



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Memory-based Bees Algorithm: An Enhancement

Nahlah Shatnawi, Shahnorbanun Sahran and Mohammad Faizul

Center of Computer and Information Technology, Faculty of Information System and Technology,
Universiti Kebangsaan Malaysia, 43650 Bandar Baru Bangi, Malaysia

Abstract: The Bees Algorithm (BA) is a new population-based optimization algorithm inspired by the foraging nature of bees. In the basic version of the Bees Algorithm, the algorithm performed a combination of neighborhood search and global search. However, the current BA has the disadvantage of not fully imitate all physical and social aspect of bees' nature. In this study, enhancements to the BA will be introduced as Memory-based Bees Algorithm (MBA) by adding memory (local and global) to two types of bees to make the algorithm more natural. The results of comparing the proposed Local-MBA, global-MBA and MBA (combination of Local-MBA and global-MBA) are tested using several benchmark functions. They had obtained approximately 59.34, 73.02, 74.9 and 75.44% improvement on mean number of evaluations over the basic BA, respectively. Novel fitness values of two engineering design problems are obtained by applying MBA. The proposed algorithms have great potential to be used in many optimization problems.

Key words: Bees algorithm, local memory, global memory, benchmark functions, welded beam, spring design

INTRODUCTION

The BA algorithm is one of many biologically inspired algorithms. It is a new population-based algorithm that has been proposed to overcome the local optima problem and is applicable to both combinatorial and functional optimization problems (Pham *et al.*, 2006). The BA follows the behavior of honey bees in food foraging. It uses neighborhood search with random search to find best food sources. The BA depends on exchange of information among a large number of scout and follower bees, where these bees' based on the information received from each to search and find new plentiful sources. The BA has proved to be the most powerful fair optimization method for sampling a large solution space because of its fair random sampling. Some other nature-based algorithms are the Ant Colony Optimization (ACO), the Particle Swarm Optimization (PSO), the Artificial Bee Colony algorithm (ABC) and the Honey Bee Mating Optimization (HBMO) algorithm (Pham *et al.*, 2007).

The BA is used to solve many problems such as control chart pattern recognition, job scheduling, data clustering and function optimization. Popov *et al.* (2006), performed experiments on the basic BA using eight benchmark functions and their results showed that the basic BA could reliably handle complex multi-model

optimization problems without being trapped at local optima. This study also revealed that the basic BA generally outperformed other techniques in terms of speed and accuracy of the results. However, the basic BA has two disadvantages: (1) A small No. of trials are needed to set the parameter values and (2) A No. of tunable parameters are used (Pham *et al.*, 2007). The enhanced BA will be used to solve these problems by adding memory to store the social information of bees in nature that they use to choose the patches to visit. These were not included in the basic BA. By adding this natural behavior of bees to the algorithm, the performance is improved.

The objective of the study is to enhance basic BA algorithm by adding two types of memory including: private and social information, which meant for individual bee and the colony, respectively. This enhancement intended to copy the decision making capability of bees.

Bees in nature: Bees live in colonies that can span more than 10 km in different directions with many food sources (Pham *et al.*, 2006). In a colony, the scout bees search for food in the following ways (Gruter and Farina, 2009): (1) Using private information which is "the information acquired via direct interaction with the environment" (Gruter and Farina, 2009). For example, many plants offer nectar during some periods of the day and bees can learn

these times of food existence after a few days (Lloyd, 1983; Rowe, 1999; Biesmeijer and de Vries, 2001); (2) Using social information which is “any information acquired through the actions (e.g., waggle dance), body structures (e.g., shapes or colors) or products (e.g., pheromones) of other individuals” (Pham *et al.*, 2007) or (3) Using innate search behaviors to find the best patch (patches with a plentiful amount of nectar or patches that need less effort to collect the nectar).

Based on the information exchanged by the scout bees using the waggle dance on the dance floor, the following information can be obtained (Gil and Farina, 2002): (1) The direction of the patch (based on the sun) (2) The distance (based on the duration of the dance) and (3) The quality (fitness, the amount of nectar). The waggle dance sends different signals with different objectives. The objective of each signal can be one of the following (Pham *et al.*, 2007): (1) Gives different types of information to the surrounding bees, (2) Informs the follower bees about good food sources, (3) Activates private navigational information (if present) and (4) Identifies the location of the food source.

Another type of bees, called follower bees use the scout bees’ information or private information (if it is reliable and present) and social information to go to the best patches. More bees go to the best patches and fewer bees go to the poor patches. A follower bee in a patch collects nectar and monitors the nectar amount. Based on the amount, the bee decides if the patch is still good or not in the next waggle dance. The follower bee based on (Beekman *et al.*, 2007) can (1) be an uncommitted follower, (2) continue the collection of nectar from the patch only, or (3) dance and tell more bees to go to the patch.

Basic BA: The BA is an optimization algorithm based on the nature of bees foraging for food, which was described in the previous section. The steps of the algorithm are (Pham *et al.*, 2006):

-
- 1 Initialize the population with a set number of scout bees distributed in the search space randomly
 - 2 Evaluate the fitness of the population
 - 3 While (the stopping criterion is not met)
 - 4 Select the highest points of fitness (by neighborhood search) to be visited by the bees
 - 5 Send bees to the selected points of fitness (more bees to the best points) and evaluate
 - 6 Select the bee with higher fitness from each patch
 - 7 Assign remaining bees to the search space randomly and evaluate
 - 8 End while
-

The BA requires seven control parameters: the number of scout bees n_s , the number of sites selected out of n visited sites m , the number of best sites out of m selected sites e , the number of bees recruited for the

best e sites n_{ep} , the number of bees recruited for the other $(m-e)$ selected sites n_{sp} , the initial size of patches around each scout bee n_{gh} and the maximum iteration number.

Enhancements of the BA: Many enhancements have been added to the BA. Although these enhancements are logical, they do not necessarily describe what happens in nature. The BA enhancements are as follows:

- Pheromones by Packianather *et al.* (2009). They employ bee pheromones to find good food patches. Their work depends on the fact that there are two types of pheromones, releaser pheromones (short term effects) and premier pheromones (long term effects). The releaser pheromones change the behavior of the recipient bee, whereas the primer pheromones change the physiology. They use the pheromones to recruit bees to specific regions, where the amount of pheromones besides the fitness of each patch is used
- Neighborhood shrinking by Ghanbarzadeh (2007). In this study, the authors decrease the n_{gh} parameter value when the algorithm reaches a state where there is little to no improvement in its best candidate solution, thus increasing the solution density around each site
- Site abandoning by Ghanbarzadeh (2007). The site is replaced with a new site when there is no improvement seen even if the new one has a lower fitness. The replaced site will be stored temporarily so that it can be retrieved if necessary
- Fuzzy greedy selection-based BA by Pham and Darwish (2008). The basic BA uses greedy selection to choose the m best patches and the e elite patches out of the patches explored by the scout bees

MEMORY-BASED BA (MBA)

As mentioned before, many enhancements have been made to the BA, but these algorithms do not mimic all of the natural behaviors of honey bees for food foraging. Based on the experiments described by Bergen *et al.* (2004), Partan and Marler (2005) and Leadbeater and Chittka (2007), there are foraging behaviors used by honey bees that not taken into account in basic BA. The bees that do not follow the dancers find food patches using their memories (Rowe, 1999), “93% of all foragers with private information about the location of a good food source ignored the dance language” (Grutere *et al.*, 2008). The bees also use social information. In addition, the honey bees have photographic memories (Gil and Farina, 2002).

Based on these observations, basic BA will be enhanced by adding two types of memory: local memory (for private information) and global memory (for social information). These memories will be added to the scout bees along with the follower bees where these bees could (Gruter and Farina, 2009) (1) Use private information about previously visited patches (2) Use social information and (3) Use innate search behaviors. Before adding the local and global memories, the basic BA will be recoded making it object-oriented. The Object-Oriented BA (OOBA) is as follows:

```

Initialize the main parameters, objects and data members; Start
1 Call the initialization method to initialize the population of scout bees
  randomly in the search space and calculate the fitness of each bee
2 While (the stopping criterion is not met); // stop when finding the
  optimum value
3 Use the sort method to reorder bees based on their fitness
4 Select the highest fitness points using the choose best m method
5 Use the recruitment method to determine the number of bees that will
  be sent to different patches (more bees to the elite sites)
6 Send the follower bees to the selected points randomly and evaluate
  the fitness using the search neighbor method
7 Send the rest of scout bees to search for new sites randomly and
  evaluate their fitness in the rest random method
8 End while
    
```

End in OOBA the following structs used to save and handle the memory of each bee:

```

1 Struct dimension // to store the position in different dimensions
2 Struct memory: dimension // to store the fitness and position of each
  bee (scout and follower), to store and retrieve information from the
  memory of bees
3 Struct bees: memory // to store the id of each bee, status (random/
  best), type (follower/ scout)
    
```

The following variables are used: imax is the maximum number of iterations, iter is the iteration count, mem-fit and mem- bee-pos are arrays used to store the fitness and position of each bee in their memories, id is the identity of the bee being processed, status tracks if this bee will go randomly or if it is one of the best scout bees and type indicates if this bee is a scout bee or a follower bee. Set methods are used to store information and get methods are used to retrieve information. The next two subsections contain an explanation of local and global memory.

Local memory (private information): Local memory or private information is the information acquired from direct interaction with the environment. This information affects the way forager bees (both scout and follower bees) choose the patch to visit and whether it will be based on a waggle dance or not (i.e., choosing a familiar food source). Foragers depend on their own memories to find particular locations when visiting food patches repeatedly

(Gruter and Farina, 2009). Private information is important, especially when food patches offer food for several days. The following pseudo code shows the main methods that added for the local memory. The first code is for the near method which determines if the new position of any bee is near its previous one stored in its memory. The naming of this method for follower bees is included in the search neighbor () method and for scout bees in the rest-rand () method.

```

// determine if the new position of any bee is near its previous position
// stored in memory or not using the near method
1 Determine if the bee is a scout bee (if yes) do
  radius = x1*end-x [dim]-start-x [dim]; // for scout bees
  otherwise radius = x2 * ngh; // for follower bees
2 While (the number of iterations less than imax) do
3 For the position in different dimensions, use the Euclidian distance to
  determine if the new position is near its previous position stored in
  the private memory
4 If yes, then find a new position randomly
5 End while
    
```

The radius value for scout bees depends on the solution space size in different dimensions and on the neighborhood size ngh for follower bees, which means that the value of radius will be more than ngh for scout bees and less than ngh for follower bees. The values of x1 and x2 are set by trial and error. Figure 1 shows how various x2 values affect the mean number of evaluations. Another method used to compare private memory (compare fitness) is to see if the bee was in better place before. If it was, then go back to that place. This method is used for both the scout and follower bees and it is called in the iterations () method.

Global memory (social information): In a colony, the scout bees search using private information or social information. Social information is used as a back-up if the private information has proven to be unreliable (Bergen *et al.*, 2004; Laland, 2004; Leadbeater and Chittka, 2007) or outdated or if individuals evaluate this option as

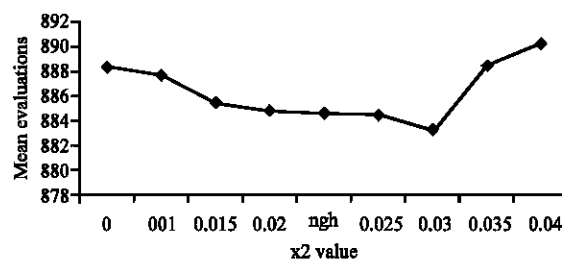


Fig. 1: Effect of choosing x2 variable value on the mean number of evaluations

unbeneficial. Given the fact that experiments on waggle dances are normally performed after the end of the flowering season or in places with few food sources, one would expect private information to be outdated or not available and that foragers would rely more on social information (Gruter and Farina, 2009). Based on the previous observation the importance of social information was shown. Therefore, method to compare the global memory of different scout bees (except the best bees: (m) was added to determine if the present position for any bee is near to any previous position. To do this first store all the information (the position in different dimensions and the fitness of each position) in different iterations for all the bees is needed. See the pseudo code below:

```

// compare global memory for bee with id = i
if the bee status is a random bee do
    while there is information in the all info array do
        for the different dimensions find the Euclidian distance
        if the distance is less than 3* ngh
            then it is near to its previous position and it need to find
            another position randomly
        end while
    end if

```

In compare global memory method the distance between two bees compared with (3* ngh) value because of wanting to give more options for the scout bees to find new places that are not stuck on local optima. The call of this method is included in the rest-random method. The next section includes the experimental results.

EXPERIMENTS AND RESULTS

To evaluate the effectiveness of the enhancements (adding local and global memory) to the basic BA, the basic BA, Local-MBA, global-MBA and MBA (combination of Local-MBA and global-MBA) were applied to nine benchmark functions: De Jong, Goldstein and Price, Branin, Martin and Gaddy, Rosenbrock (a and b), Rosenbrock, Hyper Sphere and Griewangk and to two engineering optimization problems: Welded Beam and Spring Design (Yang and Deb, 2010). In these experiments visual C++ 2005 used. The next table shows the values for the BA parameters when applied to the nine benchmark functions and the two engineering optimization problems: welded beam and spring design.

The parameters include the initial population n, the number of selected sites m, the number of elite sites e, the number of bees recruited for best e sites nep, the number

of bees recruited for the other (m-e) selected sites nsp and the size of the patches ngh. The dimension denotes the number of dimensions used for each benchmark function and optimization problem. The basic BA and MBA are applied to different problems with different numbers of dimensions and the problems become more difficult with an increasing number of dimensions. The following table shows the results of applying the basic BA, local memory-based BA, global memory-based BA and local and global memory-based BA using the values in Table 1.

In Table 2, the mean number of evaluation results that are obtained when applying the basic BA, local memory-based BA, global memory-based BA and local and global memory-based BA to nine benchmark functions shown. The results show the superiority of the memory-based BA algorithm and how the addition of local memory and global memory affects the results. The best results are for the Branin benchmark function in 2 dimensions when applying local memory, global memory and both local and global memories. The second best result is for the Goldstien and Price benchmark function in 2 dimensions. The third best result is for the Rosenbrock benchmark function in 10 dimensions. The average improvement percentage for local memory is 59.34 and 73.02% for global memory and the best s 74.9% when applying local and global memory together for scout and follower bees.

Tables 3 and 4 show the results when applying the basic BA and memory-based BA (local and global) to the welded beam and spring design problems. It is clear that applying the basic BA gives better results compared with Cagnina *et al.* (2008) and Yang and Deb (2010) when applying evolutionary and Cuckoo Search on those two problems, respectively. The memory-based BA is superior in fitness values and all other variables. Novel fitness values of two engineering design problems are obtained by applying Memory-based BA algorithm (MBA).

Table 1: Parameters values used in basic BA, Local-MBA, global-MBA and MBA algorithm for benchmark functions and the two engineering optimization problems

Benchmark function	Engineering problem						
	Dimension	n	m	e	nsp	nep	ngh
De Jong	2d	10	2	1	2	4	0.100
Goldstein and Price	2d	10	2	1	1	2	0.030
Branin	2d	30	5	1	2	3	0.200
Martin and Gaddy	2d	10	5	1	1	10	0.130
Rosenbrock (a)	2d	10	3	1	2	4	0.120
Rosenbrock (b)	2d	6	3	1	1	4	0.200
Rosenbrock	4d	10	3	1	1	3	0.100
Hyper Sphere	6d	8	3	1	1	2	0.050
Griewangk	10d	10	3	2	2	4	0.200
Welded Beam	4d	10	5	2	2	4	0.080
Spring Design	3d	6	5	5	1	8	0.001

Table 2: Mean number of evaluations results of basic BA, Local-MBA, Global-MBA and MBA, along with the improvement percentage of each comparing with basic BA when applied to nine benchmark functions

Benchmark function	Basic BA	Local-MBA		Global-MBA		MBA	
	Mean	Mean	Improvement (%)	Mean I	Improvement (%)	Mean	Improvement (%)
Dejong	1860.51	868.0	53.30	682	63.34	654	64.85
Goldstien and Price	11488.72	2190.0	80.90	1897	83.49	1002	91.28
Branin	15909.39	337.7	97.88	332	97.90	320	97.98
Martin and Gaddy	449.20	216.0	51.90	112	70.07	110	75.50
Rosenbrock (a)	1195.00	525.6	56.00	597	50.04	604	49.45
Rosenbrock (b)	7031.60	2213.0	68.50	1161	83.49	1154	83.59
Rosenbrock	19524.50	17386.0	11.00	2855	85.38	2464	87.38
Hypersphere	2406.60	898.4	62.60	881	63.39	889	63.06
Grienwangk	1936.80	1152.5	40.50	773	60.09	755	61.02
Average			59.34		73.02		74.90

Table 3: Welded beam results of Cagnina *et al.* (2008), Yang and Deb (2010), basic BA and MBA

Results	Cagnina <i>et al.</i> (2008) and Yang and Deb (2010)	Basic BA	MBA
Fitness	1.724852308597360	1.7320483620	1.723867352504770
W: width of the welded area	0.205729639786070	0.2069460130	0.709069375896481
L: length of the welded area	3.470488665627970	3.4733726000	1.800542738120670
D: depth of the main beam	9.036623910357630	9.0111453600	2.166288079470200
H: thickness of the main beam	0.205729639786079	0.2069551680	0.171830868251595

Table 4: Spring design results of Cagnina *et al.* (2008), Yang and Deb (2010), basic BA and MBA

Results	Cagnina <i>et al.</i> (2008) and Yang and Deb (2010)	Basic BA	MBA
Fitness	0.012665	0.0116658572745262	0.0115733933638806
W: wire diameter	0.051690	0.0888954435865352	0.0886629372539445
D: the mean coil diameter	0.356750	0.2982898953215120	0.2975059560838650
L: the length (or number of coils)	11.287126	2.9490189214758800	2.9485825619068000

CONCLUSION

The BA is one of the meta-heuristic algorithms that is used to solve combinatorial and functional problems. It was proposed based on the honey bees' nature to solve local optima problems and many enhancements have been made to this algorithm. Some of these enhancements are based on natural behaviors; others are not.

In this study, a new enhancement to the basic BA by adding private information (local memory) and social information (global memory) were presented. These types of memory are part of the honey bees' nature but were not included in basic BA. These two types of memory added for both the follower and the scout bees. The results of applying the BA before and after the enhancements on nine benchmark functions show that the memory-based BA outperforms the basic BA and the improvement percentage exceeds all previous enhancements results (Pham *et al.*, 2006; Packianather *et al.* 2009; Zaidi *et al.*, 2011). Additionally, the results of applying the memory-based BA on the welded beam and spring design engineering problems are novel results exceed the previous results of Yang and Deb (2010).

REFERENCES

Beekman, M., A.L. Gilchrist, M. Duncan and D.J.T. Sumpter, 2007. What makes a honeybee scout? *Behav. Ecol. Sociobiol.*, 61: 985-995.

Bergen, Y.V., I. Coolen and K. Laland, 2004. Nine-spined sticklebacks exploit the most reliable source when public and private information conflict. *Proc. Biol. Sci.*, 271: 957-962.

Biesmeijer, J.C. and H. de Vries, 2001. Exploration and exploitation of food sources by social insect colonies: A revision of the scout-recruit concept. *Behav. Ecol. Sociobiol.*, 49: 89-99.

Cagnina, L.C., S.C. Esquivel and C.A. Coello, 2008. Solving engineering optimization problems with the simple constrained particle swarm optimizer. *Informatica*, 32: 319-326.

Ghanbarzadeh, A., 2007. The bees algorithm: A novel optimization tool. Ph.D. Thesis, Manufacturing Engineering Centre, Cardiff University, UK.

Gil, M. and W.M. Farina, 2002. Foraging reactivation in the honeybee *Apis mellifera* L.: Factors affecting the return to known nectar sources. *Naturwissenschaften*, 89: 322-325.

Grueter, C. and W.M. Farina, 2009. The honeybee waggle dance: Can we follow the steps? *Trends Ecol. Evol.*, 24: 242-247.

Grueter, C., M. Balbuena and W.M. Farina, 2008. Informational conflicts created by the waggle dance. *Proc. R. Soc. B: Biol. Sci.*, 275: 1321-1327.

Laland, K.N., 2004. Social learning strategies. *Learn. Behav.*, 32: 4-14.

- Leadbeater, E. and L. Chittka, 2007. Social learning in insects-from miniature brains to consensus building. *Curr. Biol.*, 17: R703-R713.
- Lloyd, J.E., 1983. Bioluminescence and communication in insects. *Annu. Rev. Entomol.*, 28: 131-160.
- Packianather, M.S., M. Landy and D.T. Pham, 2009. Enhancing the speed of the bees algorithm using pheromone-based recruitment. Proceedings of the 7th IEEE International Conference on Industrial Informatics, June 23-26, 2009, Cardiff, Wales, pp: 789-794.
- Partan, S.R. and P. Marler, 2005. Issues in the classification of multimodal communication signals. *Am. Nat.*, 166: 231-245.
- Pham, D.T., A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim and M. Zaidi, 2006. The bees algorithm-a novel tool for complex optimization problems. Proceedings of the 2nd International Virtual Conference on Intelligent Production Machines and Systems, July 3-14, 2006, Elsevier, Cardiff, pp: 454-459.
- Pham, D.T., S. Otri, A. Afify, M. Mahmuddin and H. Al-Jabbouli, 2007. Data clustering using the bees algorithm. Proceedings of the 40th CIRP International Seminar on Manufacturing Systems, May 30-June 1, 2007, Liverpool, UK.
- Pham, D.T. and A.H. Darwish, 2008. Fuzzy selection of local search sites in the bees algorithm. Proceedings of the 4th Virtual International Conference on Intelligent Production Machines and Systems, July 1-14, 2008, Cardiff, UK., pp: 391.
- Popov, K.B., S.S. Dimov, D.T. Pham and A. Ivanov, 2006. Micromilling strategies for machining thin features. *Proc. Inst. Mech. Eng., Part C*, 220: 1677-1684.
- Rowe, C., 1999. Receiver psychology and the evolution of multi component signals. *Anim. Behav.*, 58: 921-931.
- Yang, X.S. and S. Deb, 2010. Engineering optimization by cuckoo search. *Int. J. Math Model. Num. Optim.*, 1: 330-343.
- Zaidi, M., M. Mahmuddin, M.F. Nasrudin and S. Sahran, 2011. Local search manoeuvre recruitment in the bees algorithm. Proceedings of 3rd International Conference on Computing and Informatics, June 8-9, 2011, Bandung, Indonesia, pp: 43-48.