A Bumble Bees Mating Optimization Algorithm for Global Unconstrained Optimization Problems

Yannis Marinakis, Magdalene Marinaki, and Nikolaos Matsatsinis

Abstract. A new nature inspired algorithm, that simulates the mating behavior of the bumble bees, the Bumble Bees Mating Optimization (BBMO) algorithm, is presented in this paper for solving global unconstrained optimization problems. The performance of the algorithm is compared with other popular metaheuristic and nature inspired methods when applied to the most classic global unconstrained optimization problems. The methods used for comparisons are Genetic Algorithms, Island Genetic Algorithms, Differential Evolution, Particle Swarm Optimization, and the Honey Bees Mating Optimization algorithm. A high performance of the proposed algorithm is achieved based on the results obtained.

1 Introduction

In the last years, several biological and natural processes have been influencing the methodologies in science and technology in an increasing manner. Among the most popular nature inspired approaches, when the task is optimization within complex domains of data or information, are those methods representing successful animal and micro-organism team behaviour, such as the Particle Swarm Optimization [14], the artificial immune systems [6], the Ant Colony Optimization [7], etc. Also, a number of swarm intelligence algorithms, based on the behaviour of the bees have

Yannis Marinakis · Nikolaos Matsatsinis
Decision Support Systems Laboratory,
Department of Production Engineering and Management,
Technical University of Crete, 73100 Chania, Crete, Greece
e-mail: {marinakis,nikos}@ergasya.tuc.gr

Magdalene Marinaki
Industrial Systems Control Laboratory,
Department of Production Engineering and Management,
Technical University of Crete, 73100 Chania, Crete, Greece
e-mail: magda@dssl.tuc.gr

J.R. González et al. (Eds.): NICSO 2010, SCI 284, pp. 305–318, 2010.
springerlink.com

been presented [4]. These algorithms are divided, mainly, in two categories according to their behaviour in the nature, the foraging behaviour and the mating behaviour. The most important approaches that simulate the foraging behaviour of the bees are the Artificial Bee Colony (ABC) Algorithm proposed by [13], the Virtual Bee Algorithm proposed by [27], the Bee Colony Optimization Algorithm proposed by [24], the BeeHive algorithm proposed by [26], the Bee Swarm Optimization Algorithm proposed by [8] and the Bees Algorithm proposed by [22]. Contrary to the fact that there are many algorithms that are based on the foraging behaviour of the bees, the main algorithm proposed based on the marriage behaviour is the Honey Bees Mating Optimization Algorithm (HBMO), that was presented in ([1, 2]) and simulates the mating process of the queen of the hive. Since then, it has been used on a number of different applications [3, 10, 11, 15–20, 23].

In this paper, a new algorithm that simulates the mating behavior of the Bumble bees, the ***Bumble Bees Mating Optimization (BBMO)*** algorithm is presented, analysed and used for solving unconstrained optimization problems. This algorithm is a population-based swarm intelligence algorithm that simulates the mating behavior that a swarm of bumble bees perform. An hybridized initial version of the algorithm was presented in [21] for clustering. The other methods used for the comparisons are the Genetic Algorithms [12] and their variants, the Differential Evolution [25], the Particle Swarm Optimization [14] and its variants and the Honey Bees Mating Optimization algorithm [1, 3]. The test functions used are the Rosenbrock, the Sphere, the Rastrigin and the Griewank.

The rest of the chapter is organized as follows. In the next section an analytical description of the proposed algorithm is given. In the third section, the test functions used are given while in the fourth section computational results are presented and analyzed. The last section gives the conclusions and future research.

2 The Proposed Bumble Bees Mating Optimization Algorithm

In this section, initially, the bumble bees behavior is presented, while in the following the proposed algorithm based on this behavior is presented and analyzed in detail.

2.1 Bumble Bees Behavior

Bumble bees are social insects that form colonies consisting of the queen, many workers (females) and the drones (males). Queens are the only members of the nest to survive from one season to the next, as they spend the winter months hibernating in a protected underground overwintering chamber. Upon emerging from hibernation, a queen collects pollen and nectar from flowers and searches for a suitable nest site and when she finds such a place, she prepares wax pots to store food and wax cells into which eggs are laid ([28–31]).

The bumble bee queen can lay fertilized or unfertilized eggs. The fertilized eggs have chromosomes from the queen and a male or males she mated with the previous year and they develop into workers while the unfertilized eggs contain chromosomes from the queen alone and they develop into males. After the emergence of the first workers, the queen no longer forages as the workers take over the responsibilities of collecting food (foragers) and the queen remains in the nest laying eggs and tending to her young. Some workers, also, remain in the nest and help raise the brood (household workers). Males do not contribute in collecting food or helping rear young as the sole purpose of the males are to mate with the queens. Bumble bee workers are able to lay haploid eggs when the queen's ability to suppress the workers' reproduction diminishes. These eggs are developed into viable male bumble bees ([28–31]).

A few days after the males leave the nest, new queens will emerge. After new queens and males have gone, the colony begins to deteriorate. The founder queen stops laying eggs and grows weak from old age while the remaining workers continue to forage for food but only for themselves. Away from the colony, the new queens and males live off nectar and pollen and spend the night on flowers or in holes. The queens are eventually mated (often more than once), the sperm from the mating is stored in spermatheca and she searches for a suitable location for diapause. Three different mating behaviors exist in bumble bees. The first mating behavior is where a male perches on a tall structure and waits for queens to fly by and he will pursue them for mating once one queen is spotted. The second mating behavior is when males create a scent trail, marking their flight path with pheromones and, thus, queens of the same species will be attracted to the pheromones and follow the scent trail. The third mating behavior is where males wait at the entrance of a bumble bee nest for queens to leave ([28–31]).

2.2 BBMO for Global Unconstrained Optimization

In the BBMO algorithm, there are three kind of bumble bees in the colony, the queen, the workers and the drones (males). Initially, a number of bees are selected randomly. Each bee (a bee corresponds to an individual in the population) represents a candidate solution of the problem. Let n be the total number of variables. The bees are represented by vectors of dimension n. We use a real valued representation where initially the values of each of the bees are random numbers between 0 and 1. Afterwards, the fitness of each bee is calculated using each one of the test functions depending of the problem (see section 3) and the best bee is selected as the queen. All the other bees in the initialization phase of the algorithm are the drones.

The queen selects the drones that are used for mating by using the second mating behavior where it is assumed in the algorithm that the fittest males let larger amount of pheromone in their flight paths and, thus, the queen selects the most promising paths. This procedure is realized by sorting of all drones based on their fitness function. Each time the queen successfully mates with a drone, the genotype of the drone is stored in her spermatheca until the maximum number of matings has been reached.

After the mating, the queen finds a place to hibernate and in the next year (a year corresponds to an iteration) finds a place to create the hive and to begin to lay eggs. There are three kinds of bees that a queen lays: new queens, workers and drones. The first two kinds of bees are created by crossover of the genotype of the queen and the genotype of the drones using a specific crossover operator. In this crossover operator, the points are selected randomly from the selected drones and from the queen. Thus, initially a crossover operator number is selected (Cr_1) that controls the fraction of the parameters that are selected for the drones and the queen. The Cr_1 value is compared with the output of a random number generator, $rand_i(0,1)$. If the random number is less or equal to the Cr_1 the corresponding value is inherited from the queen, otherwise it is selected, randomly, from the solutions of one of the drones' genotypes that are stored in spermatheca. Thus, if the solution of the brood i is denoted by $b_{ij}(t)$ (t is the iteration number and j is the dimension of the problem ($j = 1, \cdots, n$)), the solution of the queen is denoted by $q_j(t)$ and the solution of the drone k is denoted by $d_{kj}(t)$:

$$b_{ij}(t) = \begin{cases} q_j(t), & \text{if } rand_i(0,1) \leq Cr_1 \\ d_{kj}(t), & \text{otherwise.} \end{cases} \tag{1}$$

The fittest of the broods are selected as new queens while the rest are the workers. The new queens are selected to be equal to the maximum number of the queens. Initially, the new queens are fed from the old queen (or queens) and, afterwards, from the workers and the old queen (or queens). The reason that we use this procedure is to improve the genotype (solution) of each new queen. This is achieved by using a local search phase where each new queen selects which of the workers and the old queen (or queens) are going to feed her by using the following equation:

$$\begin{aligned} nq_{ij} = nq_{ij} + (b_{max} - \frac{(b_{max} - b_{min}) * lsi}{lsi_{max}}) * (nq_{ij} - q_j) + \\ \frac{1}{M} * \sum_{k=1}^{M} (b_{min} - \frac{(b_{min} - b_{max}) * lsi}{lsi_{max}}) * (nq_{ij} - w_{kj}) \end{aligned} \tag{2}$$

where nq_{ij} is the solution of the new queen i, q_j is the the solution of the old queen (or queens), w_{kj} is the solution of the worker, M is the number of the workers that each queen selects for feeding her and it is different for each queen, b_{max}, b_{min} are two parameters with values in the interval $(0,1)$ that control if the new queen is fed from the old queen (or queens), from the workers or from both of them, lsi is the current local search iteration and lsi_{max} is the maximum number of local search iterations. Initially, the new queens are fed more from the old queen (or queens) and as the local search iterations increase, then only the workers feed the new queen. The appropriate choice of the values of b_{max} and b_{min} controls the feeding process, i.e. in order to have the feeding process described previously, a large value for b_{max} and a value almost equal to zero for b_{min} are necessary. Afterwards, the new queens leave from the hive.

The drones are produced by mutate the old queen's (or old queens') genotype or by mutate the fittest workers' genotype using a random mutation operator. In this mutation operator, the changes in the genotype of the old queens or the workers are performed randomly.

The drones, then, leave from the hive and they are looking for new queens for mating. As the drones leave from the hive they are moving in a swarm in order to find the best places to wait for the new queens to find them by their marked flight paths. The movement of the drones away from the hive is calculated from the following equation:

$$d_{ij} = d_{ij} + \alpha * (d_{kj} - d_{lj}) \tag{3}$$

where d_{ij}, d_{kj} and d_{lj} are the solutions of the drones i, k, l respectively and α is a parameter that determines the percentage that the drone i is affected by the two other drones k and l. The new queen select the drones that are used for mating by the procedure described previously. In the next generation, the best fertilized queens survive and all the other members of the population die. A pseudocode of the proposed algorithm is presented in Table 1.

It should be noted that the proposed Bumble Bees Mating Optimization (BBMO) algorithm that is inspired from the mating behavior of the bumble bees, it has a number of differences compared to another nature inspired algorithm that is based on the mating behavior of honey bees, the Honey Bees Mating Optimization (HBMO) algorithm [1, 15, 20]. The *Honey Bees Mating Optimization* algorithm simulates the mating process of the queen of the hive, where there are three kinds of bees, the queen, the drones and the workers. The mating process of the queen begins when the queen flights away from the nest performing the mating flight during which the drones follow the queen and mate with her in the air. The main differences of the two algorithms are:

- In the BBMO the workers are different solutions while in the HBMO they are local search phases. This helps the exploration abilities of the population by searching in different places in the solution space.
- In the BBMO after the mating of the queen three kinds of bumble bees are produced, the new queens and the workers (by using a crossover operator) and the drones (by using a mutation operator). On the other hand, in the HBMO after the mating of the queen two kinds of honey bees are produced, the queen and the drones (both of them by using a crossover operator). By using in the proposed algorithm a mutation operator to produce new solutions we have the possibility to obtain completely different solutions.
- In the BBMO the fittest of the broods produced by the crossover operator are the new queens and all the others are the workers while in the HBMO the fittest of the broods is the new queen and all the others are the drones.
- In the BBMO the drones are produced by mutation of the queen or by mutation of the fittest workers. In the HBMO the drones are all the bees produced by the crossover operator except of the queen. By using in the proposed algorithm

Table 1 Bumble Bees Mating Optimization Algorithm

```
Algorithm Bumble Bees Mating Optimization Algorithm
Definition of parameters for the main phase of the algorithm
    Definition of the maximum number of iterations
    Definition of the maximum number of matings
    Definition of the maximum number of queens
Initialization Phase
    Generate the initial population of the bumble bees
    Calculation of the fitness function of each bumble bee
    Selection of the bee with the best fitness function as the queen
    Selection of the rest bees as the drones
    Sorting the drones according to their fitness' functions
    Selection of the drones for mating by the queen
    Storing the drones' genotype to queen's spermatheca
Main Phase
do while the maximum number of iterations has not been reached
    Creation of the broods by using a crossover operator
    Calculation of the fitness function of each brood
    Sorting the broods according to their fitness' functions
    Selection of the best broods as the new queens
    Selection of the rest broods as the workers
    Feeding of the new queens by the old queens and the workers
    Creation of a percentage of the drones by mutating of the old queens' genotypes
    Creation of the rest of the drones by mutating of the workers' genotypes
    Calculation of the fitness function of each drone
    Calculation of the moving direction of the drones away from the hive
    Sorting the drones according to their fitness' functions
    do while the maximum number of matings for each new queen has not been reached
        Selection of the drones for mating by each new queen
        Storing the drones' genotypes to each new queen's spermatheca
    enddo
    Survival of the new queens for the next iteration
    Dying of all the other members (workers and drones) of the population
enddo
return The best queen (best solution found)
```

a mutation operator to produce new solutions we have the possibility to obtain completely different solutions.

- In the BBMO the drones are moving away of the hive and this affects their solutions.
- The feeding procedure in the BBMO is as described previously using the Equation (2) while in the HBMO the feeding procedure is local search phases that are applied independently in each brood.

3 Test Functions

In this paper, four functions are used in order to show the effectiveness of the proposed BBMO algorithm when used for global unconstrained optimization problems. The test functions used are the Rosenbrock, the Sphere, the Rastrigin and the Griewank. The Rosenbrock function is given by:

$$f(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2) \tag{4}$$

The Sphere function is given by:

$$f(x) = \sum_{i=1}^{n} x_i^2 \tag{5}$$

The Rastrigin function is given by:

$$f(x) = \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i) + 10) \tag{6}$$

The Griewank function is given by:

$$f(x) = \sum_{i=1}^{n} \frac{(x_i - 100)^2}{4000} - \prod_{i=1}^{n} cos(\frac{x_i - 100}{\sqrt(i)}) + 1 \tag{7}$$

In functions Sphere, Rastrigin and Griewank the global minima is $f(x^*) = 0$ with $x^* = (0, \cdots, 0)$, and in Rosenbrock the global minima is $f(x^*) = 0$ with $x^* = (1, \cdots, 1)$.

4 Results

The algorithm was implemented in Fortran 90 and was compiled using the Lahey f95 compiler on a Centrino Mobile Intel Pentium M750 at 1.86GHz, running Suse Linux 9.1. The parameter settings for the Bumble Bees Mating Optimization algorithm were selected after thorough empirical testing. A number of different alternative values were tested and the ones selected are those that gave the best computational results. Thus, the selected parameters are: The number of the total bees (workers - males - queens) is set equal to 100, the number of generations is set equal to 10000. Usually the total number of bees is divided in 5 queens, 45 workers and 50 males but as it is presented in Table 2 the algorithm is tested and for different number of queens. The lsi_{max} is set equal to 100, the b_{max} is set equal to 0.99, the b_{min} is set equal to 0.001 and the α is set equal to 0.8. All the algorithms used in the comparisons are population based algorithms and, thus, in order to have fair comparisons we test the algorithms using the same number of individuals (or particles for the PSO or bees for the Honey Bees Mating Optimization) and generations (or iterations). Thus, we have the same function evaluations.

In Table 2, the performance of the proposed Bumble Bees Mating Optimization algorithm in Rosenbrock function is presented. In this Table, the final cost of the Rosenbrock for six different variables ($n = 2,4,8,10,20,50$) is presented. The effectiveness of the proposed algorithm is given using different number of queens, namely, $q = 1,2,5,10,20$. As it can be observed in all cases the proposed algorithm finds the optimum when the number of variables is less or equal to 10. When the number of variables becomes equal to 20, the optimum is found with the use of 5 or 10 queens. The algorithm did not find a solution near to the optimum only in the case when we use 50 variables. The combination of parameters that gave the best results is when 5 queens are used and, thus, in all other Tables that are presented in this section, the algorithm uses 5 queens. After the selection of the final parameters, 50 different runs with the selected parameters were performed for each of the problems. The results presented in Tables are the best results found for each problem.

Table 2 Results of Bumble Bees Mating Optimization algorithm for the Rosenbrock

	n					
Queens	2	4	8	10	20	50
1	0.00	0.00	0.00	0.00	1.28E-09	31.69
2	0.00	0.00	0.00	0.00	1.93E-09	75.85
5	0.00	0.00	0.00	0.00	0.00	24.37
10	0.00	0.00	0.00	0.00	0.00	76.63
20	0.00	0.00	0.00	0.00	2.31E-09	77.94

We, also, tested the algorithm using 100000 iterations. In Table 3, the results of all the functions used are presented. As it can be observed for all test functions when the number of variables is less or equal to 20, the proposed algorithm finds the optimum. When the number of variables is equal to 50 the proposed algorithm finds the optimum in Sphere and Griewank test functions. For the Rastrigin function if the number of iterations is equal to 10000 the solution is close to the optimum and is equal to 1.59E-08, while when the number of iterations becomes equal to 100000 the optimum is found. Only in the case of the Rosenbrock function for $n = 50$ the optimum was not found but the increase of iterations to 100000 leads the algorithm to find a better solution near to the optimum.

A comparison with other population based metaheuristic approaches for the solution of the same test functions is presented in Tables 4 and 5. In these Tables, besides the proposed algorithm, five other algorithms are used for the solution of the four test functions. The algorithms are a Honey Bees Mating Optimization algorithm, a Genetic Algorithm, an Island Genetic Algorithm [9], a Differential Evolution algorithm and a Particle Swarm Optimization algorithm. In all algorithms, we used

Table 3 Results of Bumble Bees Mating Optimization algorithm for the four functions

		n					
Function	Iterations	2	4	8	10	20	50
Rosenbrock	10000	0.00	0.00	0.00	0.00	0.00	24.37
Sphere		0.00	0.00	0.00	0.00	0.00	0.00
Rastrigin		0.00	0.00	0.00	0.00	0.00	1.59E-08
Griewank		0.00	0.00	0.00	0.00	0.00	0.00
Rosenbrock	100000	0.00	0.00	0.00	0.00	0.00	1.62E-02
Sphere		0.00	0.00	0.00	0.00	0.00	0.00
Rastrigin		0.00	0.00	0.00	0.00	0.00	0.00
Griewank		0.00	0.00	0.00	0.00	0.00	0.00

the same parameters as in the previous comparisons, the same number of individuals (or particles for the PSO or bees for the HBMO) and two different number of generations (or iterations in PSO), namely 10000 (Table 4) and 100000 (Table 5). As all these algorithms have a number of different variants we use in the comparisons the variant that worked better for global unconstrained optimization problems. Thus, the selected variants for the final comparisons are, for the Differential Evolution the rand/1/bin, where "rand" corresponds to the target vector (a random target vector), "1" corresponds to the number of different vectors and "bin" corresponds to the crossover operator (for more details for the notation in differential evolution algorithms please see [25]), for the PSO the Constriction PSO [5], for the Genetic Algorithms the combination with Linear Crossover and Roulette Wheel Selection and for the Island Genetic Algorithms, the combination with Linear Crossover, Tournament selection, ten different islands and migration of the best individuals after 100 generations.

As it can be observed the proposed BBMO algorithm performs better compared to the other population based metaheuristic algorithms used in the comparisons. The BBMO algorithm performed better than the HBMO as the performance of HBMO algorithm was, in general, very good but HBMO found the optimum in less cases than the BBMO. For n=2, independently of the number of iterations, the HBMO algorithm found the optimum, for n=4, 8, 10, the optimum was not found in all cases but values near to the optimum were found (these values were even closer to the optimum or became equal to the optimum when the number of iterations was equal to 100000), for n=20, 50 the results of HBMO were less efficient than the cases where a smaller number of variables was used for all test functions but also in these cases an increase in the performance of HBMO was observed when 100000 iterations were used. It should be noted that for the case of n=50, the results of the BBMO are much better than the ones of the HBMO. The BBMO algorithm performed better than the GA as the GA found the optimum in less cases than the BBMO. In the cases where the optimum was not found by the GA, values near to the optimum were found and these values were improved when the number of iterations was equal to 100000. When the number of variables was equal to 50,

Table 4 Comparisons of the BBMO algorithm with other metaheuristics for the four functions (10000 iterations)

			n					
	Function	Iterations	2	4	8	10	20	50
BBMO	Rosenbrock	10000	0.00	0.00	0.00	0.00	0.00	24.37
	Sphere		0.00	0.00	0.00	0.00	0.00	0.00
	Rastrigin		0.00	0.00	0.00	0.00	0.00	1.59E-08
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00
HBMO	Rosenbrock		0.00	1.32E-05	8.35E-03	5.90E-02	6.38	46.07
	Sphere		0.00	0.00	0.00	0.00	1.39E-07	0.67
	Rastrigin		0.00	0.00	0.00	1.58E-09	3.93E-05	4.03
	Griewank		0.00	0.00	0.00	0.00	0.00	1.44E-02
GA	Rosenbrock		0.00	0.00	1.30	0.90	5.28	26.85
	Sphere		0.00	0.00	6.75E-08	7.31E-08	1.10E-06	9.83E-06
	Rastrigin		0.00	2.32E-07	1.33E-05	2.05E-05	2.31E-04	2.53E-03
	Griewank		0.00	0.00	0.00	0.00	4.77E-07	5.90E-06
IGA	Rosenbrock		3.21E-08	8.67E-02	4.52	16.87	74.79	461.52
	Sphere		0.00	1.02E-03	0.12	0.29	0.81	7.51
	Rastrigin		1.50E-08	0.26	7.43	14.19	58.76	287.68
	Griewank		0.00	5.13E-04	6.06E-02	0.13	0.36	0.98
DE	Rosenbrock		0.00	0.00	0.00	0.00	0.00	0.00
	Sphere		0.00	0.00	0.00	0.00	0.00	0.00
	Rastrigin		0.00	0.00	0.00	0.00	1.99	18.22
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00
PSO	Rosenbrock		0.00	0.00	4.97E-06	3.92E-09	5.62E-08	71.31
	Sphere		0.00	0.00	0.00	0.00	0.00	0.00
	Rastrigin		0.00	0.00	1.99	1.99	6.97	34.85
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00

the GA's results were inferior than the ones obtained by the GA for all the other cases. The improvement achieved to these results when the number of iterations was increased was not so significant as the one performed by the BBMO for the corresponding case. The BBMO algorithm performed better than the IGA as the IGA found the optimum in less cases than the BBMO. The IGA did not find the optimum for all test functions even when the number of variables was equal to 2. The values found by IGA were in some cases far from the optimum. However, a small improvement in the results of IGA was performed when the number of iterations was equal to 100000 but still in some cases the values found were far from the optimum. The BBMO algorithm performed slightly better than the DE. The DE gave the optimum in most of the cases, only for Rastrigin test function and for n=20 and n=50 the optimum was not found. However, when the number of iterations was increased, these values were not improved, contrary to BBMO algorithm where an increase to the number of iterations always led to an improvement of the solution. The BBMO algorithm performed better than the PSO algorithm. The PSO algorithm

Table 5 Comparisons of the BBMO algorithm with other metaheuristics for the four functions (100000 iterations)

			n					
	Function	Iterations	2	4	8	10	20	50
BBMO	Rosenbrock	100000	0.00	0.00	0.00	0.00	0.00	1.62E-02
	Sphere		0.00	0.00	0.00	0.00	0.00	0.00
	Rastrigin		0.00	0.00	0.00	0.00	0.00	0.00
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00
HBMO	Rosenbrock		0.00	5.34E-07	7.56E-03	4.80E-02	6.38	41.00
	Sphere		0.00	0.00	0.00	0.00	0.00	5.59E-02
	Rastrigin		0.00	0.00	0.00	0.00	2.02E-06	3.00
	Griewank		0.00	0.00	0.00	0.00	0.00	4.66E-03
GA	Rosenbrock		0.00	0.00	6.69E-02	8.50E-02	0.24	18.35
	Sphere		0.00	0.00	0.00	0.00	4.14E-09	7.98E-08
	Rastrigin		0.00	6.87E-09	4.47E-08	4.11E-08	7.76E-07	1.65E-05
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00
IGA	Rosenbrock		0.00	1.47E-02	3.27	9.12	70.95	418.67
	Sphere		0.00	2.61E-04	5.53E-02	0.16	0.81	7.07
	Rastrigin		0.00	7.20E-02	3.38	9.04	49.70	260.71
	Griewank		0.00	1.06E-04	2.74E-02	8.01E-02	0.34	0.97
DE	Rosenbrock		0.00	0.00	0.00	0.00	0.00	0.00
	Sphere		0.00	0.00	0.00	0.00	0.00	0.00
	Rastrigin		0.00	0.00	0.00	0.00	1.99	18.22
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00
PSO	Rosenbrock		0.00	0.00	0.00	0.00	0.00	0.11
	Sphere		0.00	0.00	0.00	0.00	0.00	0.00
	Rastrigin		0.00	0.00	1.99	1.99	6.97	34.85
	Griewank		0.00	0.00	0.00	0.00	0.00	0.00

performed efficiently but in some cases did not find the optimum and the increase in the number of iterations did not manage to improve the results of PSO in all cases, as for example for the Rastrigin test function for n=8, 10, 20, 50 the results were not improved at all.

A statistical analysis based on the Mann-Whitney U-test is presented in Table 6. In this Table, a value equal to 1 indicates a rejection of the null hypothesis at the 5% significance level, which means that the proposed method is statistically significant different from the other methods. On the other hand, a value equal to 0 indicates a failure to reject the null hypothesis at the 5% significance level, meaning that no statistical significant difference exists between the two methods. As it can be seen from this Table, the proposed method is statistically significant different from HBMO, GA, IGA and PSO in 10000 iterations, while in 100000 iterations the proposed method is statistically significant different from HBMO, GA and IGA.

Table 6 Results of Mann - Whitney test

10000 iterations						
	BBMO	HBMO	GA	IGA	DE	PSO
BBMO	-	1	1	1	0	1
HBMO	1	-	0	1	1	0
GA	1	0	-	1	1	0
IGA	1	1	1	-	1	1
DE	0	1	1	1	-	1
PSO	1	0	0	1	1	-
100000 iterations						
	BBMO	HBMO	GA	IGA	DE	PSO
BBMO	-	1	1	1	0	0
HBMO	1	-	0	1	1	0
GA	1	0	-	1	1	0
IGA	1	1	1	-	1	1
DE	0	1	1	1	-	0
PSO	0	0	0	1	0	-

5 Conclusions

In this paper, an algorithm based on the mating behavior of the bumble bees, the Bumble Bees Mating Optimization algorithm, was proposed for the solution of global unconstrained optimization problems. This algorithm was analytically presented and tested using four test functions, the Rosenbrock, the Sphere, the Rastrigin and the Griewank. The results of the algorithm were compared with the results of other popular metaheuristic and nature inspired methods, like Genetic Algorithms, Island Genetic Algorithms, Differential Evolution, Particle Swarm Optimization and the Honey Bees Mating Optimization algorithm. The results obtained showed the efficiency of the proposed algorithm and its high performance compared to the other metaheuristic algorithms.

References

[1] Abbass, H.A.: A monogenous MBO approach to satisfiability. In: International Conference on Computational Intelligence for Modelling, Control and Automation, CIMCA 2001, Las Vegas, NV, USA (2001)
[2] Abbass, H.A.: Marriage in honey-bee optimization (MBO): a haplometrosis polygynous swarming approach. In: The Congress on Evolutionary Computation (CEC 2001), Seoul, Korea, pp. 207–214 (May 2001)
[3] Afshar, A., Haddad, O.B., Marino, M.A., Adams, B.J.: Honey-bee mating optimization (HBMO) algorithm for optimal reservoir operation. J. Franklin. Inst. 344, 452–462 (2007)

[4] Baykasoglu, A., Ozbakir, L., Tapkan, P.: Artificial bee colony algorithm and its application to generalized assignment problem. In: Chan, F.T.S., Tiwari, M.K. (eds.) Swarm Intelligence, Focus on Ant and Particle Swarm Optimization, pp. 113–144. I-Tech Education and Publishing (2007)
[5] Clerc, M., Kennedy, J.: The particle swarm: explosion, stability and convergence in a multi-dimensional complex space. IEEE T. Evolut. Comput. 6, 58–73 (2002)
[6] Dasgupta, D. (ed.): Artificial immune systems and their application. Springer, Heidelberg (1998)
[7] Dorigo, M., Stützle, T.: Ant colony optimization. A Bradford Book. The MIT Press, Cambridge (2004)
[8] Drias, H., Sadeg, S., Yahi, S.: Cooperative bees swarm for solving the maximum weighted satisfiability problem. In: Cabestany, J., Prieto, A.G., Sandoval, F. (eds.) IWANN 2005. LNCS, vol. 3512, pp. 318–325. Springer, Heidelberg (2005)
[9] Engelbrecht, A.P.: Computational intelligence: An introduction, 2nd edn. John Wiley and Sons, England (2007)
[10] Fathian, M., Amiri, B., Maroosi, A.: Application of honey bee mating optimization algorithm on clustering. Appl. Math. Comput. 190, 1502–1513 (2007)
[11] Haddad, O.B., Afshar, A., Marino, M.A.: Honey-bees mating optimization (HBMO) algorithm: A new heuristic approach for water resources optimization. Water Resour. Manag. 20, 661–680 (2006)
[12] Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
[13] Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. 8, 687–697 (2008)
[14] Kennedy, J., Eberhart, R.: Particle swarm optimization. IEEE International Conference on Neural Networks 4, 1942–1948 (1995)
[15] Marinaki, M., Marinakis, Y., Zopounidis, C.: Honey bees mating optimization algorithm for financial classification problems. Appl. Soft Comput. (2009), doi: 10.1016/j.asoc.2009.09.010
[16] Marinakis, Y., Marinaki, M.: ŞA hybrid honey bees mating optimization algorithm for the probabilistic traveling salesman problem. In: IEEE Congress on Evolutionary Computation (CEC 2009), Trondheim, Norway (2009)
[17] Marinakis, Y., Marinaki, M., Dounias, G.: Honey bees mating optimization algorithm for the vehicle routing problem. In: Krasnogor, N., Nicosia, G., Pavone, M., Pelta, D. (eds.) Nature inspired cooperative strategies for optimization - NICSO 2007. Studies in Computational Intelligence, vol. 129, pp. 139–148. Springer, Berlin (2008)
[18] Marinakis, Y., Marinaki, M., Matsatsinis, N.: A hybrid clustering algorithm based on Honey Bees Mating Optimization and Greedy Randomized Adaptive Search Procedure. In: Maniezzo, V., Battiti, R., Watson, J.-P. (eds.) LION 2007 II. LNCS, vol. 5313, pp. 138–152. Springer, Heidelberg (2008)
[19] Marinakis, Y., Marinaki, M., Matsatsinis, N.: Honey bees mating optimization for the location routing problem. In: IEEE International Engineering Management Conference (IEMC – Europe 2008), Estoril, Portugal (2008)
[20] Marinakis, Y., Marinaki, M., Dounias, G.: Honey bees mating optimization algorithm for large scale vehicle routing problems. Nat. Comput. (2009), doi: 10.1007/s11047-009-9136-x

[21] Marinakis, Y., Marinaki, M., Matsatsinis, N.: A hybrid bumble bees mating optimization – GRASP algorithm for clusterin. In: Corchado, E., Wu, X., Oja, E., Herrero, Á., Baruque, B. (eds.) HAIS 2009. LNCS, vol. 5572, pp. 549–556. Springer, Heidelberg (2009)

[22] Pham, D.T., Ghanbarzadeh, A., Koc, E., Otri, S., Rahim, S., Zaidi, M.: The bees algorithm - A novel tool for complex optimization problems. In: IPROMS 2006 Proceeding 2nd International Virtual Conference on Intelligent Production Machines and Systems, Oxford. Elsevier, Amsterdam (2006)

[23] Teo, J., Abbass, H.A.: A true annealing approach to the marriage in honey bees optimization algorithm. Int. J. Comput. Intell. Appl. 3(2), 199–211 (2003)

[24] Teodorovic, D., Dell'Orco, M.: Bee colony optimization - A cooperative learning approach to complex transportation problems. Advanced OR and AI Methods in Transportation. In: Proceedings of the 16th Mini - EURO Conference and 10th Meeting of EWGT, pp. 51–60 (2005)

[25] Storn, R., Price, K.: Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. 11(4), 341–359 (1997)

[26] Wedde, H.F., Farooq, M., Zhang, Y.: BeeHive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 83–94. Springer, Heidelberg (2004)

[27] Yang, X.S.: Engineering optimizations via nature-inspired virtual bee algorithms. In: Mira, J., Álvarez, J.R. (eds.) IWINAC 2005. LNCS, vol. 3562, pp. 317–323. Springer, Heidelberg (2005)

[28] http://www.bumblebee.org

[29] http://www.everythingabout.net/articles/biology/animals/arthropods/insects/bees/bumble_bee

[30] http://bumbleboosters.unl.edu/biology.shtml

[31] http://www.colostate.edu/Depts/Entomology/courses/en570/papers_1998/walter.htm