minimization problems. Each candidate solution is a vector with dimensions D, x_{ij} $(j = 1, 2, \ldots, D)$, with i showing the index of each candidate solution $(i = 1, 2, \ldots, SN)$ where SN is the number of food sources (sites number). At the initialization phase, the solution candidate is defined as in Equation (1). Then, the solution candidates are evaluated in their fitness values as a representation of the number and quality of nectars at the flower being exploited. Each time, a bee evaluates the fitness value at a food source, the bees will spread odor at that location. Aside from being a location marker, odor also provides information on the food source quality of the site.

4.2.2. *Exploitation.* This phase consists of two parts: employed bees and unemployed bees. The employed bee is a bee aimed at exploiting a particular site. The number of employed bees equals the number of the sites. In this phase, each site will be exploited

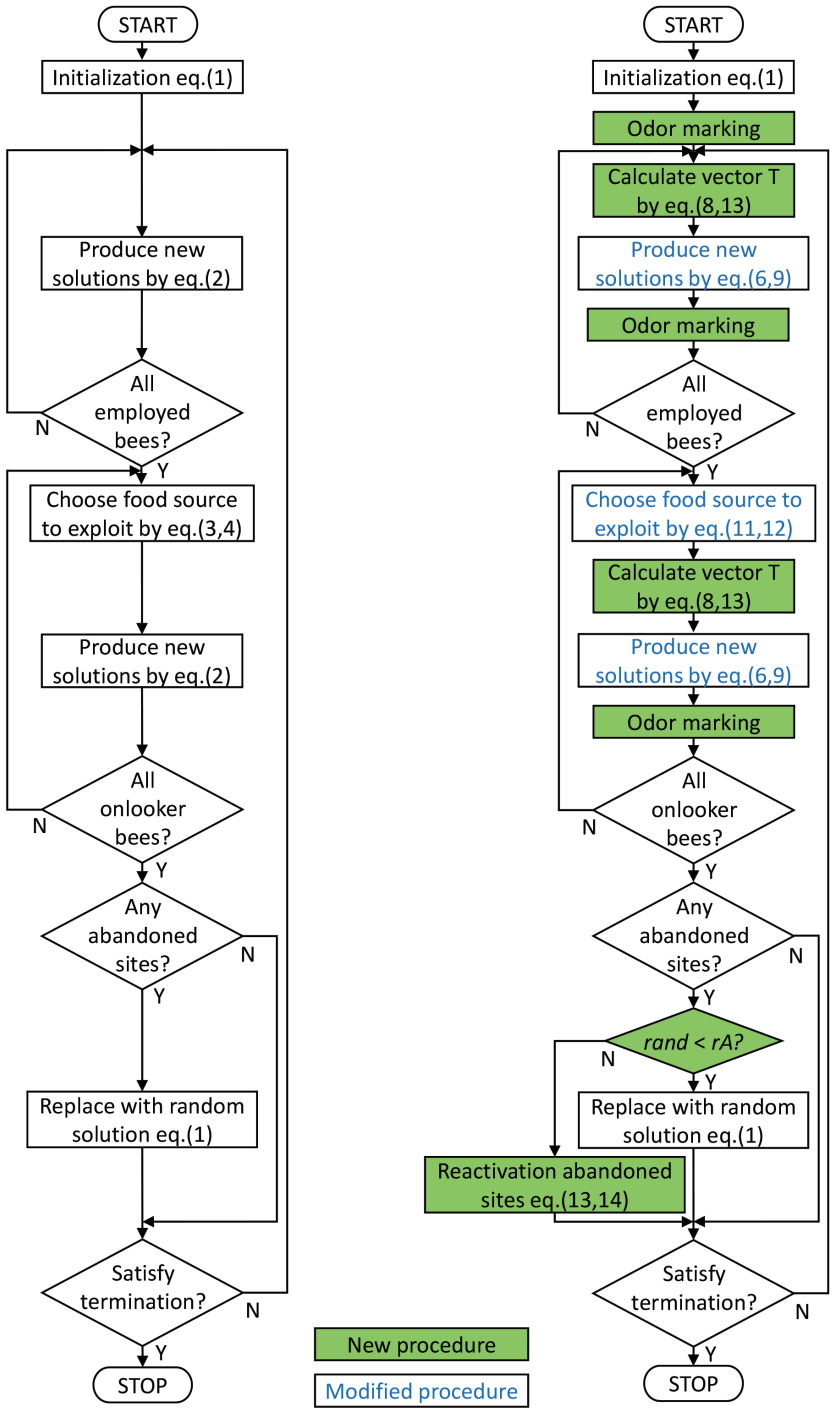

FIGURE 2. ABC (left) and SBA (right) flowchart comparison

one time each. When a bee visits a site, the bee will exploit the area with the purpose of increasing the quantity and quality of the foods.

When an artificial bee visits a site, if the bee does not find enough odor information around the site to get new candidates, it is represented by Equations (5) and (6). If the bee does get odor information, then it will be used to obtain the $\mathbf{T}_i$ vector (exploitation direction), as defined in Equations (7) and (8). The variation mechanism uses the equation described in (6).

$$\mathbf{T}_i = rand[-1, 1]R \tag{5}$$

$$v_{ij} = x_{ij} + rand[0,1]\mathbf{T}_{ij} \tag{6}$$

$$r_{iv} = \begin{cases} x'_{iv} - x_i & \text{if } f(x'_{iv}) > f(x_i)\ (repellent) \\ -(x'_{iv} - x_i) & \text{if } f(x'_{iv}) < f(x_i)\ (attractant) \end{cases} \tag{7}$$

$$\mathbf{T}_i = -\sum_{v=1}^{N_v} r_{iv} \frac{R - \|r_{iv}\|}{\|r_{iv}\|} \tag{8}$$

$\mathbf{T}_i$ is the exploitation vector direction for solution candidate x_i. If in the same direction as vector $\mathbf{T}$, there is a set of repellent odors $r_{iv}(sign(T) = sign(r_{iv}))$, and then the length of the $\mathbf{T}$ vector will be normalized to the length of one of the repellent odors in that direction chosen randomly. R is the Euclidean distance, and a control parameter represents the bee visual range. x'_{iv} is the repellent/attractant vector relative to the origin, r_{iv} is the repellent/attractant vector relative to the current candidate position (x_i), and N_v is the number of the nearest odors around the i-site within the range R. The maximum number N_v is V. V is volatile optimization parameters. v_{ij} is modified solution from x_{ij}, and $j \in \{1, 2, \ldots, D\}$ is index for variable in vector x to be changed.

In the first run, all variables in the vector x will be changed until the number of odors in the search space equals $SN * V$, and then j will be selected probabilistically by the equation described in (9).

$$p_j = \frac{\mathbf{T}_{ij}}{\sum_{k=1}^{D} \mathbf{T}_{ik}} \tag{9}$$

In nature, the odor that bees dissipate will disappear within a certain time and it is adopted into the algorithm where the old odor will be replaced with a new odor when the number of odors in the search space has already exceeded the maximum $SN * V$.

After gaining a new candidate and evaluating its fitness value, the bee spreads out the odor at the new location according to its fitness value. At this stage, greedy selection mechanism will be applied [32, 37, 39]. The site will be moved to the new location (solution) with the same or better fitness value (10).

$$x_i = \begin{cases} v_i & \text{if } f(v_i) \leq f(x_i) \\ x_i & \text{else} \end{cases} \tag{10}$$

The exploitation phase by the employed bee will take as much as SN, and each site will be exploited once $(i = 1, \ldots, SN)$.

After the exploitation by the employed bee takes place for all sites, the bees will recruit unemployed/onlooker bees located in the nest by spreading odor. In this phase, bees will be recruited to a particular site based on the quality of the site. The possibility of a bee in selecting a site is proportional to its fitness value. Many ways and formulas can be used to determine this probability based on the fitness values [75]. In this paper, Equations (11) and (12) will be used to define the probability of the fitness value.

$$fit_i = \frac{1}{\sqrt{r(x_i)}} \tag{11}$$

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \tag{12}$$

where p_i is the possibility that i-site will be exploited by the unemployed bee, and $r(x_i)$ is the rank of i-site sorted by its objective function evaluation [32, 39]. The equation describes that the lower quality of a site than others, then the possibility to be exploited will be smaller. The unemployed bee phase aims to allow the potential candidate solutions shall be exploited more often than the less potential candidates in order to reduce the number of function evaluations.

The unemployed bee phase lasts until all the bees in a colony exploit a site. Once the cycle ends, it will recur from the employed bee phase. Thus, the number of unemployed bees (OL) equals number of colonies (CS) subtracted with the number of employed bees (SN) or in other word $OL = CS - SN$. A site will continue to be exploited by bees until the available source of food is exhausted. In this case, the site will be abandoned if in L visit, and its fitness value cannot be improved any more. The existence of an abandoned site will force the search bee (scout bee) to explore potential new sources.

4.2.3. *Exploration.* The exploration phase is essentially similar to the initialization stage where the solution candidate is randomly generated by Equation (1). In this algorithm, some modifications will be added to adopt stingless bee foraging behaviors. Scout bee can choose either to explore another food source or to visit previously abandon sites for investigation. The probability for the scout bee to choose an old site is defined by the parameter rA. This parameter controls the balance between exploitation and exploration.

If a bee chooses to visit an abandoned site, the site is called a reactivation site. To investigate this site, bees will try to exploit as usual but in the search space R is scaled down. Exploitations are defined in Equations (13) and (14).

$$\mathbf{T}_i = -\sum_{v=1}^{N_v} r_{iv} \frac{\alpha_i R - \|r_{iv}\|}{\|r_{iv}\|} \tag{13}$$

$$\alpha_i = \frac{\sum_{v=1}^{N_v} \|r_{iv}\|}{RN_v} \tag{14}$$

$\mathbf{T}_i$ describe investigations using R that is scaled down by the α multiplier obtained from the surrounding odors just before the site abandoned. In other words, Equations (13) and (14) mention that R for the reactivation site is the average distance of the site to the surrounding repellent odors.

5. **Results and Discussions.**

Benchmark functions. To evaluate SBA performances, in this paper, 50 well-known benchmark functions will be used as in [32]. In this paper, four performance conditions exist to be used with the purpose of comparing the performances of the algorithm.

- *Success rate (SR)* is defined as the percentage of success of the algorithm to get the solution of an optimization problem. An optimization process is said to be successful if it can find a solution that creates a fitness value below or equal to an acceptable minimum fitness level (MFL) [75]. MFL in this experiment is classified as 5% of the mean of the best fitness value at the initialization phase minus the minimum fitness value of each benchmark function.
- *Final error (FE)* is defined as the differences between the minimum value of each benchmark function and the minimum value found at the termination.
- *Average function evaluation to a solution (AES)* represents the average of function evaluation required for the algorithm to find MFL [75].
- *Sum of errors (SE)* is the area below the optimization convergence curve.

Common settings. To make a fair comparison, all experiments in this work will use common parameters: *Colony Size* ($CS = 20$), *Sites Number* ($SN = 10$) and maximum evaluation number 20000 for all functions. A different value less than 10^{-6} is considered the same. 50 benchmark functions will be optimized 30 times for each algorithm.

- *ABC Settings.* Except common parameters, the ABC algorithm in this experiment uses one control parameter: *Limit for abandonment* ($L = 100$).

- *SBA Settings.* Except common parameters, SBA in this experiment uses four control parameters: *Limit for abandonment* ($L = 100$), *Volatile* ($V = 10$), *Radius of sight* ($0.5 * \|ub - lb\|$) and *Reactivation* ($rA = 0$).

The results of numerical experiment are shown in Tables 1-4. The p-value more than 0.05 is considered not significantly different. Regarding the SR criteria (Table 1), ABC

Table 1. SBA and ABC performance in success rate criteria

No	Function	SBA (%)	ABC (%)	Difference (%)	SIGN
1	Schwefel 1.2	73	17	57	SBA
2	Langerman5	17	50	33	ABC
3	Langerman10	13	37	23	ABC
4	Schwefel	37	40	3	ABC
5	Beale	100	97	3	SBA
6	Shekel10	97	97	0	–
7	Stepint	100	100	0	–
8	Step	100	100	0	–
9	Sphere	100	100	0	–
10	SumSquares	100	100	0	–
11	Quartic	100	100	0	–
12	Easom	100	100	0	–
13	Matyas	100	100	0	–
14	Colville	100	100	0	–
15	Trid6	100	100	0	–
16	Trid10	100	100	0	–
17	Zakharov	100	100	0	–
18	Powell	100	100	0	–
19	Schwefel 2.22	100	100	0	–
20	Rosenbrock	100	100	0	–
21	Dixon-Price	100	100	0	–
22	Foxholes	100	100	0	–
23	Branin	100	100	0	–
24	Bohachevsky1	100	100	0	–
25	Booth	100	100	0	–
26	Rastrigin	100	100	0	–
27	Michalewicz2	100	100	0	–
28	Michalewicz5	100	100	0	–
29	Michalewicz10	100	100	0	–
30	Schaffer	100	100	0	–
31	6HumpCamelBack	100	100	0	–
32	Bohachevsky2	100	100	0	–
33	Bohachevsky3	100	100	0	–
34	Shubert	100	100	0	–
35	GoldStein-Price	100	100	0	–
36	Kowalik	100	100	0	–
37	Shekel5	100	100	0	–
38	Shekel7	100	100	0	–
39	Perm	100	100	0	–
40	PowerSum	100	100	0	–
41	Hartman3	100	100	0	–
42	Hartman6	100	100	0	–
43	Griewank	100	100	0	–
44	Ackley	100	100	0	–
45	Penalized	100	100	0	–
46	Penalized2	100	100	0	–
47	Langerman2	100	100	0	–
48	FletcherPowell2	100	100	0	–
49	FletcherPowell5	100	100	0	–
50	FletcherPowell10	100	100	0	–

Table 2. SBA and ABC performance in average function evaluation to a solution criteria

No	Function	SBA		ABC		p-value	SIGN
		AES	SD	AES	SD		
1	Ackley	15875	1561	9099	1006	0.000	ABC
2	Rastrigin	9497	1986	5918	819	0.000	ABC
3	Stepint	47	14	340	161	0.000	SBA
4	Easom	543	229	5538	3467	0.000	SBA
5	Hartman3	87	51	226	133	0.000	SBA
6	Penalized	597	232	951	318	0.000	SBA
7	Penalized2	677	341	1013	250	0.000	SBA
8	SumSquares	2220	382	1851	285	0.000	ABC
9	Sphere	2211	421	1839	292	0.000	ABC
10	Michalewicz10	6641	3327	4131	1474	0.000	ABC
11	Bohachevsky3	46	15	66	25	0.000	SBA
12	Bohachevsky1	42	12	64	31	0.000	SBA
13	Booth	49	20	97	71	0.001	SBA
14	Quartic	1536	410	1179	388	0.001	ABC
15	Schaffer	876	561	501	265	0.001	ABC
16	Beale	55	26	102	76	0.002	SBA
17	Michalewicz2	105	46	153	70	0.002	SBA
18	Branin	65	26	106	67	0.002	SBA
19	6HumpCamelBack	39	14	51	19	0.003	SBA
20	Matyas	54	25	194	261	0.003	SBA
21	Schwefel 1.2	14528	3943	17638	1377	0.004	SBA
22	Bohachevsky2	43	11	60	32	0.004	SBA
23	Kowalik	51	23	76	43	0.004	SBA
24	Schwefel 2.22	30	0	34	8	0.006	SBA
25	Step	2084	461	1816	341	0.007	ABC
26	Shubert	209	94	272	133	0.019	SBA
27	Rosenbrock	975	351	1135	227	0.020	SBA
28	GoldStein-Price	45	20	906	2326	0.026	SBA
29	Colville	57	26	75	51	0.046	SBA
30	Powell	843	220	727	297	0.046	ABC
31	Dixon-Price	918	361	1069	329	0.048	SBA
32	Shekel7	1555	2445	2523	3663	–	–
33	Trid6	299	136	312	177	–	–
34	Trid10	740	354	625	246	–	–
35	Zakharov	30	0	31	4	–	–
36	Foxholes	73	55	96	53	–	–
37	Schwefel	15481	4195	15983	2693	–	–
38	Michalewicz5	1457	990	1136	1121	–	–
39	Shekel5	1363	1981	2190	3255	–	–
40	Shekel10	2010	3520	2816	4052	–	–
41	Perm	68	35	66	60	–	–
42	PowerSum	39	19	45	23	–	–
43	Hartman6	773	1450	499	239	–	–
44	Griewank	2025	403	1903	367	–	–
45	Langerman2	296	578	330	191	–	–
46	Langerman5	4552	1893	6372	3998	–	–
47	Langerman10	10902	5649	12486	5761	–	–
48	FletcherPowell2	59	38	77	45	–	–
49	FletcherPowell5	178	106	203	117	–	–
50	FletcherPowell10	697	331	779	254	–	–

Table 3. SBA and ABC performance in sum of errors criteria

No	Function	SBA		ABC		p-value	SIGN
		SE	SD	SE	SD		
1	Rosenbrock	2.67×10^9	1.06×10^9	6.79×10^9	1.64×10^9	0.000	SBA
2	Penalized	3.32×10^9	1.47×10^9	1.25×10^{10}	4.25×10^9	0.000	SBA
3	Ackley	4.72×10^3	4.28×10^2	3.79×10^3	3.27×10^2	0.000	ABC
4	Penalized2	7.63×10^9	4.12×10^9	2.73×10^{10}	9.56×10^9	0.000	SBA
5	Sphere	1.76×10^6	3.78×10^5	2.57×10^6	4.06×10^5	0.000	SBA
6	Step	1.67×10^6	3.42×10^5	2.53×10^6	4.85×10^5	0.000	SBA
7	Dixon-Price	1.69×10^7	7.93×10^6	4.65×10^7	1.76×10^7	0.000	SBA
8	SumSquares	2.66×10^5	6.38×10^4	3.94×10^5	6.62×10^4	0.000	SBA
9	Stepint	4.13	4.91	7.56×10^1	4.42×10^1	0.000	SBA
10	Griewank	1.55×10^4	3.10×10^3	2.35×10^4	5.15×10^3	0.000	SBA
11	Easom	2.18×10^1	1.11×10^1	2.11×10^2	1.43×10^2	0.000	SBA
12	Rastrigin	4.50×10^4	4.33×10^3	3.85×10^4	3.85×10^3	0.000	ABC
13	Michalewicz10	4.98×10^2	1.19×10^2	3.72×10^2	7.44×10^1	0.000	ABC
14	Hartman3	7.47×10^{-1}	5.38×10^{-1}	4.23	3.59	0.000	SBA
15	Beale	2.50	2.35	1.11×10^1	1.07×10^1	0.000	SBA
16	Matyas	4.40×10^{-1}	4.02×10^{-1}	2.44	2.67	0.000	SBA
17	Schwefel	1.48×10^6	2.20×10^5	1.80×10^6	1.09×10^5	0.000	SBA
18	Bohachevsky1	2.07×10^2	2.19×10^2	1.17×10^3	1.40×10^3	0.000	SBA
19	Booth	9.04	1.36×10^1	3.43×10^1	3.90×10^1	0.001	SBA
20	Kowalik	1.03	4.08×10^{-1}	2.44	2.39	0.002	SBA
21	Bohachevsky2	1.78×10^2	1.71×10^2	6.73×10^2	8.64×10^2	0.002	SBA
22	Schaffer	8.85	3.67	6.35	3.02	0.003	ABC
23	FletcherPowell10	3.75×10^6	1.24×10^6	4.76×10^6	1.52×10^6	0.003	SBA
24	Branin	2.26	2.42	5.02	4.69	0.003	SBA
25	FletcherPowell2	6.93×10^2	7.96×10^2	2.28×10^3	2.87×10^3	0.003	SBA
26	6HumpCamelBack	1.07	9.92×10^{-1}	3.67	4.79	0.003	SBA
27	Quartic	2.98×10^3	7.68×10^2	3.64×10^3	1.15×10^3	0.006	SBA
28	Hartman6	1.24×10^1	7.86	1.73×10^1	6.73	0.006	SBA
29	PowerSum	1.33×10^2	9.88×10^1	6.05×10^2	9.67×10^2	0.006	SBA
30	Michalewicz5	6.66×10^1	3.34×10^1	4.86×10^1	1.79×10^1	0.006	ABC
31	Trid10	1.23×10^5	5.57×10^4	1.60×10^5	5.97×10^4	0.007	SBA
32	Trid6	2.96×10^3	1.35×10^3	4.21×10^3	2.58×10^3	0.012	SBA
33	Bohachevsky3	3.16×10^2	3.21×10^2	1.08×10^3	1.72×10^3	0.012	SBA
34	GoldStein-Price	4.37×10^1	5.89×10^1	1.43×10^3	3.27×10^3	0.014	SBA
35	Colville	7.39×10^3	5.68×10^3	1.70×10^4	2.50×10^4	0.025	SBA
36	Michalewicz2	1.61	1.42	2.39	1.67	0.028	SBA
37	Zakharov	1.88×10^4	8.59×10^3	4.85×10^4	1.33×10^5	–	–
38	Powell	9.07×10^4	2.36×10^4	1.06×10^5	4.51×10^4	–	–
39	Schwefel 2.22	3.82×10^{10}	1.58×10^{11}	1.78×10^{14}	7.00×10^{14}	–	–
40	Schwefel 1.2	1.81×10^7	2.91×10^6	2.08×10^7	3.94×10^6	–	–
41	Foxholes	3.52×10^2	2.27×10^2	4.53×10^2	4.14×10^2	–	–
42	Shubert	7.28×10^2	4.01×10^2	7.04×10^2	4.01×10^2	–	–
43	Shekel5	3.78×10^2	5.98×10^2	4.11×10^2	4.95×10^2	–	–
44	Shekel7	4.42×10^2	7.50×10^2	4.50×10^2	6.94×10^2	–	–
45	Shekel10	6.74×10^2	1.29×10^3	7.66×10^2	1.24×10^3	–	–
46	Perm	5.12×10^3	6.34×10^3	7.07×10^3	1.34×10^4	–	–
47	Langerman2	7.90	8.66	5.86	4.11	–	–
48	Langerman5	1.04×10^2	3.76×10^1	1.21×10^2	5.84×10^1	–	–
49	Langerman10	1.93×10^2	7.76×10^1	1.76×10^2	6.83×10^1	–	–
50	FletcherPowell5	1.75×10^5	1.40×10^5	2.35×10^5	1.39×10^5	–	–

TABLE 4. SBA and ABC performance in final error criteria

No	Function	SBA		ABC		p-value	SIGN
		FE	SD	FE	SD		
1	Rastrigin	1.04×10^{1}	2.17	1.60	1.25	0.000	ABC
2	Quartic	9.23×10^{-1}	1.78×10^{-1}	3.04×10^{-1}	6.64×10^{-2}	0.000	ABC
3	Dixon-Price	2.43	8.64×10^{-1}	2.22×10^{-1}	5.03×10^{-1}	0.000	ABC
4	Rosenbrock	1.00×10^{2}	3.48×10^{1}	9.84	7.74	0.000	ABC
5	Schwefel 2.22	2.06×10^{-2}	8.29×10^{-3}	1.08×10^{-5}	7.01×10^{-6}	0.000	ABC
6	Griewank	1.25×10^{-1}	7.49×10^{-2}	1.09×10^{-2}	1.31×10^{-2}	0.000	ABC
7	Colville	6.75×10^{-2}	1.17×10^{-1}	8.14×10^{-1}	6.00×10^{-1}	0.000	SBA
8	Ackley	2.77×10^{-1}	2.27×10^{-1}	9.92×10^{-5}	7.90×10^{-5}	0.000	ABC
9	Perm	3.71×10^{-2}	9.30×10^{-2}	5.80×10^{-1}	4.66×10^{-1}	0.000	SBA
10	Step	2.50	2.06	1.67×10^{-1}	3.79×10^{-1}	0.000	ABC
11	PowerSum	5.88×10^{-3}	7.39×10^{-3}	4.83×10^{-2}	3.90×10^{-2}	0.000	SBA
12	Sphere	7.21×10^{-3}	7.29×10^{-3}	0	0	0.000	ABC
13	Powell	3.36×10^{-1}	1.61×10^{-1}	1.75×10^{-1}	6.36×10^{-2}	0.000	ABC
14	Zakharov	9.55×10^{-1}	7.59×10^{-1}	5.26	4.58	0.000	SBA
15	Michalewicz10	8.06×10^{-2}	7.01×10^{-2}	1.38×10^{-2}	3.32×10^{-2}	0.000	ABC
16	Penalized2	1.83×10^{-3}	2.06×10^{-3}	0	0	0.000	ABC
17	Trid10	4.96×10^{-1}	8.66×10^{-1}	3.77	4.03	0.000	SBA
18	SumSquares	1.23×10^{-3}	1.79×10^{-3}	0	0	0.000	ABC
19	Trid6	5.80×10^{-7}	1.54×10^{-6}	7.19×10^{-5}	1.09×10^{-4}	0.001	SBA
20	Penalized	1.89×10^{-4}	3.09×10^{-4}	4.68×10^{-8}	2.56×10^{-7}	0.001	ABC
21	Schaffer	2.27×10^{-3}	4.18×10^{-3}	1.58×10^{-5}	5.83×10^{-5}	0.003	ABC
22	Kowalik	3.81×10^{-4}	2.24×10^{-4}	5.19×10^{-4}	1.96×10^{-4}	0.007	SBA
23	Schwefel 1.2	8.51×10^{3}	1.83×10^{3}	1.01×10^{4}	9.20×10^{2}	0.008	SBA
24	Shekel10	0	0	1.30×10^{-2}	3.66×10^{-2}	0.033	SBA
25	Easom	0	0	1.01×10^{-3}	3.14×10^{-3}	0.044	SBA
26	GoldStein-Price	0	0	2.27×10^{-3}	7.08×10^{-3}	0.045	SBA
27	Schwefel	4.33×10^{2}	7.29×10^{1}	4.20×10^{2}	5.95×10^{1}	–	–
28	FletcherPowell10	2.61×10^{2}	3.24×10^{2}	2.02×10^{2}	2.06×10^{2}	–	–
29	FletcherPowell5	3.26	4.47	6.17	1.20×10^{1}	–	–
30	Langerman5	5.03×10^{-3}	1.13×10^{-2}	5.50×10^{-3}	1.06×10^{-2}	–	–
31	Langerman2	1.64×10^{-3}	6.25×10^{-3}	3.70×10^{-5}	1.53×10^{-4}	–	–
32	Michalewicz5	1.39×10^{-3}	7.62×10^{-3}	1.90×10^{-4}	1.04×10^{-3}	–	–
33	Shekel7	0	0	1.33×10^{-2}	5.02×10^{-2}	–	–
34	Shekel5	0	0	1.94×10^{-3}	9.13×10^{-3}	–	–
35	Langerman10	0	0	5.70×10^{-6}	1.28×10^{-5}	–	–
36	Beale	0	0	1.47×10^{-6}	7.45×10^{-6}	–	–
37	Stepint	0	0	0	0	–	–
38	Matyas	0	0	0	0	–	–
39	Foxholes	0	0	0	0	–	–
40	Branin	0	0	0	0	–	–
41	Bohachevsky1	0	0	0	0	–	–
42	Booth	0	0	0	0	–	–
43	Michalewicz2	0	0	0	0	–	–
44	6HumpCamelBack	0	0	0	0	–	–
45	Bohachevsky2	0	0	0	0	–	–
46	Bohachevsky3	0	0	0	0	–	–
47	Shubert	0	0	0	0	–	–
48	Hartman3	0	0	0	0	–	–
49	Hartman6	0	0	0	0	–	–
50	FletcherPowell2	0	0	0	0	–	–

outperforms SBA on 3 functions and SBA outperforms ABC on 2 functions. The performance difference of only 5 out of 50 functions indicates both algorithms generally exhibit the same SR level. ABC is superior to SBA especially on Schwefel, Langerman5 and Langerman10 functions which are highly multi-modal functions. The SBA exploration mechanism using repellent odor might prevent artificial bees exploring the area that is surrounded by the local maximum. However, this mechanism seems to be effective in dealing with a shallow area around the minimum point as in Schwefel 1.2 and Beale functions. In these functions, ABC hardly finds the minimum point because basic ABC algorithm does not use local information in the exploration phase.

On the AES criteria (Table 2), SBA outperforms ABC on 22 functions while ABC outperforms on 9 functions. On the SE criteria (Table 3), SBA outperforms ABC on 31 functions while ABC outperforms on only 5 functions. From these two criteria (AES and SE), SBA succeeded in obtaining solutions faster than ABC. More eminence of SBA in SE criteria indicates that if the iteration time is shortened or the solution criterion is higher, then SBA superiority over ABC will increase. Figure 3 shows the comparisons of the mean best values evolution between the SBA and ABC algorithms. In FE criteria (Table 4), ABC outperforms SBA on 15 functions while SBA outperforms on 11 functions. ABC outperforms SBA on Rastrigin, Quartic, Dixon-Price and Rosenbrock functions. It showed the best SBA performance with FE = 0 in Shekel10, Easom, and GoldStein-Price functions, while ABC outperforms SBA on Sphere, Penalized and SumSquares functions. These functions are uni-modal functions with sloping to minimum solutions that require high precision. In these functions, ABC's exploitation strategy seems more effective than SBA to achieve minimum error.

Shekel10 function, Easom, and GoldStein-Price are slightly multi-modal functions, ramps, and small minimum points. In these functions, exploitation mechanism of ABC does not work well because chances at some sites are stuck at a local minimum and hold up the algorithm to find the solution faster. SBA seems better in these functions because it could use odor information around the site without being affected by other sites that

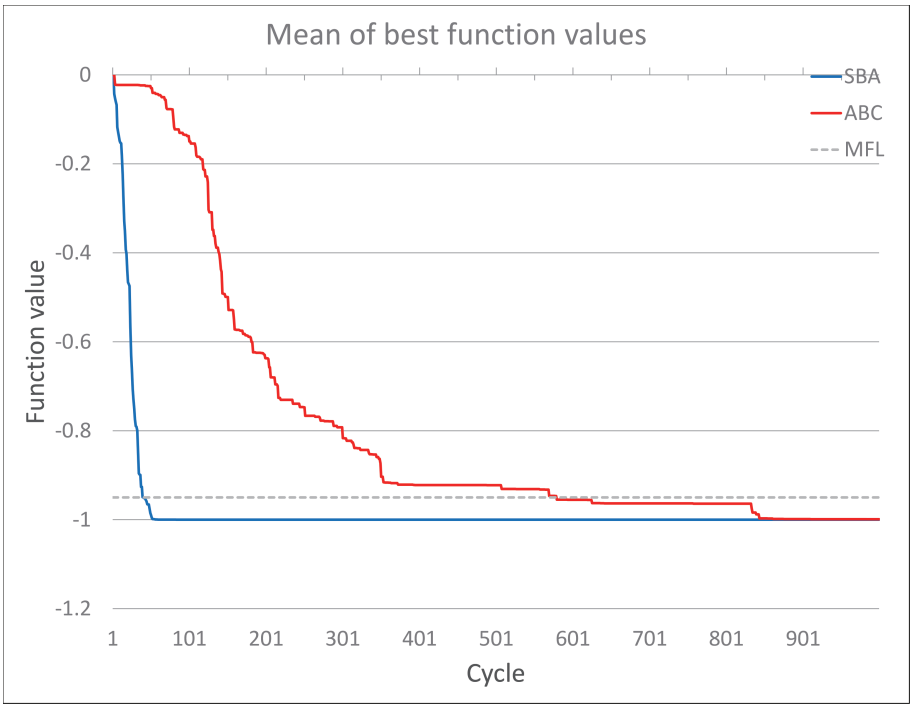

FIGURE 3. Evolution of mean best values of Easom function

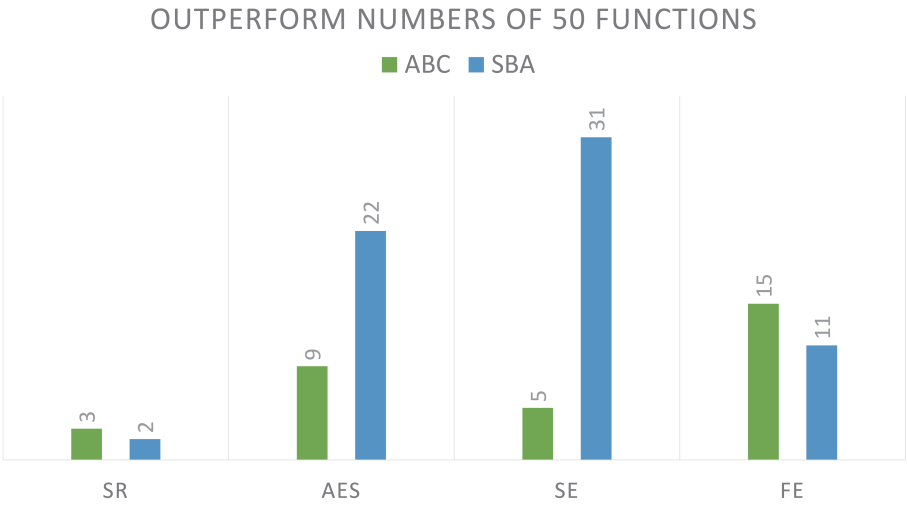

FIGURE 4. Comparison of outperform performance numbers of ABC and SBA

stuck to a local minimum. Figure 4 shows the outperform performance numbers of both ABC and SBA on 50 benchmark functions. In SR and FE performance criteria, the ABC demonstrated better performance than the SBA with small differences. However, in AES and SE performance criteria, the SBA algorithm resulted in very high outperform numbers compared to the ABC. The conclusion exists that in general the performance of the SBA is, according to the four performance criteria, similar to or better than the ABC.

6. **Conclusions and Future Work.** The paper considered the development of stingless bee algorithm (SBA) which naturally exhibits the reactivation parameter for balancing exploitation and exploration practical for the search algorithm. The performances of the SBA and ABC in solving numerical problems were found comparable. The SBA resulted in better performances than the ABC on average evaluation to a solution (AES) and sum of errors (SE) criteria on 22 and 31 benchmark functions while the ABC surpassed the SBA at 9 and 5 functions respectively, showing the SBA performance to obtain solution faster. On the success rate (SR) and the final error (FE) criteria, the ABC outperforms the SBA on 3 and 15 benchmark functions while the SBA improved on 2 and 11 functions. The SBA was able to result in better convergence of the computation speed; furthermore the effectiveness of the SBA was almost similar to the ABC. Based on four performance criteria, the SBA leaded to similar or better performances compared to ABC algorithm.

To summarize, the nature of the two groups of bees, the honey bee (Apini) and stingless bee (Meliponini) strive to survive highlighting the way of the SBA and ABC algorithms operated and performed on numerical optimization problems. The two bee colonies survived through years of their foraging behaviors evolution. However, stingless bees live in small size population that makes them develop efficient and effective way to locate and find foods. The proposed SBA complemented the existing artificial bee colony algorithm and offered several alternative advantages on solving numerical optimization problems. Future research works are to improve the SBA by investigating and incorporating other foraging behaviors of stingless bees that were reported in many literatures. On the one hand, unifying the foraging behaviors of honey bees and stingless bees into one powerful algorithm is possible. Conversely, following the nature, let each bee algorithm grows by fully exploiting its foraging behaviors following the living way of bees.

Acknowledgment. This research was initially supported in part by the Ministry of Research, Technology and Higher Education, the Republic of Indonesia under the Decentralized Research Program on Excellent Research University, Institut Teknologi Bandung, Indonesia 2017 and was partially funded by the Research Program, Institut Teknologi Bandung 2019-2020.

REFERENCES

[1] P. C. Gilmore and R. E. Gomory, A linear programming approach to the cutting-stock problem, *Operations Research*, vol.9, no.6, pp.849-859, 1961.

[2] S. Krishnamurthy and R. Tzoneva, Comparison of the Lagrange's and particle swarm optimisation solutions of an economic emission dispatch problem with transmission constraints, *2012 IEEE International Conference on Power Electronics, Drives and Energy Systems (PEDES)*, pp.1-8, 2012.

[3] H. A. Q. Tran, A. Namatame, A. Widyotriatmo and E. Joelianto, An optimization procedure for enhancing network robustness against cascading failures, *The 7th IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, pp.1-7, 2014.

[4] D. Bajer and B. Zorié, An effective refined artificial bee colony algorithm for numerical optimisation, *Information Sciences*, vol.504, pp.221-275, 2019.

[5] L. Cui, G. Li, X. Wang, Q. Lin, J. Chen, N. Lu and J. Lu, A ranking-based adaptive artificial bee colony algorithm for global numerical optimization, *Information Sciences*, vol.417, pp.169-185, 2017.

[6] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer-Verlag, New York, 1999.

[7] X. S. Yang, Metaheuristic optimization, *Scholarpedia*, 2011.

[8] A. F. Vereshchagin, Modeling and control of motion of manipulational robots, *Soviet Journal of Computer and Systems Sciences*, vol.27, no.5, pp.29-38, 1989.

[9] G. Kaur, S. S. Gill and M. Rattan, Whale optimization algorithm for performance improvement of silicon-on-insulator finfet, *International Journal of Artificial Intelligence*, vol.18, no.1, pp.63-81, 2020.

[10] B. P. De, R. Kar, D. Mandal and S. P. Ghoshal, Optimal selection of components value for analog active filter design using simplex particle swarm optimization, *International Journal of Machine Learning and Cybernetics*, vol.6, no.4, pp.621-636, 2015.

[11] J. C. Cervantes-González, J. E. Rayas-Sánchez, C. A. López, J. R. Camacho-Pérez, Z. Brito-Brito and J. L. Chávez-Hurtado, Space mapping optimization of handset antennas considering EM effects of mobile phone components and human body, *International Journal of RF and Microwave Computer-Aided Engineering*, vol.26, no.2, pp.121-128, 2016.

[12] S. Koziel and J. W. Bandler, Space mapping with multiple coarse models for optimization of microwave components, *IEEE Microwave and Wireless Components Letters*, vol.18, no.1, pp.1-3, 2008.

[13] A. S. Majid and E. Joelianto, Optimal sensor deployment in non-convex region using discrete particle swarm optimization algorithm, *2012 IEEE Conference on Control, Systems and Industrial Informatics*, pp.109-113, 2012.

[14] B. Qiu, Resource allocation optimization strategy using improved differential evolution algorithm in multi-radio multi-channel WSNS, *International Journal of Innovative Computing, Information and Control*, vol.16, no.3, pp.923-937, 2020.

[15] R. E. Kalman, Contributions to the theory of optimal control, *Bol. Soc. Mat. Mexicana*, vol.5, no.2, pp.102-119, 1960.

[16] S. Salmah, S. Sutrisno, E. Joelianto, A. Budiyono, I. E. Wijayanti and N. Y. Megawati, Model predictive control for obstacle avoidance as hybrid systems of small scale helicopter, *2013 the 3rd International Conference on Instrumentation Control and Automation (ICA)*, pp.127-132, 2013.

[17] E. Joelianto and I. Wiranto, An application of ant colony optimization, Kalman filter and artificial neural network for multiple target tracking problems, *International Journal of Artificial Intelligence*, vol.7, no.A11, pp.384-400, 2011.

[18] E. T. Papoutsakis, Equations and calculations for fermentations of butyric acid bacteria, *Biotechnology and Bioengineering*, vol.67, no.6, pp.813-826, 2000.

[19] R. Dorfman, An economic interpretation of optimal control theory, *The American Economic Review*, vol.59, no.5, pp.817-831, 1969.

[20] T. J. Sargent, *Dynamic Macroeconomic Theory*, Harvard University Press, 2009.

[21] J. J. Rotemberg and M. Woodford, An optimization-based econometric framework for the evaluation of monetary policy, *NBER Macroeconomics Annual*, vol.12, pp.297-346, 1997.

[22] A. Ahmadi-Javid, Z. Jalali and K. J. Klassen, Outpatient appointment systems in healthcare: A review of optimization studies, *European Journal of Operational Research*, vol.258, no.1, pp.3-4, 2017.

[23] Z. Zhai, J. E. Martínez Ortega, N. Lucas Martínez and J. Rodríguez-Molina, A mission planning approach for precision farming systems based on multi-objective optimization, *Sensors*, vol.18, no.6, 2018.

[24] R. P. A. Gil, Z. C. Johanyák and T. Kovács, Surrogate model based optimization of traffic lights cycles and green period ratios using microscopic simulation and fuzzy rule interpolation, *International Journal of Artificial Intelligence*, vol.16, no.1, pp.20-40, 2018.

[25] C. Li, H. Zhang, H. Zhang and Y. Liu, Short-term traffic flow prediction algorithm by support vector regression based on artificial bee colony optimization, *ICIC Express Letters*, vol.13, no.6, pp.475-482, 2019.

[26] L. Zang, Dynamic programming algorithm and bat algorithm based storm nodes scheduling in edge computing, *International Journal of Innovative Computing, Information and Control*, vol.16, no.3, pp.1021-1033, 2020.

[27] S. El-Kenawy and M. Eid, Hybrid gray wolf and particle swarm optimization for feature selection, *International Journal of Innovative Computing, Information and Control*, vol.16, no.3, pp.831-844, 2020.

[28] E. Bonabeau, M. Dorigo and G. Thereulaz, *Swarm Intelligence: From Natural to Artificial System*, Oxford University Press, New York, 1999.

[29] A. Chakraborty and A. Kar, Swarm intelligence: A review of algorithms, *Nature-Inspired Computing and Optimization*, Springer, Cham, pp.475-494, 2017.

[30] D. Karaboga, *An Idea Based on Honey Bee Swarm for Numerical Optimization*, Technical Report-TR06, Erciyes University, Kayseri, Turkey, 2005.

[31] V. Tereshko and A. Loengarov, Collective decision-making in honey bee foraging dynamics, *Computing and Information Systems*, vol.9, no.3, 2005.

[32] D. Karaboga and B. Akay, A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, vol.214, pp.108-132, 2009.

[33] D. Kong, T. Chang, W. Dai, Q. Wang and H. Sun, An improved artificial bee colony algorithm based on elite group guidance and combined breadth-depth search strategy, *Information Sciences*, vol.442-443, pp.54-71, 2018.

[34] P. Moradi, N. Imanian, N. N. Qader and M. Jalili, Improving exploration property of velocity-based artificial bee colony algorithm using chaotic systems, *Information Sciences*, vol.465, pp.130-143, 2018.

[35] Q. Wei, Z. Guo, H. C. Lau and Z. He, An artificial bee colony-based hybrid approach for waste collection problem with midway disposal pattern, *Applied Soft Computing*, vol.76, pp.629-637, 2019.

[36] V. Pandiri and A. Singhtitle, An artificial bee colony algorithm with variable degree of perturbation for the generalized covering traveling salesman problem, *Applied Soft Computing*, vol.78, pp.481-495, 2019.

[37] H. Wang, W. Wang, S. Xiao, Z. Cui, M. Xu and X. Zhou, Improving artificial bee colony algorithm using a new neighborhood selection mechanism, *Information Sciences*, vol.527, pp.227-240, 2020.

[38] S. Aslan and D. Karaboga, A genetic artificial bee colony algorithm for signal reconstruction based big data optimization, *Applied Soft Computing*, 2020.

[39] D. Karaboga and B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, pp.789-798, 2007.

[40] D. Karaboga and B. Akay, A modified artificial bee colony (ABC) algorithm for constrained optimization problems, *Applied Soft Computing*, vol.11, pp.3021-3031, 2011.

[41] D. Karaboga and B. Gorkemli, A combinatorial artificial bee colony algorithm for traveling salesman problem, *IEEE*, vol.978, pp.50-53, 2011.

[42] D. Karaboga and S. Aslan, A discrete artificial bee colony algorithm for detecting transcription factor binding sites in DNA sequences, *Genetics and Molecular Research*, vol.15, no.2, pp.1-11, 2016.

[43] D. Karaboga and B. Gorkemli, A quick artificial bee colony (qABC) algorithm and its performance on optimization problems, *Applied Soft Computing*, vol.23, pp.227-238, 2014.

[44] F. Harfouchi, H. Habbi, C. Ozturk and D. Karaboga, Modified multiple search cooperative foraging strategy for improved artificial bee colony optimization with robustness analysis, *Soft Computing*, vol.22, no.19, pp.6371-6394, 2018.

[45] H. Tsai, Artificial bee colony directive for continuous optimization, *Applied Soft Computing*, vol.87, 2020.
[46] D. Kumar and K. K. Mishra, Co-variance guided artificial bee colony, *Applied Soft Computing*, vol.70, pp.86-107, 2018.
[47] X. Wang, Y. Zhang, X. Sun, Y. Wang and C. Du, Multi-objective feature selection based on artificial bee colony: An acceleration approach with variable sample size, *Applied Soft Computing*, vol.88, 2020.
[48] H. Badem, A. Basturk, A. Caliskan and M. E. Yuksel, A new hybrid optimization method combining artificial bee colony and limited-memory BFGS algorithms for efficient numerical optimization, *Applied Soft Computing*, vol.70, pp.826-844, 2018.
[49] C. D. Michener, *The Bees of the World*, John Hopkins University Press, Baltimore and London, 2000.
[50] J. C. Biesmeijer and E Tóth, Individual foraging, activity level and longetivity in the singles bee Melipona beecheii in Costa Rica (Hymenoptera, Apidae, Meliponini), *Insectes Sociaux*, vol.45, pp.427-443, 1998.
[51] S. O'Donnell and R. L. Jeanne, Lifelong patterns of forager behaviour in a tropical swarm-founding wasp: Effects of specialization and activity level on longevity, *Animal Behaviour*, vol.44, no.6, pp.1021-1027, 1992.
[52] J. H. Fewell and M. L. Winston, Colony state and regulation of pollen foraging in the honey bee, *Apis mellifera* L., *Behavioral Ecology and Sociobiology*, vol.30, no.6, pp.387-393, 1992.
[53] M. Lindauer, Recent advances in bee communication and orientation, *Annual Review of Entomology*, vol.12, no.1, pp.439-470, 1967.
[54] C. D. Michener, *The Social Behavior of the Bees: A Comparative Study*, Harvard University Press, 1974.
[55] V. M. Schmidt, R. Zucchi and F. G. Barth, A stingless bee marks the feeding site in addition to the scent path (Scaptotrigona aff. depilis), *Apidologie*, vol.34, no.3, pp.237-248, 2003.
[56] S. Jarau, M. Hrncir, M. Ayasse, C. Schulz, W. Francke, R. Zucchi and F. G. Barth, A stingless bee (Melipona seminigra) marks food sources with a pheromone from its claw retractor tendons, *Journal of Chemical Ecology*, vol.30, no.4, pp.793-804, 2004.
[57] V. M. Schmidt, D. L. P. Schorkopf, M. Hrncir, R. Zucchi and F. G. Barth, Collective foraging in a stingless bee: Dependence on food profitability and sequence of discovery, *Animal Behaviour*, vol.72, no.6, pp.1309-1317, 2006.
[58] M. Hrncir and C. Maia-Silva, On the diversity of foraging-related traits in stingless bees, *Pot-Honey*, pp.201-215, 2013.
[59] M. Hrncir, S. Jarau and F. G. Barth, Stingless bees (Meliponini): Senses and behavior, *Journal of Comparative Physiology A*, vol.202, pp.9-10, 2016.
[60] M. Hrncir, V. M. Schmidt, D. L. P. Schorkopf, S. Jarau, R. Zucchi and F. G. Barth, Vibrating the food receivers: A direct way of signal transmission in stingless bees (Melipona seminigra), *Journal of Comparative Physiology A*, vol.192, no.8, pp.879-887, 2006.
[61] K. Krausa, F. A. Hager and W. H. Kirchner, The effect of food profitability on foraging behaviors and vibrational signals in the African stingless bee Plebeina hildebrandti, *Insectes Sociaux*, vol.64, no.4, pp.567-578, 2017.
[62] T. Inoue, S. Salmah, I. Abbas and E. Yusuf, Foraging behavior of individual workers and foraging dynamics of colonies of three Sumatran stingless bees, *Res. Population Ecology*, vol.27, pp.373-392, 1985.
[63] E. Joelianto and B. Prakoso, Stingless bee foraging behavior algorithm for optimisation, *International Journal of Artificial Intelligence*, vol.15, no.1, pp.1-20, 2017.
[64] B. Prakoso, *Energy Based Ad-Hoc Wireless Network Routing Optimisation Using Stingless Bee Optimisation*, Master Thesis, Bandung Institute of Technology, 2013 (in Indonesian).
[65] R. Ciar, L. S. Bonto, M. H. P. Bayer, J. F. Rabajante, S. P. Lubag, A. C. Fajardo and C. R. Cervancia, Foraging behavior of stingless bees (tetragonula biroi friese): Distance, direction and height of preferred food source, *arXiv Preprint*, arXiv:1310.3919, 2013.
[66] K. Von Frisch, *The Dance Language and Orientation*, Harvard University Press, Cambridge, MA, 1967.
[67] D. W. Roubik, Foraging behavior of competing africanized honeybees and stingless bees, *Ecology*, vol.61, no.4, pp.836-845, 1980.
[68] J. C. Nieh, Recruitment communication in stingless bees (Hymenoptera, Apidae, Meliponini), *Apidologie*, vol.35, pp.159-182, 2004.

[69] S. Jarau, *Chemical Communication during Food Exploitation in Stingless Bees*, Taylor and Francis Group, Florida, USA, 2009.

[70] E. J. Slaa, J. Wassemberg and J. C. Biesmeijer, The use of field-based social information in eusocial foragers: Local enhancement among nestmates and heterospecifics in stingless bees, *Ecological Entomology*, vol.28, pp.369-379, 2003.

[71] D. Sanchez, J. C. Nieh and R. Vandame, Experience-based interpretation of visual and chemical information at food sources in the stingless bee Scaptotrigona mexicana, *Animal Behavior*, vol.76, pp.407-414, 2008.

[72] D. Goulson, J. S. Chamman and W. O. H. Hughes, Discrimination of unrewarding flowers by bees; direct detection of rewards and use of repellent scent marks, *Journal of Insect Behavior*, vol.14, pp.669-678, 2001.

[73] L. Cui, G. Li, Z. Zhu, Q. Lin, Z. Wen, N. Lu, K. C. Wong and J. Chen, A novel artificial bee colony algorithm with an adaptive population size for numerical function optimization, *Information Sciences*, vol.414, pp.53-67, 2017.

[74] W. F. Gao, S. Y. Liu and L. L. Huang, A novel artificial bee colony algorithm based on modified search equation and orthogonal learning, *IEEE Trans. Cybernetics*, vol.43, no.3, pp.1011-1024, 2013.

[75] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd Edition, Springer, Berlin, 2015.