An Artificial Beehive Algorithm for Continuous Optimization

Mario A. Muñoz,[1,*] Jesús A. López,[2,†] Eduardo Caicedo[1,‡]
[1] *Grupo de Investigación Percepción y Sistemas Inteligentes, Universidad del Valle, Cali, Colombia*
[2] *Departamento de Automática y Electrónica, Universidad Autonoma de Occidente, Cali, Colombia*

This paper presents an artificial beehive algorithm for optimization in continuous search spaces based on a model aimed at individual bee behavior. The algorithm defines a set of behavioral rules for each agent to determine what kind of actions must be carried out. Also, the algorithm proposed includes some adaptations not considered in the biological model to increase the performance in the search for better solutions. To compare the performance of the algorithm with other swarm-based Techniques, we conducted statistical analyses by using the so-called t test. This comparison is done with several common benchmark functions.

1. INTRODUCTION

A continuous optimization algorithm is a numerical method to find a value $\theta_i \in R^n$, where R^n is an n-dimensional search space such that it minimizes or maximizes a function $J(\theta)$. This is achieved by systematically choosing values for the variable θ possibly with some constraints. The variable θ can be a scalar or vector of continuous or discrete values. While θ is called a feasible solution, $J(\theta)$ is called an objective function. A feasible solution that minimizes or maximizes the objective function is called an optimal solution.

Generally, when the feasible region or the objective function of the problem does not present convexity, there may be several local minima and maxima and only one global minimum or maximum of the objective function. Several algorithms that have been proposed to solve nonconvex problems, including the majority of commercially available solvers, cannot distinguish between local optimal solutions and global optimal solutions. Global optimization is the branch of applied mathematics and numerical analysis concerned with the development of deterministic algorithms

*Author to whom all correspondence should be addressed: e-mail: andremun@univalle.edu.co.
†e-mail: jalopez@uao.edu.co.
‡e-mail: ecaicedo@univalle.edu.co.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 24, 1080–1093 (2009)
 Published online in Wiley InterScience (www.interscience.wiley.com). • DOI 10.1002/int.20376

capable of guaranteeing convergence in finite time to the actual optimal solution of a nonconvex problem.[1]

Given the difficulty in obtaining optimal solutions with some methods or the failure to guarantee convergence, researchers seek to find new methods to solve optimization problems. One field addressing several applications is the social modeling of simple-being communities, which present, as a whole, complex intelligence far superior to that exhibited by each individual. This has allowed the development of a branch in computational intelligence called swarm intelligence (SI), being applied for the solution of several optimization problems such as vehicle routing, control system design, network routing, image processing, and other fields.[2–8]

This paper presents an algorithm based on the honeybee foraging behavior. The algorithm was developed using the individual oriented (IO) model presented by de Vries and Biesmeijer[9] as the behavioral rule. Also, we tested the performance of the algorithm against a group of SI algorithms for continuous optimization in a statistical study based on several common benchmark functions. The paper is organized as follows: Section 2 presents a brief background in SI and bee algorithms. Section 3 describes the functional characteristics of the artificial beehive algorithm (ABHA). Section 4 presents an experiment to evaluate the performance of the algorithm proposed. Finally, Section 5 presents conclusions to this work.

2. SWARM INTELLIGENCE AND BEE ALGORITHMS

To solve complex optimization problems, researchers have used constructive, local search, and population-based methods.[10] Population-based methods are now very popular since they provide good solutions by using constructive methods to generate an initial set of solutions and a local search method to improve them. Also, said methods can combine good solutions into new ones, thus enhancing them, because it is thought that good solutions share components with optimal solutions.

These methods, also known as evolutionary computation (EC) algorithms, are iterative techniques that, through a set of clearly specified rules, modify their group of solutions alternating between self-adaptation stages, implying changes in the individual, and cooperation stages, in which there is information exchange between individuals. EC is now a broad research area in which two tendencies can be recognized: evolutionary algorithms (EAs), based on the biological evolution and survival-of-the-fittest concepts,[10] and SI.

SI techniques are based on the study of collective behavior in decentralized, self-organized systems. Beni and Wang introduced the expression "swarm intelligence" in 1989, in the context of cellular robotic systems.[11] SI systems are typically made up of a population of simple agents interacting locally among each other and with their environment. Although there is normally no centralized control structure dictating how individual agents should behave, local interactions among such agents often lead to the emergence of global behavior. Examples of systems such as these can be found in nature, including ant colonies, bird flocks, animal herds, bacteria molds, and fish schools. The main characteristics of a swarm are as follows[12]:

- It is distributed and no central control or data source exist.
- It has no explicit model of the environment.
- The agents can perceive and produce changes in their environment.

Different natural systems have inspired several approaches in SI, some of which include the following: particle swarm optimization (PSO),[13–15] based on the movement of bird flocks and fish schools; bacteria swarm foraging optimization (BSFO),[16–18] which models the chemotactic behavior of *Escherichia coli*; and ant colony optimization,[19–21] which is inspired on the foraging behavior of ants. These methods are widely used in the solution of continuous and combinatorial optimization problems.[22–26]

Honeybees have also attracted the attention of SI researchers because they follow a series of rules that determine, among several food sources, the most profitable and allocate a series of agents to exploit it. Besides, they use direct communication methods based on dances. There are several algorithms based on bee behavior such as the artificial bee colony,[27] the bee colony optimization,[28] the *bees* algorithm,[29] and the bee nectar search optimization.[30] Some applications of these algorithms are the traveling salesman problem,[31] neural network training,[32] fuzzy controller tuning,[33] support vector machine optimization,[34] manufacturing cell formation,[35] multiobjective optimization,[36] and temperature regulation over a surface.[37]

3. ARTIFICIAL BEEHIVE ALGORITHM

The beehive algorithm proposed is based on the IO model presented by de Vries and Biesmeijer.[9] In this model, each bee is represented as an individual whose behavior is regulated by a behavior-control structure. At each moment, the behavior of one bee is determined by the internal and external information available to it and its motivational state, according to a set of specific rules. The set of rules is identical for each bee, but since the perceptible environment differs for bees with a different spatial location, the behavior also differs. Bees can show different behaviors as well, given differences in their foraging experience and/or their motivational state.

The algorithm proposed defines a set of individual characteristics that represent internal and external information:

- The current position of the individual, $\theta(t)$, which represents its solution point.
- The current cost value, $J(\theta(t))$, and the past cost value, $J(\theta(t-1))$.
- The abandon tendency, p_{ab}, between 0 and 1, which represents individual's desire to forget its food source information. It is initialized in 0 and incremented in fixed steps defined by the value of ρ.
- The homing motivation, p_{h}, between 0 and 1, which represents individual's desire to continue searching for a food source. It is initialized in 0 and incremented in fixed steps defined by the value of C.
- The bee state that represents the actions the bee will follow next. We defined four states. The first is the *novice* state, where the bee is in the "nest" (an abstract position represented only by the state of the bee, where the information is exchanged) and does not have information about a source. In this state, the bee can begin a random search or follow a dance if it is available. To represent this state, θ value is set at Not a Number (*NaN*). The

second is the *experimented* state, where the bee is in the "nest" and has information about a food source. Information about the source is valid if its profit is high and its abandon tendency is low. Valid information can be transmitted to other individuals through a dance represented by a selection probability p_{si} calculated by Equation 1, where i indicates one of the j individuals with available dances. The agent uses a lottery based selection to define which dance should follow. If the bee does not have valid information, it can begin a random search or follow a dance.

$$p_{si} = -\left(\frac{1}{\max(J_j) - \min(J_j)}\right)(J_i - \max(J_i)) \tag{1}$$

The third state is the *search* state, where the bee, after leaving the nest, looks for a better foraging source than the current. The bee changes its position using Equation 2, where $SS(i)$ is the step size at the direction $\psi(t)$, and it is initiated as 1.0% of the search space dimensions and decreased by the step-reduction parameter SR. The direction is maintained while it minimizes or maximizes the cost value, otherwise it is changed. This search is performed until the step counter is exhausted.

$$\theta_i(t+1) = \theta_i(t) + SS(i)\psi(t) \tag{2}$$

The fourth is the *food source* state. After exhausting the search counter, the bee determines whether its source is valid or not by storing the cost value of the source. If the algorithm is performing a minimization and this value is less than a threshold, the source has enough food to be informed; otherwise, this information is not valuable. If the algorithm is performing maximization, the source value must be higher than the threshold. The threshold value, equal to the mean cost of all the available dances, is recalculated at each iteration. This is analogous to the exhaustion of the food source.

- The set of probabilities, a group of random values, used to calculate some choices of the population. These are the probability to begin a random search, p_{srs}, the probability to listen to a dance, p_{rul}, and the probability to receive contaminated information form a dance, p_{e}. When information is contaminated, the value of an informed source is modified by a random vector similar to that presented in Equation 2, replacing SS by σ, which represents the maximum information error. These probabilities are compared with fixed values that are parameters of the algorithm and are p_{rs} for beginning a random search and p_{e} for including information error. These parameters control some of the exploitation/exploration characteristics of the algorithm.

With these characteristics, we can now present the full algorithm. The pseudo code is presented in Table I.

4. EXPERIMENTS AND RESULTS

We conducted a simulation study by using a suite of 10 benchmark functions and 5 SI algorithms for continuous optimization to test the algorithm proposed: PSO,[14] the inertia weight particle swarm optimization,[38] the PSO with Trelea's first parameter set, and the PSO with Trelea's second parameter set,[39] the constriction factor particle swarm optimization,[40] and BSFO.[16–18] The parameters used for each algorithm are shown in Table II, where d is the diagonal of the search space. The test was carried out for 100 runs of 1000 epochs. The number of agents for all algorithms in each of the runs was constant and equal to 50 agents. All the benchmark

Table I. Artificial beehive algorithm.

```
Initiate the population
FOR n = 1 UNTIL epochs
  Calculate the probabilities
    IF Experimented AND Information is valid
      Increase Abandon
      Leave the nest
    END
    IF Dances are not available
      IF begin a random search
        Place the bees in a random position
        Leave the nest
      END
      IF Abandon the source
        Forget the source
        Stay in the nest
      END
    ELSE
      Calculate dance duration
      IF follows a dance
        Select a random dance
        Add noise to the information
        Leave the nest
      ELSE
        Increase abandon
      END
      IF begin a random search
        Place the bees in a random position
        Leave the nest
      END
    END
    Calculate the current cost
    IF Can search
      IF direction is valid
        Increase the position
      ELSE
        Change the direction
        Increase the position
      END
    ELSE
      Determine if the source is valid
    END
    Update the threshold
    Update the memory
END
```

Table II. Algorithm's parameters for the simulation study.

No.	Name	Parameters
1	BSFO	$p_{\mathrm{ed}} = 0.10$, $p_{\mathrm{re}} = 0.20$, $C_i = 0.05d$, $N_s = 10$
2	PSO	$mv = 0.01d$, $c_1 = c_2 = 2.000$
3	IWPSO	$mv = 0.01d$, $w = 0.600$, $c_1 = c_2 = 2.000$
4	T1PSO	$mv = 0.01d$, $w = 0.600$, $c_1 = c_2 = 1.700$
5	T2PSO	$mv = 0.01d$, $w = 0.729$, $c_1 = c_2 = 1.494$
6	CFPSO	$mv = 0.01d$, $c = 0.7298$, $c_1 = 2.8000$, $c_2 = 1.3000$
7	ABHA	$p_{\mathrm{rs}} = 0.5$, $p_{\mathrm{e}} = 0.8$, $s = 0.2$, $r = C = 0.1$, $SR = 0.95$

Abbreviations: BSFO, bacteria swarm foraging optimization; PSO, particle swarm optimization; IWPSO, inertia weight particle swarm optimization; T1PSO, PSO with Trelea's first parameter set; T2PSO, PSO with Trelea's second parameter set; CFPSO, constriction factor particle swarm optimization; ABHA, artificial beehive algorithm.

functions used are nonlinear, and they are commonly used in optimization algorithm benchmarks. These are the Six-Hump Camel Back (SCB), PFUNC,[18] MATLAB Peaks, DeJong's F1, Grienwank, Rastrigin, Ackley, DeJong's F2, Schaffer's F6, and Schewefel functions.[41] The number of dimensions, range, and minimum value are shown in Table III.

We selected these functions because they are related to real-world problems, where there are unimodal and multimodal functions with correlated or uncorrelated variables. The DeJong F1 function contains no local optima and provides a smooth

Table III. Benchmark functions parameters.

No.	Name	Dimension	Range	$J_{\min}$
1	SCB	2	$[-2, 2; -1, 1]$	−1.0316
2	PFUNC	2	$[0, 30]^2$	−3.9867
3	Peaks	2	$[-3, 3]^2$	−6.5511
4	DeJong's F1	30	$[-100, 100]^n$	0.0000
5	Grienwank	30	$[-100, 600]^n$	0.0000
6	Rastrigin	30	$[-5.12, 5.12]^n$	0.0000
7	Ackley	30	$[-32, 32]^n$	0.0000
8	DeJong's F2	2	$[-100, 100]^2$	0.0000
9	Schaffer's F6	2	$[-100, 100]^2$	0.0000
10	Schewefel	30	$[-500, 500]^n$	−12569.5

gradient toward a broad global optimum. The Grienwank function introduces interdependency among the variables. The Rastrigin function has lattice-shaped semioptimum solutions around the global optima, and there is no correlation among design variables. The Ackley function is also multimodal at low resolution. The search space defined by the De Jong F2 function is unimodal and has correlation among its design variables. The Schewefel function has a semioptimum solution far from the global optima where many search algorithms are trapped. Moreover, the global optimum exists near the bounds of the domain. There is no correlation among its design variables.[41]

We compared the different algorithms by conducting a statistical study using the minimal and mean objective values, as well as the standard deviation. For visualization, we used a box and whisker plot for each of the algorithms in each test to better view the results obtained. This plot allows the comparison between several distributions. Its components include the following: the box, which encloses 50% of the data and it is made up of lines at the lower, median, and upper quartile values; the whiskers, which are the lines extending from each end of the box that show the extent of 99.3% of the data; and the outliers, represented by crosses, which show data with values beyond the ends of the whiskers and represent 0.07% of the data. If there are no data outside the whisker, a dot is placed at the bottom whisker.

Figures 1 to 3 show each the results for the SCB, PFUN, and Peaks functions. The results obtained for these functions are very similar and show that all of the algorithms achieve the global minima with a very little spreading. For the SCB function, the ABHA achieves a result close to the PSO variations and a better one than that obtained by the BSFO algorithm. For the PFUNC function, the existence of outliers for the PSO algorithms, which give them a higher spreading, can be noticed. In the case of the Peaks function, only the PSO algorithm has outliers.

Figure 4 shows the result for the DeJong F1 function. Although this function is very simple, the difficulty lies in its large search space and a high dimension number. The experiment shows that there is a large difference between the results obtained with the ABHA and the PSO algorithm, but the ABHA performs better than the BSFO algorithm. Neither the BSFO algorithm nor the ABHA achieves the global minima during 1000 epochs, which indicates that both need more time to converge.

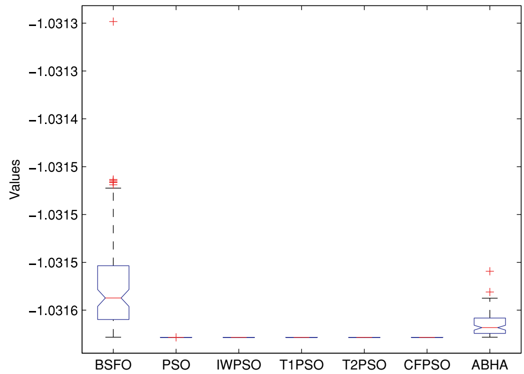

Figure 1. Box plots of the results obtained with the SCB function. Abbreviations: BSFO, bacteria swarm foraging optimization; PSO, particle swarm optimization; IWPSO, inertia weight particle swarm optimization; T1PSO, PSO with Trelea's first parameter set; T2PSO, PSO with Trelea's second parameter set; CFPSO, constriction factor particle swarm optimization; ABHA, artificial beehive algorithm.

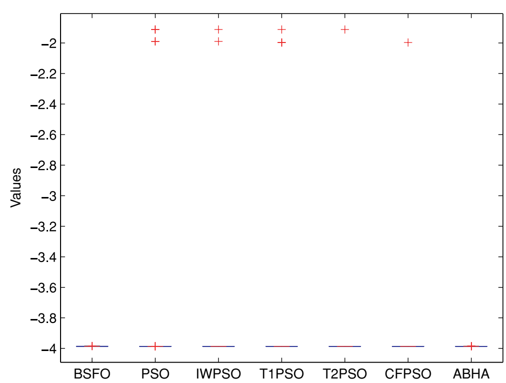

Figure 2. Box plots of the results obtained with the PFUNC function. Abbreviations as in Figure 1.

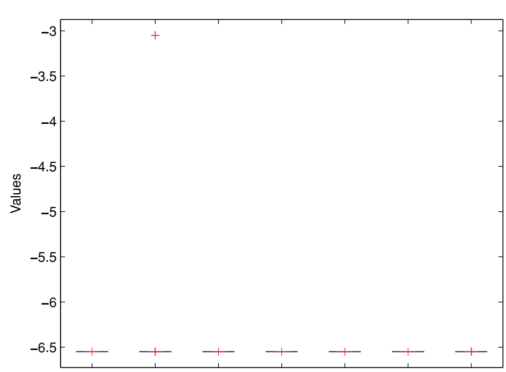

Figure 3. Box plots of the results obtained with the Peaks function. Abbreviations as in Figure 1.

Figure 5 shows the results for the Grienwank function. In this experiment, the results are very similar to that obtained for the De Jong F1 function, where the performance of the ABHA is lower than that of the PSO variations but higher than that of the BSFO algorithm.

For the Rastrigin function, as shown in Figure 6, the ABHA achieves a performance very similar to that obtained with the BSFO algorithm. Again, the PSO variations give better results. For this function, the results of the ABHA are disappointing.

The results for the Ackley function are shown in Figure 7, which show similarity with the Rastrigin function. The performance of the ABHA is roughly the same as that of the BSFO algorithm.

Figure 8 shows the results for the DeJong F2 function. For this function, the ABHA again achieves a better performance than the BSFO algorithm. Also, it achieves the global minima at least once. Although the repeatability is lower than that presented by the PSO algorithms, the performance is satisfactory.

Figure 9 shows the results for the Schaffer F6 function. Similar to the DeJong F2 function, all the tested algorithms achieve the global minimum value. Again, the performance of the ABHA is better than BSFO algorithm, by achieving a lower mean and lower outliers. Although the performance of the ABHA in comparison with the PSO algorithm is not very satisfactory, the algorithm performs within the expectations.

Finally, Figure 10 shows the results for the Schewefel function. In this function, where its objective value is a negative number, the ABHA performs somewhat better than the BSFO algorithm. Again, the PSO algorithm achieves better values but does not reach the optimal. Also, the repeatability of the ABHA is better than that obtained by the PSO algorithm, which has more spreading.

In conclusion, the figures show that the performance of the ABHA are closely related to the performance of the BSFO algorithm. It is clear that in some functions, the performance of the ABHA is not satisfactory; the response in some functions is better than BSFO; and in others, as good as the PSO algorithms. The best results with the ABHA are achieved with the PFUNC function, where it achieved better results than those shown by all the algorithms.

For a one-on-one performance comparison, we used the t test with a 95% confidence interval range.[42] The t test allows the verification of the statistical validity of the result and at the same time if the algorithm under test can be considered statistically better than the control algorithm. The t test, considered to be a signal-to-noise ratio, provides the difference between the two groups and is calculated by Equation 13, where T is the test group and C the control group, n is the number of samples and t the probability in a Student's t distribution. The t value is positive if the first mean is higher than the second and negative if it is lower.[42] If we have $n = 100$ for both groups, the lower limit for a 95% confidence interval range is 1.66055. According to t value, it is possible to conclude about the performance of the test algorithm: if $t \leq -1.66055$, the performance is better than the control algorithm; if $-1.66055 < t < 1.06605$, the performance is equal; and the performance is lower if $1.66055 \leq t$. The results are shown in Table IV, where the boldface values represent a better or equal performance of the ABHA. The results

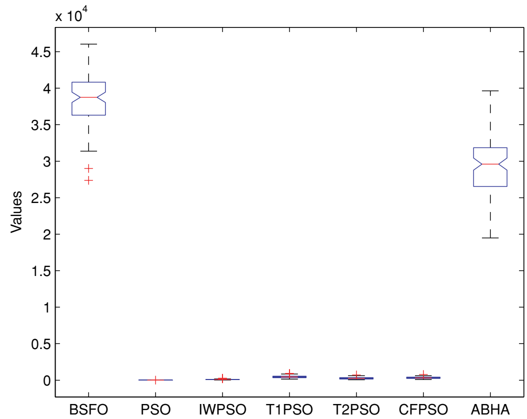

Figure 4. Box plots of the results obtained with the De Jong F1 function. Abbreviations as in Figure 1.

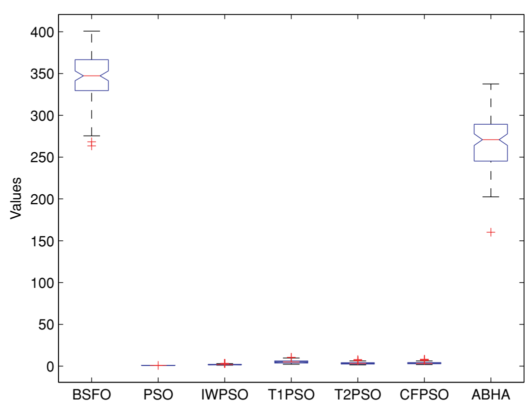

Figure 5. Box plots of the results obtained with the Grienwank function. Abbreviations as in Figure 1.

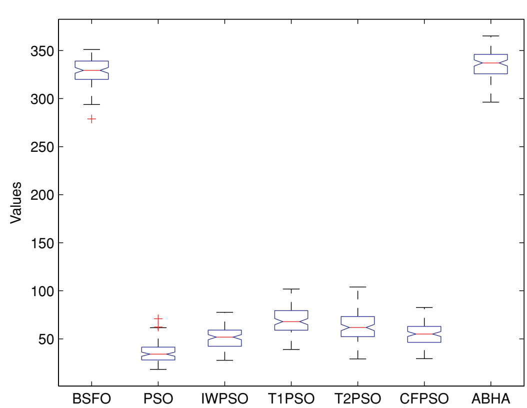

Figure 6. Box plots of the results obtained with the Rastrigin function.

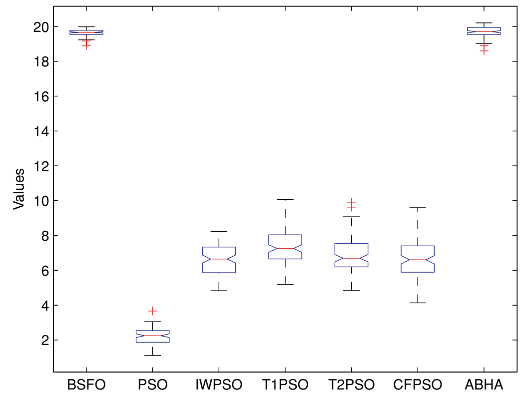

Figure 7. Box plots of the results obtained with the Ackley function. Abbreviations as in Figure 1.

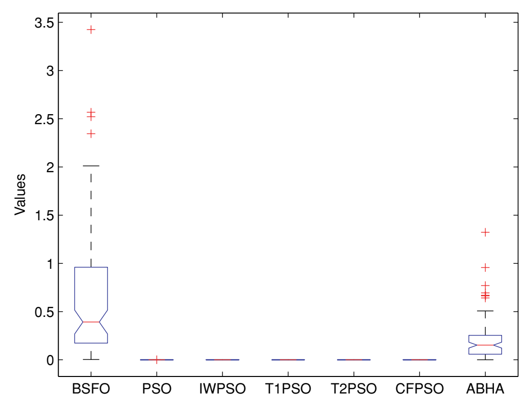

Figure 8. Box plots of the results obtained with the De Jong F2 function. Abbreviations as in Figure 1.

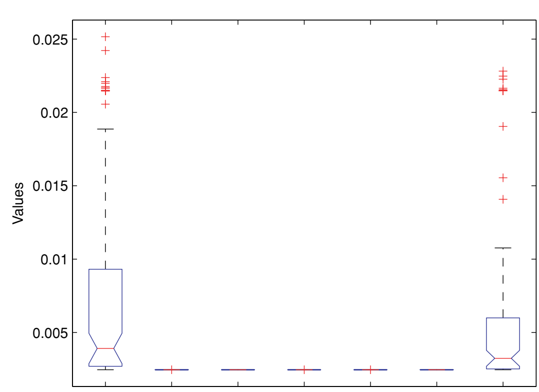

Figure 9. Box plots of the results obtained with the Schaffer F6 function. Abbreviations as in Figure 1.

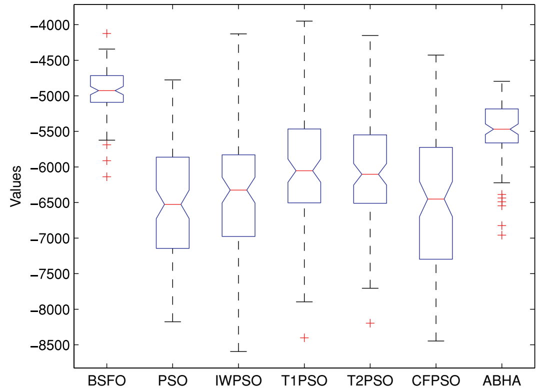

Figure 10. Box plots of the results obtained with the Schewefel function. Abbreviations as in Figure 1.

Table IV. Results of the t–test for the ABHA algorithm.

	BSFO	PSO	IWPSO	T1PSO	T2PSO	CFPSO
F1	**−7.68**	11.11	11.11	11.11	11.11	11.11
F2	**−5.19**	**−2.02**	**−1.41**	**−2.02**	**−2.02**	**−0.99**
F3	**−2.87**	**−1.41**	8.83	8.83	8.83	8.83
F4	**−17.27**	70.48	70.26	69.34	69.82	69.66
F5	**−18.09**	84.65	84.33	83.23	83.79	83.71
F6	3.67	167.52	150.92	130.59	131.71	143.76
F7	**1.45**	313.72	141.16	119.03	117.45	114.07
F8	**−6.36**	9.30	9.30	9.30	9.30	9.30
F9	**−1.84**	5.84	5.84	5.84	5.84	5.84
F10	**−10.29**	10.60	8.96	5.93	6.13	9.62

confirm the data obtained by the box and whiskers plots, where the performance of the ABHA is, in most cases, better than the performance of the BSFO algorithm. Also, the results show rather low performance of the ABHA in the more complex functions. Again, the best performance against all the test algorithms is obtained for the PFUNC function because of ABHA's low standard deviation.

$$t = \frac{\bar{x}_T - \bar{x}_C}{\sqrt{\left(\sigma_T^2/n_T\right) + \left(\sigma_C^2/n_C\right)}} \tag{3}$$

5. CONCLUSIONS

This paper presented an SI algorithm based on a beehive IO model. The algorithm uses a set of rules to represent bee behavior and includes a communication strategy based on food source profitability. The agents can be represented as state machines with four states: naive, experimented, explorer, and exploiting. In the naive and experimented states, the agents are located in the "nest" and acquire or gather information from their partners, or begin a random search. In the explorer state, the bee uses its information to find a better food source. In the exploiting state, the agent calculates the profitability of its source and decides if it is worth announcing.

The algorithm was tested against other SI algorithms. The results showed that the algorithm achieved good results, in a few cases with a better performance than the test algorithms, although on other occasions when the performance was not satisfactory, it showed good repeatability, represented by a low standard deviation value.

Further work must include thorough testing of algorithm parameters, seeking to obtain better performance in the complex functions, and also explore the possibility to perform parameter control through evolutionary computing techniques that also mimic social aspects of the bees. A possible candidate for this process is the marriage in honeybee optimization.[43] Some applications proposed would be used in parameter identification in control and identification problems, neural network, and fuzzy systems learning. Finally, a profound mathematical study of algorithm properties such as stability and convergence is required.

Acknowledgment

This work was partly supported by Colciencias and Universidad del Valle, through a graduate research scholarship awarded to Mario A. Muñoz (grant no. 1106-11-17707).

References

1. Boyd S, Vandenberghe L. "Convex Optimization." New York: Cambridge University Press 2004. Chapter 1, pp 1–15.
2. Bishop J, Torr P. The stochastic search network. In: Linggard R, Myers D, Nightingale C, editors. Neural networks for images, speech and natural language. New York, NY: Chapman & Hall; 1992. pp 370–387.
3. Beattie P, Bishop J. Self-localisation in the "senario" autonomous wheelchair. J Intell. Robot. Syst Theory Appl 1998;22:255–267.
4. Conradie A, Miikkulainen R, Aldrich C. Adaptive control utilising neural swarming. In: Proc Genetic and Evolutionary Computation Conference 2002—New York, NY, July 9–13, 2002. pp 60–67.
5. Esmin AAA, Aoki AR, Lambert-Torres G. Particle swarm optimization for fuzzy membership functions optimization. In Proc IEEE Yasmine Hammamet, Tunisia, October 6–9, 2002. pp 108–113.
6. Kim DH, Cho JH. Adaptive tuning of PID controller for multivariable system using bacterial foraging based optimization. In: Advances in Web Intelligence, Series Lecture Notes in Computer Science. Berlin: Springer/Heidelberg; 2005. Vol 3528, pp 231–235.
7. Omran MGH, Engelbrecht AP, Salman A. Dynamic clustering using particle swarm optimization with application in unsupervised image classification. Trans Eng Comput Technol 2005;9:199–204.
8. Yuan R, Guang-yi C, Xin-jian Z. Particle swarm optimization based predictive control of proton exchange membrane fuel cell (PEMFC). J Zhejiang Univ Sci A 2006;7(3):458–462.
9. de Vries H, Biesmeijer JC. Modelling collective foraging by means of individual behaviour rules in honey-bees. Behav Ecol Sociobiol 1998;44:109–124.

10. Hertz A, Kobler D. A framework for the description of evolutionary algorithms. Eur J Oper Res 2000;126(1):1–12.
11. Beni G, Wang J. "Swarm intelligence in cellular robotics systems," In: NATO Advanced Workshop Robots Biological System, vol. 1, Il Ciocco, Toscana, Italy, June 26–30, 1989, pp 703–712.
12. Bonabeau E, Dorigo M, Theraulaz G. "Swarm Intelligence: From Natural to Artificial Systems," New York, NY: Oxford University Press, 1999. Chapter 1, pp 1–23.
13. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Sixth Int Symp Micro Machine and Human Science, Nagoya, Japan, October 4–6, 1995. pp 39–43.
14. Kennedy J, Eberhart R. Particle swarm optimization In: Proc IEEE Int Conf Neural Networks, Perth, Australia, November 27–December 1, 1995. pp 1942–1498.
15. Kennedy J. The particle swarm: Social adaptation of knowledge. In: Proc IEEE Int Conf Evolutionary Computation, Indianapolis. IN, April 13–16, 1997. pp 303–308.
16. Passino KM. Distributed optimization and control using only a germ of intelligence. In: 2000 IEEE Int Symp on Intelligent Control, Patras, Greece, July 17–19, 2000. pp 5–13.
17. Liu Y, Passino KM. Biomimicry of social foraging bacteria for distributed optimization: Models, principles and emergent behaviors. J Optim Theory Appl 2002;115(3):603–628.
18. Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst Mag 2002;22(3):52–67.
19. Dorigo M, Maniezzo V, Colorni A. Ant system: An autocatalytic optimizing process. Technical Report, Politecnico di Milano, Milan, Italy, 1991;91-016.
20. Dorigo M, Di Caro G. The ant colony optimization meta-heuristic. New Ideas in Optimization, McGraw-Hill Advanced Topics in Computer Science, London: McGraw-Hill; 1999. Chap 2, pp 11–32.
21. Dorigo M, Stützle T. The ant colony optimization metaheuristic: Algorithms, applications, and advances. In: Glover F, Kochenberger G, editors. Handbook of Metaheuristics, ser. International Series in Operations Research and Management Science. Norwell, MA: Kluwer Academic Publishers; 2001. pp 251–285.
22. Colorni A, Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: Proc ECAL91—European Conf Artificial Life. Paris, France, December 11–13, 1991. pp 134–142.
23. Dorigo M, Maniezzo V, Colorni A. Positive feedback as a search strategy. Technical Report, Politecnico di Milano, Milan, Italy, 1991;91-016.
24. Dorigo M, Maniezzo V, Colorni A. The ant system: Optimization by a colony of cooperating agents. IEEE Trans Syst Man Cybern B 1996;26(1):1–13.
25. Salerno J. Using the particle swarm optimization technique to train a recurrent neural model. In: IEEE Int Conf on Tools with Artificial Intelligence, Newport Beach, CA, November 3–8, 1997. pp 45–49.
26. Voss MS, Feng X. A new methodology for emergent system identification using particle swarm optimization (PSO) and the group method of data handling (GMDH). In: Proc Genetic and Evolutionary Computation Conference 2002—GECCO 2002, New York, NY, July 9–13, 2002. pp 1227–1232.
27. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report, Erciyes University, Kayseri, Turkey, October 2005;TR06.
28. Teodorovic D, Dell'Orco M. Bee colony optimization—A cooperative learning approach to complex transportation problems. In: Proc 10th EWGT Meeting and 16th Mini-EURO Conf, Poznan, Poland, September 13–16, 2005.
29. Pham D, Ghanbarzadeh A, Koç E, Otri S, Rahim S, Zaidi M. The bees algorithm—A novel tool for complex optimisation problems. In: Innovative Production Machines and Systems Virtual Conf, Cardiff, UK, July 3–14, 2006.
30. Alfonso W, Muñoz M, López J, Caicedo E. Optimización de funciones inspirada en el comportamiento de búsqueda de néctar en abejas. In: Congreso Internacional de Inteligenicia Computacional (CIIC2007), Bogotá, Colombia, September 5–7, 2007.
31. Lucic P, Teodorovic D. Computing with bees: Attacking complex transportation engineering problems. Int J Artif Intell Tools 2003;12(3):375–394.

32. Pham D, Koç E, Ghanbarzadeh A, Otri S. Optimisation of the weights of multi-layered perceptrons using the bees algorithm. In: Proc 5th Int Symp Intelligent Manufacturing Systems, Kocaeli, Turkey 2006. pp 38–46.
33. Pham D, Darwish AH, Eldukhr E, Otri S. Using the bees algorithm to tune a fuzzy logic controller for a robot gymnast. In: Proc Innovative Production Machines and Systems Virtual Conf, Cardiff, UK. July 2–13, 2007.
34. Pham D, Muhamad Z, Mahmuddin M, Ghanbarzadeh A, Koç E, Otri S. Using the bees algorithm to optimise a support vector machine for wood defect classification. In: Proc Innovative Production Machines and Systems Virtual Conf, 2007.
35. Pham D, Afify A, Koç E. Manufacturing cell formation using the bees algorithm. In: Proc Innovative Production Machines and Systems Virtual Conf, 2007.
36. Pham D, Ghanbarzadeha A. Multi-objective optimisation using the bees algorithm. In: Proc Innovative Production Machines and Systems Virtual Conf, 2007.
37. Alfonso W. Regulación de temperatura en la plataforma uv-ptm01 basada en agentes cooperativos para la asignación dinámica de recursos. Technical Report, Universidad del Valle, Cali, Colombia, 2007.
38. Shi Y, Eberhart RC. Parameter selection in particle swarm optimization. In: Evolutionary Programming VII: Proc Seventh Annu Conf Evolutionary Programming, San Diego, CA, March 25–27, 1998. pp 591–600.
39. Trelea I. The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf Process Lett 2003;85:317–325.
40. Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proc 1999 Congress on Evolutionary Computation (CEC 99), Washington, DC, July 6–9, 1999.
41. Yisu J, Knowles J, Hongmei L, Yizeng L, Kell DB. The landscape adaptive particle swarm optimizer Appl Soft Comput 2007;8(1):295–304.
42. Allen MP. "Understanding Regression Analysis" New York: Plenum Press. Chapter 13, pp 61–65.
43. Abbass HA, Teo J. A true annealing approach to the marriage in honey-bees optimization algorithm. In: Proc inaugural workshop on Artificial Life (AL'01), Adelaide, Australia, December 11, 2001. pp 1–14.