Journal of Process Management and New Technologies
www.japmnt.com ISSN: 2334-735X (Print) ISSN: 2334-7449 (Online)
Vol. 9, Issue 3-4, 2021, pp. 89-99

SOLVING INTEGER PROGRAMMING PROBLEMS BY USING POPULATION-BASED BEETLE ANTENNAE SEARCH ALGORITHM

Ivona BRAJEVIĆ[1*], Miodrag BRZAKOVIĆ[2], Goran JOCIĆ[3]

[1] *Faculty of Applied Management, Economics and Finance, University Business Academy in Novi Sad, Jevrejska 24, 11000 Belgrade, Serbia, ivona.brajevic@mef.edu.rs*

[2] *Faculty of Applied Management, Economics and Finance, University Business Academy in Novi Sad, Jevrejska 24, 11000 Belgrade, Serbia, miodrag.brzakovic@mef.edu.rs*

[3] *Faculty of Applied Management, Economics and Finance, University Business Academy in Novi Sad, Jevrejska 24, 11000 Belgrade, Serbia, goran.jocic@mef.edu.rs*

Abstract: *Beetle antennae search (BAS) algorithm is a newly proposed single-solution based metaheuristic technique inspired by the beetle preying process. Although BAS algorithm has shown good search abilities, it can be easily trapped into local optimum when it is used to solve hard optimization problems. With the intention to overcome this drawback, this paper presents a population-based beetle antennae search (PBAS) algorithm for solving integer programming problems. This method employs the population's capability to search diverse regions of the search space to provide better guarantee for finding the optimal solution. The PBAS method was tested on nine integer programming problems and one mechanical design problem. The proposed algorithm was compared to other state-of-the-art metaheuristic techniques. The comparisons show that the proposed PBAS algorithm produces better results for majority of tested problems.*

Keywords: *stochastic optimization, integer programming problems, beetle antennae search algorithm, metaheuristic algorithms*

Original scientific paper
Received: 12.11.2021
Accepted: 08.12.2021
Available online: 08.12.2021

1. Introduction

A discrete optimization problem where all the variables are restricted to integer values is an integer programming problem. This problem can be stated as (Tawhid et al., 2019):

$$min\ f(x), \qquad x \in S \subseteq Z^n, \tag{1}$$

where S is the feasible region, while Z is the set of integers.

* Corresponding author

Note: Paper is presented at the 7th international scientific conference „Innovation as the initiator of development". Extended version of the paper is submitted to the Journal of Process Management and New Technologies.

Solving integer programming problems is a difficult task since these problems are known to be NP-hard. High computational cost *is* significant drawback of the exact optimization techniques when they are used to solve large-scale optimization problems. On the other hand, metaheuristic techniques are global optimization algorithms which can be adjusted to suit specific problem requirements. These techniques can usually reach quality solutions with less computational work.

Two important classes of metaheuristics are single-solution based and population based algorithms. *Single*-solution based metaheuristic methods, such as simulated annealing or recently proposed beetle antennae search (BAS) algorithm focus on enhancing a single potential solution. Methods which maintain and enhance multiple candidate solutions are population-based approaches. These methods use population features to lead the search and can investigate exceptionally large spaces of candidate solutions. Some of the most popular population-based metaheuristics are particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), gravitational search algorithm (GSA) (Wang et al., 2021), artificial bee colony (ABC) (Karaboga, 2005), cuckoo search (CS) (Yang, 2008) and whale optimization algorithm (WOA) (Mafarja & Mirjalili, 2017). Recently, a lot of metaheuristic algorithms are being proposed and employed to solve problems from diverse fields (Brajević & Stanimirović, 2018; Brajević & Ignjatović, 2019; Brajević et al., 2020; Brajević 2021; Du et al. 2020; Wang et al. 2020).

The BAS is a novel population-based metaheuristic method that mimics the beetle preying process. In this algorithm, when the beetle preys, two antennae of the beetle can feel the intensity of the odor released by the food (Wu et al., 2019; Khan et al., 2021). According to the diverse information of antennas, the beetle updates the flight direction and potentially reaches the food. This method is mostly applied to the optimization of single-objective problems, and it has local optimization performance (Zhang et al., 2021). However, when BAS is employed to solve multidimensional and multimodal problems, it can easily fall into a local optimum.

No single metaheuristics is suitable for solving all optimization problems as it is stated in no free lunch theorem (Brajević & Ignjatović, 2019). To find out which method has the best performance on which type of problems, metaheuristic methods have been modified to enhance their search abilities. The BAS algorithm is recently developed metaheuristic method which has shown good performance for solving certain optimization problems. To our knowledge, the BAS is not tested to solve integer programming problems. Therefore, proposing proper modifications to improve its global search capability to avoid local optimums and to solve this class of problems is a research problem. The main contribution of this study is development of a population-based beetle antennae search (PBAS) algorithm for solving integer programming problems. The PBAS employs the population's capability to search different regions of the search space. This method was tested on nine well-known integer programming problems and one mechanical design problem.

The paper is organized as follows. Section 2 presents beetle antennae search algorithm. The proposed PBAS is described in Section 3. Section 4 describes benchmark functions, while Section 5 presents analysis of the achieved results.

2. Beetle antennae search algorithm

The BAS algorithm is inspirited by searching behavior of beetles with two antennae (Jiang & Li, 2018). This algorithm considers the surrounding natural ambience as the search region. The algorithm leads the beetles forward with the odor concentration in the air. Most of beetles have two long antennae and use them as signal receivers. Basic functions of these antennae are to bind

to odor of prey. A beetle investigates neighbor region randomly employing both antennae. In addition, when the antennae in one side discover a higher concentration of odor, the beetle would spin in the direction of the same side. In other case, the beetle would focus on the other side.

This behavior could be expressed so that it is related to an objective function. The BAS algorithm employs two rules, searching behavior and detecting behavior of beetles. Since the beetle searches randomly to investigate search space, a random direction of searching is described using the following equation:

$$\vec{b} = \frac{rnd(k,1)}{\|rnd(k,1)\|} \tag{2}$$

where k is the dimensions of position and *rnd(.)* denotes a random function. Also, the searching behaviors of right-hand and left-hand sides mimic the actions of the beetle's antennae. These behaviors are described as follows:

$$\begin{aligned} x_r &= x^t + d^t\vec{b}, \\ x_l &= x^t + d^t\vec{b}, \end{aligned} \tag{3}$$

where d is the sensing length of antennae, x_r is a position in the searching region of right-hand side, while x_l denotes that of the left-hand side. This value should be large enough to exploit a suitable searching region. The following equation is used to update the beetle position:

$$x = x + \delta * \vec{b} * sign(f(x_r) - f(x_l), \tag{4}$$

where δ is the step size of searching and *sign()* is the sign function. The BAS algorithm uses the step size δ to control the convergence speed. The antennae length d and the step size δ are updated through the search as follows:

$$\begin{aligned} d^t &= 0.95d^{t-1} + 0.01, \\ \delta^t &= 0.95\delta^{t-1}, \end{aligned} \tag{5}$$

where t is the current iteration number.

Algorithm 1. Pseudo code of the BAS algorithm

```
Initialize algorithm's parameters MNI, d, δ;
Initialize solution x_i randomly in the search space and evaluate it;
t = 1;
while (t <= MNI) do
    Create the direction vector b⃗ by Eq. (2);
    Provide search with two kinds of antennae by Eq. (3);
    Update the beetle position by Eq. (4);
    Update the best solution obtained so far;
    Update parameters d and δ according to Eq. (5);
    t = t + 1;
end while
```

The main steps of this method are presented as Algorithm 1 (Jiang & Li, 2018).

3. The proposed algorithm: PBAS

In the beetle antennae search technique only one beetle is used for the optimization. The left and right antennas enable the beetle to pick up the information and to improve toward the global optimum. However, if the BAS method is employed to optimize some hard optimization problems, it will easily get stuck into the local minimum. To solve integer programming problems, the population-based BAS (PBAS) algorithm is proposed. The proposed PBAS method reaches quality solutions by iteratively choosing and integrating solutions from the population. In the optimization process, those beetles not only learn their own optimal experience, but also learn the experience of the other individuals in the population.

Algorithm 2. Pseudo code of the PBAS algorithm

Initialize algorithm's parameters *SP*, *MNI*, d, δ;
Initialize population of search solutions x_i , $i = 1, 2, \ldots, SP$ randomly in the search space;
Evaluate x_i , $i = 1, 2, \ldots, SP$;
$t = 1$;
while ($t <= MNI$) **do**
  for $i = 1$ do SP **do**
    Create a potential new solution v_i by Eq. (6) and evaluate it;
    Apply greedy selection process between x_i and v_i;
  end for
  Update the parameters of d and δ;
  Update the best solution obtained so far;
  $t = t + 1$;
end while

Motivated by the strong ability to generate solutions with plenty of diversity of mutation strategies used in the ABC (Karaboga, 2005; Zhu and Kwong, 2010), the search strategy used in the PBAS is given as follows:

$$v_{\mathrm{ij}} = \begin{cases} x_{ij} + \varphi_{ij}\left(x_{\mathrm{ij}} - x_{\mathrm{kj}}\right) + \delta * b_j * \ sign(f(x_r) \ - \ f(x_l)), & \text{if} \quad R_j \leq 0.5 \\ x_{ij} + \varphi_{ij}\left(x_{\mathrm{ij}} - x_{\mathrm{kj}}\right) + \omega_{\mathrm{ij}}\left(y_j - x_{\mathrm{kj}}\right) \ + \delta * b_j * \ sign(f(x_r) \ - \ f(x_l)), & \text{otherwise} \end{cases} \tag{6}$$

Where φ_{ij} is randomly picked real number in range (-1, 1), x_k is a randomly picked solution that is different from x_i, δ is the step size control parameter, d is the antennae length, b is direction vector, $sign()$ is the sign function, y_j is the jth element of the global best solution, w_{ij} is a uniform random number in range (0, 1.5), R_j is a randomly picked real number in range (0,1), $i \in \{1, 2, \ldots, SP\}$ and SP is the size of population, $j \in \{1, 2, \ldots, D\}$ and D is dimension of the problem. The greedy selection process between x_i and v_i is then performed to decide whether the solution will be updated. Algorithm 2 presented the pseudo code of the PBAS technique.

According to Eq. (6), if Rj is less or equal to 0.5 the first equation is performed, otherwise the second one is executed. The first equation has a good ability to investigate diverse regions of a search space due to its second term. On the other hand, the second equation integrates the information of the global best solution which enhances the ability of the algorithm to search in the regions of formerly investigated quality solutions. It follows that the exploration and exploitation abilities of the PBAS method are well balanced which is of great significance to reach

good performance. It is also important to note that the proposed approach has simple implementation.

The disadvantage of the proposed PBAS, as in many other metaheuristics, is in finding a proper combination of parameter settings. These settings have crucial influence on the algorithm performance. The parameter settings which generate quality solutions for specific problem, might not produce good solutions for other problems. A possible way for overcoming this lack is to extend the PBAS with self-adaptive control parameters.

4. Benchmark problems

To examine the performance of the PBAS method nine integer optimization problems and one mechanical design problem often used in the literature are used (Akay & Karaboga, 2009). These problems are presented as follows:

Test Problem 1. This problem is defined by:

$$F_1(x) = |x_1| + \ldots + |x_D|,$$

with $x = (x_1, x_2, \ldots, x_D)$, where D is the dimension. The solution is $x_i^* = 0$, $i=1, 2,\ldots, D$. The global minimum is $F_1(x^*) = 0$.

Test Problem 2. This problem is defined by:

$$F_2(x) = x^T x = (x_1 \ldots x_D) \begin{pmatrix} x_1 \\ \vdots \\ x_D \end{pmatrix},$$

with $x = (x_1, x_2, \ldots, x_D)$, where D is the dimension. The solution is $x_i^* = 0$, $i=1, 2,\ldots, D$. The global minimum is $F_2(x^*) = 0$.

Test Problem 3. This problem is defined by:

$$F_3(x) = -(15 \quad 27 \quad 36 \quad 18 \quad 12)x + x^T \begin{pmatrix} 35 & -20 & -10 & 32 & -10 \\ -20 & 40 & -6 & -31 & 32 \\ -10 & -6 & 11 & -6 & -10 \\ 32 & -31 & -6 & 38 & -10 \\ -10 & 32 & -10 & -20 & 31 \end{pmatrix} x.$$

The best known solutions are $x^* = (0 , 11 , 22 , 16 , 6)$ or $x^* = (0 , 12 , 23 , 17 , 6)$ and $F_3(x^*) = -737$.

Test Problem 4. This problem is defined by:

$$F_4(x) = (9x_1^2 + 2x_2^2 - 11)^2 + (3x_1^2 + 4x_2^2 - 7)^2.$$

The global minimum is $F_4(x^*) = 0$ at $x^* = (1, 1)$.

Test Problem 5. This problem is defined by:

$$F_5(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

The global minimum is $F_5(x^*) = 0$ at $x^* = (0, 0, 0, 0)$.

Test Problem 6. This problem is defined by:

$$F_6(x)= 2x_1^2+3x_2^2+4x_1x_2- 6x_1- 3x_2.$$

The global minimum is $F_6(x^*) = -6$ at $x^*= (2, -1)$.

Test Problem 7. This problem is defined by:

$$F_7(x) = -3803.84 - 138.08x_1 - 232.92x_2 + 123.08x_1^2 + 203.64\,x_2^2 + 182.25\,x_1x_2.$$

The global minimum is $F_7(x^*) = -3833.12$ at $x^*= (0, 1)$.

Test Problem 8. This problem is defined by:

$$F_8(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$$

The global minimum is $F_7(x^*) = 0$ at $x^*= (3, 2)$.

Test Problem 9. This problem is defined by:

$$F_9(x) = 100(x_2 - x_1^2)^2 + (1 - x_1^2)^2$$

The global minimum is $F_7(x^*) = 0$ at $x^*= (1, 1)$.

For each test problem the solutions were constrained in $[-100, 100]^D$, where D is the dimension of the corresponding problem. For problems F_1 and F_2, the dimension D is set to 5.

To further test the performance of the proposed algorithm, the gear train design problem is employed. The goal of this problem is to minimize the cost of the gear ratio of the gear train. The design of this problem is shown in Figure 1 (Akay & Karaboga, 2012).

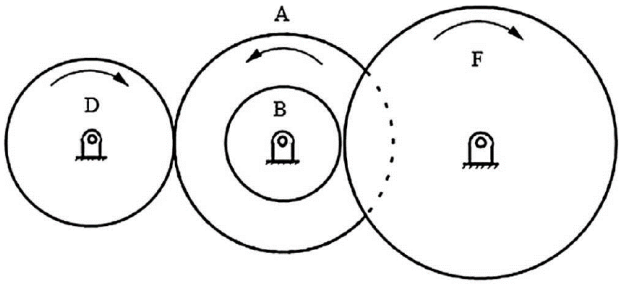

Figure 1. The gear train design

The gear ratio is defined as:

$$gear = \frac{n_B n_D}{n_F n_A}$$

Let us denote design variables of the problem n_A, n_B, n_D, n_F as x_1, x_2, x_3, x_4 respectively. Each variable must take an integer value between 12 and 60. The problem is formulated as:

$$min\, f(X) = \left(\frac{1}{6.931} - \frac{x_3x_2}{x_1x_4}\right)^2$$

The global optimum for this problem is 2. 700857E−12.

5. Experimental analysis

The proposed PBAS and ABC are implemented in Java programming language. Both algorithms were tested to solve nine integer programming benchmark problems.

Table 1. Comparative results achieved by the PSO, ABC, CS, GSA, WOA and PBAS for the F1–F9 integer programming problems (best results bold).

Prob.	Stats	PSO	ABC	CS	GSA	WOA	PBAS
F_1	Mean	20,000	**1118.4**	11,880.15	2020	18,436.36	1383.2
	Std	0.00	133.61	623.41	112.45	568.47	153.12
	SR	NA	50/50	NA	NA	NA	50/50
F_2	Mean	17,540.17	**1385.6**	7176.23	**1060**	10,134.53	1436.8
	Std	1054.56	219.78	637.75	78.69	483.25	139.73
	SR	NA	50/50	NA	NA	NA	50/50
F_3	Mean	20,000	18036.2	6400.25	**5160**	2946.63	**9678.4**
	Std	0.00	2976.58	819.94	214.25	24.25	1802.09
	SR	NA	21/50	NA	NA	NA	50/50
F_4	Mean	16,240.36	**288.05**	4920.35	1680	9255.42	412.0
	Std	1484.96	182.42	247.19	89.41	857.36	113.84
	SR	NA	50/50	NA	NA	NA	50/50
F_5	Mean	13,120.45	5409.12	7540.38	7250	6272.47	**4811.2**
	Std	1711.83	1882.33	440.82	425.36	925.35	919.91
	SR	NA	50/50	NA	NA	NA	50/50
F_6	Mean	1340.14	410.5	4875.35	1520.23	18,420.18	**392.8**
	Std	265.21	248.36	865.11	231.56	869.25	155.01
	SR	NA	50/50	NA	NA	NA	50/50
F_7	Mean	1220.46	**388.82**	3660.45	1100.24	9248.12	484.8
	Std	177.19	201.12	383.23	85.23	962.35	134.73
	SR	NA	50/50	NA	NA	NA	50/50
F_8	Mean	NA	537.02	NA	NA	NA	**485.6**
	Std	NA	331.70	NA	NA	NA	159.85
	SR	NA	50/50	NA	NA	NA	50/50
F_9	Mean	NA	1187.56	NA	NA	NA	**942.0**
	Std	NA	1490.51	NA	NA	NA	481.65
	SR	NA	50/50	NA	NA	NA	50/50

Additionally, to reveal the effectiveness of the PBAS, it is also compared to four metaheuristics that were previously applied to solve the first seven benchmarks. These metaheuristics are the basic particle swarm optimization (PSO), standard cuckoo search (CS), gravitational search algorithm (GSA) and whale optimization algorithm (WOA). The results of the PSO, CS, GSA, and WOA are taken from (Tawhid et al., 2019).

In the PBAS the *SP* is set to 20, while the maximum number of function evaluations (*MaxNFEs*) was set to 20,000. Techniques used for comparison with the PBAS also performed the same *MaxNFEs* of 20,000. In Table 1, the mean, standard deviation values and success rate values of the PSO, ABC, CS, GSA, WOA and PBAS methods are presented. From Table 1 the PBAS achieved better mean results on most test problems in comparison with its rivals. Concretely, the PBAS is better than PSO, ABC, CS, GSA and WOA in seven, five, six, five and six benchmarks, respectively. On the other hand, the PBAS is outperformed by the PSO, ABC, CS, GSA and WOA in zero, four, one, two and one test problems, respectively.

Also, the *SR* values show that the PBAS obtained a 100% success rate for each problem. Compared with ABC, the PBAS can produce better performance in terms of success rate for the benchmark F_3. For the other eight benchmarks the PBAS and ABC reached the same SR values. Figure 2 shows four representative convergence graphs obtained by the PBAS of the selected test problems.

To solve gear train problem, in the PBAS method the *SP* was set to 10, while the *MaxNFEs* was set to 5000. The statistical results of the PBAS method and five other metaheuristic techniques: unified particle swarm optimization (UPSOm) (Parsopoulos & Vrahatis, 2005), artificial bee colony (ABC) (Akay & Karaboga, 2012), improved accelerated particle swarm optimization (IAPSO) (Guedria, 2016), mine blast algorithm (MBA) (Sadollah et al., 2013) and CS (Gandomi et al., 2013) are presented in Table 2.

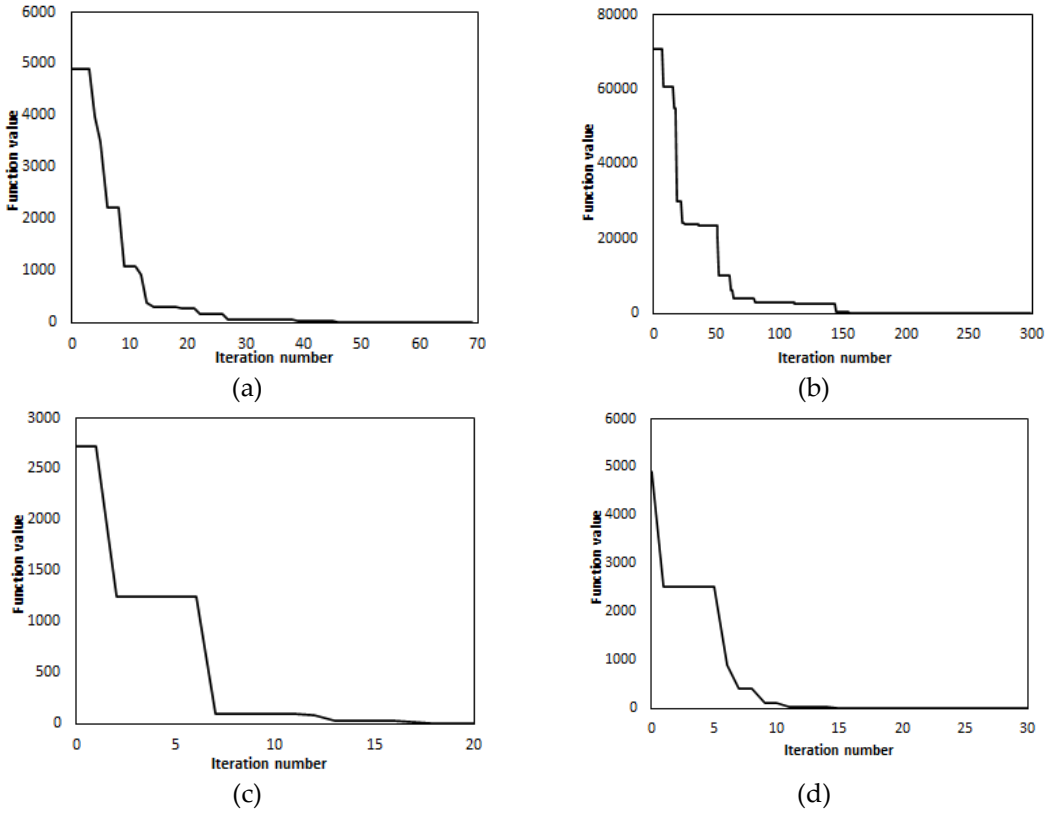

Figure 2. The convergence curve obtained by the PBAS: (a) F_2, (b) F_5, (c) F_8, (d) F_9.

Table 2. Comparative results achieved by the UPSOm, ABC, IAPSO, MBA, CS and the PBAS for gear train problem (best results bold).

Algorithm	Best	Mean	Worst	SD	NFEs
UPSOm	**2.70085E−12**	3.80562E−8	NA	1.09631E−07	100,000
ABC	**2.700857E-12**	3.641339E-10	NA	5.525811E-10	**60**
IAPSO	**2.700857E−12**	5.492477E−09	1.827380E−08	6.36E−09	800
MBA	**2.700857E−12**	2.471635E−09	2.062904E − 08	3.94E−09	1120
CS	**2.7009E−12**	1.9841E−09	2.3576E−09	3.5546−09	5000
PBAS	**2.700857E−12**	**2.986882E−10**	**1.361649E−9**	**4.2851E−10**	5000

The results of the UPSOm, ABC, IAPSO, MBA and CS were taken from their respective literature. In Table 2, the mark *NA* means that the results are not presented in the corresponding paper. The results presented in Table 2 *showed* that each method can achieve the same best solution. The reached best solution is equal to the global optimum. The PBAS has achieved the smallest mean and standard deviation values among all compared algorithms.

Conclusion

In this paper, a population-based beetle antennae search (PBAS) technique for solving integer programming problems is proposed. The PBAS approach achieves quality solutions by iteratively choosing and combining solutions from a population. The PBAS algorithm was tested on nine integer programming problems and one mechanical design problem. The proposed approach was compared to several prominent metaheuristic algorithms. These algorithms showed a good performance when they were applied to the same benchmark integer programming problems. Reached results confirmed that the PBAS performs better than the other algorithms in most benchmark problems. Compared with five other metaheuristics, the PBAS achieved competitive performance for solving the mechanical design problem. Therefore, it can be concluded that the proposed PBAS method is an encouraging approach for tackling integer programming problems.

Future work will encompass several directions. Including ideas from opposition-based learning in the proposed population-based beetle antennae search algorithm with the intention to enhance its efficiency will be investigated. To efficiently solve hard constrained optimization problems, incorporating different constraint handling methods are important problems for further investigation. Also, proposing suitable modifications of the proposed algorithm for solving more complex optimization problems should be explored.

References

Akay, B., & Karaboga, D. (2009). Solving Integer Programming Problems by Using Artificial Bee Colony Algorithm. In: Serra R., Cucchiara R. (eds) AI*IA 2009: Emergent Perspectives in Artificial Intelligence. AI*IA 2009. *Lecture Notes in Computer Science*, 5883 (pp. 355-364). Springer, Berlin, Heidelberg.

Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, 23(4), 1001–1014. https://doi.org/10.1007/s10845-010-0393-4.

Brajević I. (2021) A Shuffle-Based Artificial Bee Colony Algorithm for Solving Integer Programming and Minimax Problems. *Mathematics*, 9(11), 1211. https://doi.org/10.3390/math9111211.

Brajević, I., & Ignjatović, J. (2019). An upgraded firefly algorithm with feasibility-based rules for constrained engineering optimization problems. *Journal of Intelligent Manufacturing*, 30(6), 2545–2574. https://doi.org/10.1007/s10845-018-1419-6.

Brajević, I., & Stanimirović, P. S. (2018). An improved chaotic firefly algorithm for global numerical optimization. *International Journal of Computational Intelligence Systems*, 12(1), 131 – 148. https://doi.org/10.2991/ijcis.2018.25905187

Brajević, I., Stanimirović, P. S., Li, S., & Cao, X. (2020). A Hybrid Firefly and Multi-Strategy Artificial Bee Colony Algorithm. *International Journal of Computational Intelligence Systems*, 13(1), 810 – 821. https://doi.org/10.2991/ijcis.d.200612.001.

Du, B., He, Y., & Zhang, Y. (2020). Open-Circuit Fault Diagnosis of Three-Phase PWM Rectifier Using Beetle Antennae Search Algorithm Optimized Deep Belief Network. *Electronics*, 9, 1570. https://doi.org/10.3390/electronics9101570.

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, 29(1), 17–35. https://doi.org/10.1007/s00366-011-0241-y.

Guedria, N. B. (2016). Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Applied Soft Computing*, 40, 455–467. https://doi.org/10.1016/j.asoc.2015.10.048.

Jiang, X., & Li, S. (2018). BAS: Beetle antennae search algorithm for optimization problems. *International Journal of Robotics and Control*, 1(1) 1–5. https://doi.org/10.5430/ijrc.v1n1p1.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Kennedy, J., & Eberhart, R. (1995), Particle swarm optimization. In Proceedings of the 1995 IEEE international conference on neural networks (Perth, Australia) (pp. 1942–1948). Piscataway, NJ: IEEE Service Center.

Khan, A. T., Cao, X., Li, Z., & Li, S. (2021). Enhanced Beetle Antennae Search with Zeroing Neural Network for online solution of constrained optimization. *Neurocomputing*, 447, 294-306. https://doi.org/10.1016/j.neucom.2021.03.027.

Mafarja, M. M., & Mirjalili, S. (2017). Hybrid Whale Optimization Algorithm with simulated annealing for feature selection. *Neurocomputing*, 260, 302-312. https://doi.org/10.1016/j.neucom.2017.04.053.

Parsopoulos, K., & Vrahatis, M. (2005) Unified Particle Swarm Optimization for Solving Constrained Engineering Optimization Problems. In: Wang L., Chen K., Ong Y.S. (eds) Advances in Natural Computation. ICNC 2005. *Lecture Notes in Computer Science*, 3612 (pp. 582–591). Springer, Berlin, Heidelberg.

Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, 13(5), 2592–2612. https://doi.org/10.1016/j.asoc.2012.11.026.

Tawhid, M.A., Ali, A.F., & Tawhid, M.A. (2019). Multidirectional harmony search algorithm for solving integer programming and minimax problems. *International Journal of Bio-Inspired Computation*, 13, 141–158. https://doi.org/10.1504/IJBIC.2019.099179.

Wang, P., Gao, Y., Wu, M., Zhang, F., & Li, G. (2020), In-Field Calibration of Triaxial Accelerometer Based on Beetle Swarm Antenna Search Algorithm. *Sensors*, 20, 947. https://doi.org/10.3390/s20030947.

Wang, Y., Gao, S., Zhou, M., & Yu, Y. (2021). A Multi-Layered Gravitational Search Algorithm for Function Optimization and Real-World Problems. *IEEE/CAA Journal of Automatica Sinica*, 8 (1), 94-109. https://doi.org/10.1109/JAS.2020.1003462.

Wu, Q., Ma, Z., Xu, G., Li, S., & Chen, D. (2019). A Novel Neural Network Classifier Using Beetle Antennae Search Algorithm for Pattern Classification. *IEEE Access*, 7, 64686-64696. https://doi.org/10.1109/ACCESS.2019.2917526.

Yang, X. S. (2008). Nature-Inspired Metaheuristic Algorithms, Luniver Press.

Zhang, Y., Li, S., & Xu, B. (2021). Convergence analysis of beetle antennae search algorithm and its applications. *Soft Computing*, 25, 10595–10608. https://doi.org/10.1007/s00500-021-05991-z.

Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217, 3166–3173. https://doi.org/10.1016/j.amc.2010.08.049.