Improving Bees Algorithm Using Gradual Search Space Reduction

Turki Binbakir (✉ turkirob@gmail.com)

University of Birmingham Edgbaston campus: University of Birmingham

Research Article

Keywords: Engineering Design, Optimization, Bees Algorithm, Bracketing, Regional Elimination, Exhaustive Search, Metaheuristic, Swarm Intelligence

Posted Date: August 9th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-790818/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Improving Bees Algorithm Using Gradual Search Space Reduction

Turki Binbakir
trb519@student.bham.ac.uk

University of Birmingham, School of Mechanical Engineering

November 16, 2020

Abstract

The aim of this research is to propose a new technique to improve Bees Algorithm. Bees Algorithm is one of the well-known metaheuristic optimization method which have been subject to several attempts to improve it by overcoming some of the weaknesses. The suggested method is derived from the numerical optimization methods, namely bracketing and region elimination methods. It employs an adaption of the regional elimination method to achieve abandonment and reduction of search space within the Bees Algorithm. The utilization of the exhaustive search involves exploring the whole search space to find the optimum at equally located intervals. To assess performance, the proposed method was evaluated on twenty-four benchmark functions and two engineering problems. The acquired result indicated a statistically significant improvement.

Keywords: Engineering Design, Optimization, Bees Algorithm, Bracketing, Regional Elimination, Exhaustive Search, Metaheuristic, Swarm Intelligence

1 Introduction

Bees Algorithm (BA) belongs to relatively new category of algorithms called Swarm Intelligence (SI) algorithms that mimic the social behaviour of natural creations living together in swarms. SI algorithms are metaheuristic algorithms that aim to solve certain types of problems where it is infeasible to solve using

deterministic approach due to its high complexity. Those types of problems belong to NP-Hard problems and can be solved using SI methods which is stochastic in nature. Many of the SI methods are experiencing slow convergence rates and getting trapped in local optimum particularly for hard problems ([14] ; [3]). Moreover, due to their stochasticity, they need massive processing during their search for the optimum. The Bees algorithm is not an exception ([19]; [2]; [24]; [11]). The reminder of this paper is structured as follows: sections 2 provide an overview of Bees Algorithm (BA) and issues with BA. Section 3 is discussing the search space method and its implementation. Section 4 is designing the experiment for testing the proposed method and discussion the result. Section 5 is testing the proposed method on engineering problems.

2 Bees Algorithm

Bees Algorithm originally developed by Pham and his team in 2005 ([20]). BA is nature inspired algorithms inspired by the foraging behaviour of bees in real environment. BA starts with the initialization stage where several scout bees, randomly distributed to search for best sites. Those visited sites ranked according to their fitness. The local search commences when the best sites selected to search in their vicinity according to certain neighbourhood distance. This requires recruiting more bees for neighbourhood search around the elite site while less bees for the sites with lower quality. The global search will start when the remaining bees which have not been selected among the best sites are assigned to search randomly again as scout bees in the search area. The new population will consist of fittest bees from the local search and the global search. The searching process will continue until the stopping criteria satisfied. The stopping criteria usually defined as either the optimum solution or near the optimum is found or when a certain number of iterations reached. The population size can be calculated as follows:

$$p = ns + nre.ne + nrb.(nb - ne) \quad (1)$$

Below is a list of Bees algorithm parameters that need to be initialized by the user:

- i number of scout bees (ns),
- ii number of best sites (nb) out of sites visited by ns
- iii number of elite sites out of nb selected sites (ne)
- iv number of bees recruited for ne sites (nre),
- v number of bees recruited for the other nb - ne selected sites (nrb),
- vi size of neighbourhood (ngh),

The steps for the Bees Algorithm in its basic form:

1. Initialise the scout population with random solutions.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) //Forming the new population.
4. Select sites for the neighbourhood search.
5. Recruit bees for selected sites (more bees for fittest sites) and evaluate fitness.
6. Select the fittest bee from each patch.
7. Assign remaining bees to search randomly and evaluate their fitness.
8. End While.

2.1 Issues with Bees Algorithm

. While many successful applications of Bees Algorithm were reported, like many other populations-based metaheuristics, BA experiencing some problems due to their stochastic nature. For example, getting trapped in local optima in functions such as Rosenbrock, Langerman and Bukin6. The slow convergence to the optimum due to lack of search guidance. Although BA, performing reliably on noisy problems ([18]), its performance will degrade down on smooth unimodal functions ([18];[21]). BA greatly involves randomness, and the introduced method is aiming to minimize this randomness by gradually reducing the search space for a more focused acquisition of population.

3 Gradual Search Space Reduction Method

The proposed method in this research inspired by numerical optimization methods, namely bracketing and region elimination methods ([4]; [5]). It borrows exhaustive search technique from bracketing while employing an adapted notion of the region elimination method to achieve abandonment and reduction of search space within the Bees Algorithm as metaheuristic approach. Additionally, to make global searches more intelligent, the roles of bees for searching in the global search stage have been varied, with different swarm of bees performing their task in various parts of the search area.

3.1 Bracketing methods and Regional Elimination:

The proposed algorithm utilizes exhaustive search from bracketing method in addition to region elimination method which are both numerical optimization approaches ([4]; [5]). The exhaustive search involves calculating the optimum from samples taken within equally located intervals. This process has to take place within the defined boundaries of the search space ([16]; [5]). In the region elimination method, the core concept is to consecutively eliminates parts of the search space until the exact minimum is found. The following are the steps of this method:

1. A search space region will be specified, for example (a, b) , where $b > a$.
2. Two points will be selected, x_1 and x_2 , where $x_1 < x_2$.
3. $f(x_1)$ will be evaluated, and
 - If $f(x_1) < f(x_2)$, the minimum cannot exist beyond x_2 in the period (x_2, b) , hence the segment (x_2, b) is abandoned from the search space.
 - If $f(x_1) > f(x_2)$, the minimum cannot exist before x_1 in the period (a, x_1) . hence, the segment (a, x_1) is abandoned from the search space ([4]). See Figure 1 below

Figure 1: Bracketing-Region Elimination Method

3.2 The Bees Algorithm with Search Space Reduction (BASR)

To adapt the region elimination method, a few essential changes should be applied. The region elimination method was originally designed for optimizing unimodal

functions, whereas the proposed method should cater for multimodal problems where more than one local or global optimum might exist. Moreover, the region elimination method is a deterministic method used to find the exact minimum. Thus, given that the bees algorithm is a metaheuristic approximation method, the minimum or near minimum with a stipulated error margin will satisfy the requirements. Consequently, the following modifications to the region elimination method were introduced:

- The search space region (a, b) is considered as the whole search space specified for the domain of problem being tested, where $b > a$.
- The number of points selected should be equal to the number of the initial sample specified by parameter n of the bees algorithm ($x_1, x_2, \dots, x_n$)
- The elimination interval

$$L = \Delta x = (b - a)/n \quad (2)$$

where n is the number of initial samples. Then,

$$F(x) \quad X = (x_1, x_2, \dots, x_n) \quad (3)$$

is evaluated, and if

$$F(x) - F_{optm} > 0.001 \quad (4)$$

where $F(x)$ is the calculated optimum, F_{optm} is standard optimum for the problem being tested, 0.001 is the stipulated error margin. The segment, L will be eliminated from the search space from both ends. The new search space will be calculated as follows:

$$(b - L, a + L)$$

The process will continue until the search yield values close to the optimum within the stipulated error value where $F(x) - F_{optm} \leq 0.001$ then, the search process will be terminated. If too many unsuccessful attempts has taken place, the search area will become too small, the search will be restarted from the original search space (a, b). If the maximum number of functions evaluation is reached (500000) without finding a good solution with the stipulated error margin, the search then will be stopped.

Furthermore, to better use the global search, bees are assigned to search in different areas of the search space. There are five different suggested search scenarios:

- searching the whole search space, which decreases gradually from both ends of the search space by the elimination factor L equation (2)

- searching after first quarter ($1/4$) of the search space from the two different ends of the search space, which gradually decreases from both ends by the elimination factor L .
- searching the area between the center and the right end (b) of the search space then gradually decreased from both sides by the elimination factor L .
- searching only the area between the center and the left end (a) of the search space and gradually decreased from both sides by the elimination factor L .
- searching the whole search space, which gradually decreases from the left end (a) only by elimination factor L .

The Steps for BASR algorithm:

1. Initialize population with random solutions using search space reduction technique.
2. Evaluate fitness of the population.
3. While (stopping criterion not met) //forming new bee population
4. Select best and elite sites for neighborhood search.
5. Recruit bees around best and elite sites and evaluate fitness.
6. Select the fittest bee from each site.
7. Assign remaining bees to search randomly using search space reduction technique with 5 different scenarios and evaluate their fitness,
8. If (stopping criterion not met) reduce the search space by L
9. End while

4 Experiment Setup

To evaluate the performance of the proposed algorithm, twenty-four continuous benchmark functions used. The criteria for selecting benchmark functions for testing was designed to account for the varying degree of search space topography, complexity, and modality. This is essential to ensure testing objectivity and reliability ([10]). Complexity includes separability and dimensionality. Separability refers to the interdependency between function parameters. Non-separable functions are mostly harder to solve than separable functions ([10]). Dimensionality of a function demonstrates its number of decision variables or parameters. As the

number of dimensions in a function increases, it generally becomes harder to optimize. Modality is a property related to the number of peaks in the search space. A function is unimodal if it has only one optimum, otherwise, with many global and/or local optima, it is regarded as multimodal ([10]). Test functions used in this research are briefly displayed in the list and classified according to their properties in Table 1. Although more than half of them is two-dimensional, they are predominantly multimodal and non-separable, which adds to their difficulty.

Table 1: List of test functions and their properties

Functions	Differentiability	Separability	Scalability	Modality
Sphere (10D)	Yes	No	Yes	Unimodal
Rosenbrock (10D)	Yes	No	Yes	Unimodal
Quartic (30D)	Yes	Yes	Yes	Unimodal
Ackley (10D)	Yes	No	Yes	Multimodal
Schaffer (2D)	Yes	No	Yes	Multimodal
Easom (2D)	Yes	Yes	No	Multimodal
Rastrigin (10D)	Yes	Yes	Yes	Multimodal
Shekel (4D)	Yes	Yes	Yes	Multimodal
Langerman (10D)	Yes	No	Yes	Multimodal
Griewank (10D)	Yes	No	Yes	Multimodal
Branin (2D)	Yes	No	No	Multimodal
Sumpow (10D)	Yes	Yes	Yes	Unimodal
Bukin6 (2D)	No	No	No	Multimodal
Crossit (2D)	No	No	No	Multimodal
Drop (2D)	No	Yes	No	Multimodal
Shubert (2D)	Yes	Yes	No	Multimodal
Beale (2D)	Yes	No	No	Multimodal
McCorm (2D)	Yes	No	No	Multimodal
Camel6 (2D)	Yes	No	No	Multimodal
Boha1 (2D)	Yes	No	No	Multimodal
Colville (2D)	Yes	No	No	Multimodal
Powersum (2D)	Yes			Unimodal
Salomon (2D)	Yes	No	Yes	Multimodal
Alpine (2D)	Yes	Yes	No	Multimodal

The three key performance metrics used in this investigation are: the accuracy of fitness solution, the success rate, and the number of functions evaluation (NFE). The accuracy of fitness solution refers to how close the found optimum to the standard one. It can be described also as the quality of the solution. The success rate refers to the number of times the algorithm was able to converge to the optimum within the maximum number of function evaluations permitted in all the runs. The number of function evaluations (NFE) denotes the number of times the benchmark function being tested were executed in every individual run. It can also be described as the speed with which the algorithm converged to with the optimum. This is because the algorithm will need less time to converge as the function under testing executed fewer times. The methodology of this research

is primarily considering the quality of the solution or accuracy then to look at the success rate and the number of function evaluation (NFE). Hence, the main objective of his research should mainly contribute to the quality of the solution. The results of the test will be compared to those of the basic Bees algorithm (BBA) and two other well-known algorithms versions of PSO and ABC. For PSO, the relatively new version SPSO2011 will be used and for ABC, the quick ABC variant qABC will be used ([13]; [11]). The algorithm should continue to run until the stopping criteria defined below are met:

- either the global optimum found with acceptable error rate (ER) (chosen here as $ER \leq 0.001$) or
- maximum number of function evaluations reached (stipulated in this research as 500000).

Table 2: List of parameters values for testing BASR

No.	Functions	n	m	nsp	e	nep	ngh	stlim
1	Sphere (10D)	11	4	10	2	30	0.003	10
2	Rosenbrock (10D)	11	4	10	2	30	0.003	10
3	Quartic(30D)	13	4	10	2	30	0.005	10
4	Ackley (10D)	11	4	10	2	30	0.00001	10
5	Schaffer (2D)	11	4	10	2	30	0.01	10
6	Easom (2D)	100	4	10	2	30	0.0009	10
7	Rastrigin (10D)	11	4	10	2	30	0.003	10
8	Shekel (4D)	1000	4	10	2	30	0.001	10
9	Langerman (10D)	1000	4	10	2	30	0.09	10
10	Griewank (10D)	17	10	10	2	30	0.001	10
11	Branin (2D)	15	10	10	2	30	0.01	10
12	Sumpow (10D)	20	4	10	2	30	0.0001	10
13	Bukin6 (2D)	10	4	10	2	30	0.00001	10
14	Crossit (2D)	15	4	10	2	30	0.05	10
15	Drop (2D)	11	4	10	2	30	0.01	10
16	Shubert (2D)	1000	4	10	2	30	0.0005	10
17	Beale (2D)	45	4	10	2	30	0.045	10
18	McCorm (2D)	10	4	10	2	30	0.05	10
19	Camel6 (2D)	100	4	10	2	30	0.005	10
20	Boha1 (2D)	11	4	10	2	30	0.05	10
21	Colville (2D)	10	4	10	2	30	0.01	10
22	Powersum (2D)	10	4	10	2	30	0.01	10
23	Solomon (2D)	10	4	10	2	30	0.001	10
24	Alpine (2D)	13	4	10	2	30	0.001	10

Table 3: List of parameters values for testing BBA

No.	Functions	n	m	nsp	e	nep	ngh	stlim
1	Sphere (10D)	11	4	10	2	30	0.003	10
2	Rosenbrock (10D)	10	4	10	2	30	003	10
3	Quartic (30D)	13	4	10	2	30	005	10
4	Ackley (10D)	10	4	10	2	30	0.01	10
5	Schaffer (2D)	10	4	10	2	30	0.01	10
6	Easom (2D)	10	4	10	2	30	0.01	10
7	Rastrigin (10D)	10	4	10	2	30	0.03	10
8	Shekel (4D)	10	4	10	2	30	0.01	10
9	Langerman (10D)	10	4	10	2	30	0.09	10
10	Griewank (10D)	10	10	10	2	30	0.1	10
11	Branin (2D)	10	10	10	2	30	0.001	10
12	Sumpow (10D)	10	4	10	2	30	0.1	10
13	Bukin6 (2D)	10	4	10	2	30	0.0001	10
14	Crossit (2D)	10	4	10	2	30	0.1	10
15	Drop (2D)	10	4	10	2	30	0.01	10
16	Shubert (2D)	10	4	10	2	30	0.1	10
17	Beale (2D)	10	4	10	2	30	0.05	10
18	McCorm (2D)	10	4	10	2	30	0.5	10
19	Camel6 (2D)	10	4	10	2	30	0.05	10
20	Boha1 (2D)	10	4	10	2	30	0.05	10
21	Colville (2D)	10	4	10	2	30	0.0005	10
22	Powersum (2D)	10	4	10	2	30	0.1	10
23	Solomon (2D)	10	4	10	2	30	0.01	10
24	Alpine (2D)	13	4	10	2	30	0.001	10

Table 4: qABC Parameters Settings

Parameter	Value
Population size	10
cycles (max number)	100
Employed bees ne	5
Onlooker bees ne	4
Random scouts	1
Stagnation limit for site (abandonment stlim)	200
ra	1

Table 5: SPSO2011 Parameters Settings

Parameter	Value
Population size	10
PSO cycles (max number) T	500000
Connectivity	Default (3)
Maximum velocity	Default 1.1
C1	Default 1.1
C2	200
Wmax	Default 0.7
Wmin	Default 0.7

4.1 Results and Discussion

The assessment of the performance of the proposed algorithm BASR involves the comparison with other three algorithms namely: BBA, qABC and SPSO2011. The parameters setting for these algorithms are shown above in Tables 2, 3, 4 and 5. The analysis of performance according to figures in Table 6 are showing that the proposed algorithm BASR performed better than all other algorithms in at least 20 function. Mean and standard deviation figures in Table 7 confirms the conclusion about the performance data with the proposed algorithm achieved better mean in 19 function and better standard deviation in 17 function. However, in two functions, BASR reached the maximum NFE without finding the optimum within the stipulated error margin. Additionally, in Shekel function, BASR achieved better mean value even though qABC performed better and achieved more accurate optimum than BASR. Despite the above consideration, BASR performance remain remarkable considering the performance in the functions with complex topography and multi pocketed landscape such as: Rosenbrock, Ackley, Shaffer, Rastrigin, Griewank, and Drop where BASR getting better optimum in all the 50 runs. In order to assess the statistical significance of the performance of BASR, it requires the calculation of the p-value using Mann-Whitney significance test. The inspection of Table 8 reveals that the figures confirms the good performance by BASR with better p-value in at least 18 functions which comprise three-quarter of the functions. These findings support the conclusion that gradual search space reduction technique used in the proposed algorithm BASR, helped in improving the performance of Bees Algorithm.

Table 6: Accuracy Figures for BASR, BBA, qABC and SPSO2011

No.	Functions	Result of 50 Runs		Result of 50 Runs		Result of 50 Runs	
		BASR	BBA	BASR	SPSO2011	BASR	qABC
1	Sphere (10D)	50	0	50	0	50	0
2	Rosenbrock (10D)	50	0	41	9	50	0
3	Quartic (30D)	50	0	49	1	50	0
4	Ackley (10D)	50	0	50	0	50	0
5	Schaffer (2D)	50	0	50	0	50	0
6	Easom (2D)	32	18	50	0	19	31
7	Rastrigin (10D)	50	0	50	0	50	0
8	Shekel (4D)	34	16	50	0	20	30
9	Langerman (10D)	27	23	50	0	36	14
10	Griewank (10D)	50	0	50	0	50	0
11	Branin (2D)	39	11	0	50	19	31
12	Sumpow (10D)	45	5	38	12	37	13
13	Bukin6 (2D)	38	12	48	2	42	8
14	Crossit (2D)	28	22	50	0	31	19
15	Drop (2D)	50	0	50	0	50	0
16	Shubert (2D)	44	6	50	0	14	36
17	Beale (2D)	33	17	28	22	34	16
18	McCorm (2D)	29	21	50	0	28	22
19	Camel6 (2D)	38	12	50	0	28	22
20	Boha1 (2D)	50	0	50	0	50	0
21	Colville (2D)	38	12	25	25	47	3
22	Powersum (2D)	28	22	39	11	48	2
23	Solomon (2D)	50	0	50	0	50	0
24	Alpine (2D)	50	0	50	0	50	0
Total:		BASR=24 BBA=0		BASR=22 None=1 SPSO2011=1		BASR=20 qABC=4	

Table 7: Accuracy Mean and Standard Deviation Figures for BASR, BBA, qABC and SPSO2011

No.	Functions	BBA		qABC		SPSO2011		BASR	
		Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.
1	Sphere (10D)	8.19E-04	1.27E-04	6.24E-04	2.49E-04	7.98E-04	1.44E-04	4.1E-32	1.31E-32
2	Rosenbrock (10D)	1.8E+00	9.23E-01	4.09E-01	8.85E-01	8.88E+00	4.36E+01	9.55E-04	3.72E-05
3	Quartic (30D)	3.7E+00	7.34E-01	5.23E-01	2.88E-01	1.18E-01	6.7E-02	4.18E-03	2.14E-02
4	Ackley (10D)	1.38E+01	1.25E+00	3.11E-03	3.05E-03	3.7E-01	6.35E-01	1.42E-14	2.25E-15
5	Schaffer (2D)	2.57E-04	2.61E-04	9.65E-04	8.04E-04	2.69E-04	2.6E-04	0.0E+00	0.0E+00
6	Easom (2D)	4.56E-04	2.64E-04	2.73E-04	3.28E-04	1.0E+00	2.74E-06	3.41E-04	2.29E-04
7	Rastrigin (10D)	1.33E+01	3.84E+00	6.22E-04	2.78E-04	1.93E+01	1.01E+01	0.0E+00	0.0E+00
8	Shekel (4D)	6.67E-04	1.96E-04	2.34E-02	1.61E-01	9.32E+00	3.44E-01	5.01E-04	2.26E-04
9	Langerman (10D)	2.74E-01	1.79E-01	3.5E-01	1.63E-01	7.06E-01	9.64E-05	3.21E-01	1.5E-01
10	Griewank (10D)	5.68E-01	7.78E-02	4.65E-02	2.19E-02	1.38E-01	8.28E-02	0.0E+00	0.0E+00
11	Branin (2D)	6.43E-04	1.96E-04	2.63E-04	2.76E-04	3.58E-07	0.0E+00	3.85E-04	2.95E-04
12	Sumpow (10D)	6.86E-04	2.44E-04	3.82E-04	1.12E-04	5.48E-04	2.73E-04	2.17E-04	3.05E-04
13	Bukin6 (2D)	1.59E-02	4.93E-03	2.81E-02	1.61E-02	7.35E-02	3.97E-02	1.02E-02	4.07E-03
14	Crossit (2D)	4.23E-04	2.71E-04	5.65E-04	3.01E-04	4.43E-02	4.49E-02	3.77E-04	2.57E-04
15	Drop (2D)	4.06E-04	2.48E-04	1.12E-02	2.17E-02	2.25E-01	1.14E-01	0.0E+00	0.0E+00
16	Shubert (2D)	5.24E-03	4.89E-03	2.69E-04	3.13E-04	7.88E+01	4.48E+01	4.93E-04	2.6E-04
17	Beale (2D)	4.82E-04	2.74E-04	1.98E-03	3.49E-03	1.58E-02	1.07E-01	4.21E-04	2.72E-04
18	McCorm (2D)	5.26E-04	2.92E-04	5.06E-04	2.83E-04	1.58E-01	1.32E-01	4.68E-04	2.6E-04
19	Camel6 (2D)	4.87E-04	2.86E-04	4.01E-04	3.26E-04	4.44E-01	2.85E-01	2.63E-04	2.94E-04
20	Boha1 (2D)	4.68E-04	2.92E-04	3.2E-04	2.78E-04	4.12E-04	2.65E-04	0.0E+00	0.0E+00
21	Colville (2D)	9.28E-04	7.65E-05	3.43E-02	3.1E-02	8.08E-04	1.76E-04	8.45E-04	1.45E-04
22	Powersum (4D)	6.84E-04	2.26E-04	7.13E-03	8.58E-03	8.31E-04	2.21E-04	6.42E-04	2.1E-04
23	Solomon (2D)	5.49E-04	2.57E-04	4.8E-02	4.88E-02	2.64E-03	1.39E-02	1.06E-15	3.07E-16
24	Alpine (2D)	6.25E-04	2.41E-04	5.58E-04	2.84E-04	6.69E-04	2.35E-04	1.07E-16	3.93E-17

Table 8: P-values using Mann-Whitney test ($\alpha = 0.05$) for accuracy acquired by BASR over ABC, BBA, and PSO

No.	Functions	BASR-qABC		BASR-SPSO2011		BASR-BBA	
		p value	Significant	p value	Significant	p value	Significant
1	Sphere (10D)	7.07E-18	Yes	7.07E-18	Yes	7.07E-18	Yes
2	Rosenbrock (10D)	7.07E-18	Yes	7.75E-09	Yes	7.07E-18	Yes
3	Quartic (30D)	1.08E-17	Yes	6.34E-17	Yes	7.07E-18	Yes
4	Ackley (10D)	1.86E-18	Yes	1.82E-18	Yes	1.86E-18	Yes
5	Schaffer (2D)	3.31E-20	Yes	3.31E-20	Yes	3.31E-20	Yes
6	Easom (2D)	7.40E-03	qABC	9.12E-20	Yes	2.40E-02	Yes
7	Rastrigin (10D)	3.31E-20	Yes	3.29E-20	Yes	3.31E-20	Yes
8	Shekel (4D)	1.43E-01	No	7.07E-18	Yes	2.92E-04	Yes
9	Langerman (10D)	4.22E-04	Yes	7.07E-18	Yes	6.97E-01	No
10	Griewank (10D)	3.31E-20	Yes	3.31E-20	Yes	3.31E-20	Yes
11	Branin (2D)	1.43E-02	qABC	3.31E-20	PSO	8.87E-06	Yes
12	Sumpow (10D)	3.95E-05	Yes	1.58E-07	Yes	4.71E-10	Yes
13	Bukin6 (2D)	5.89E-08	Yes	5.85E-15	Yes	1.84E-07	Yes
14	Crossit (2D)	2.13E-03	Yes	7.07E-18	Yes	4.18E-01	No
15	Drop (2D)	3.31E-20	Yes	3.31E-20	Yes	3.31E-20	Yes
16	Shubert (2D)	2.99E-05	qABC	7.07E-18	Yes	8.11E-13	Yes
17	Beale (2D)	1.25E-04	Yes	6.03E-02	No	2.43E-01	No
18	McCorm (2D)	5.74E-01	No	7.07E-18	Yes	3.26E-01	No
19	Camel6 (2D)	6.72E-02	No	7.07E-18	Yes	9.14E-05	Yes
20	Boha1 (2D)	3.31E-20	Yes	3.31E-20	Yes	3.31E-20	Yes
21	Colville (2D)	1.35E-13	Yes	4.18E-01	No	3.90E-04	Yes
22	Powersum (2D)	2.03E-14	Yes	5.90E-07	Yes	2.63E-01	No
23	Solomon (4D)	6.06E-18	Yes	7.07E-18	Yes	7.07E-18	Yes
24	Alpine (2D)	7.07E-18	Yes	7.07E-18	Yes	7.07E-18	Yes
Total:		BASR:18 None:3	qABC:3	BASR:21 SPSO2011:1 None:2		BASR:19 None: 5	BBA:0

4.2 Success Rate and Number of Function Evaluation

The examination of Table 9, indicate that the proposed algorithm BASR, achieved the highest SR average among all the algorithms. It was able to achieve %100 SR in all functions except in three functions. In Table 10, another aspect of improved performance by BASR against qABC and BBA is indicated. Nonetheless, the performance of BASR against SPSO2011 reveals less improved performance by BASR compared to other algorithms. Although the figures show lower NFE figures by SPSO2011 which suggest a better performance, it cannot be considered an indication of good performance due to the fact that SPSO2011 ended the search earlier even though its obtained optimum does not fall within the stipulated error margin of 0.001. In fact, for the functions (Easom, Shekel, Langerman, Crossit, Drop, Shubert, and McCorm) the optimum was located far away from the error margin. This phenomenon referred to as premature convergence. This is also reflected in the mean and standard deviation figures in Table 11. However, the above findings yet to be confirmed by the statistical significance test using Mann-Whitney test.

Table 9: Success rate figures for the BASR, BBA, qABC and SPSO2011 algorithms

No.	Functions	qABC SR	SPSO2011 SR	BBA SR	BASR SR
1	Sphere (10D)	100%	100%	100%	100%
2	Rosenbrock (10D)	0%	74%	0%	100%
3	Quartic	38%	0%	0%	96%
4	Ackley (10D)	24%	74%	0%	100%
5	Schaffer (2D)	66%	100%	100%	100%
6	Easom (2D)	100%	100%	100%	100%
7	Rastrigin (10D)	98%	0%	0%	100%
8	Shekel (4D)	98%	100%	100%	100%
9	Langerman (10D)	4%	100%	30%	18%
10	Griewank (10D)	0%	0%	0%	100%
11	Branin (2D)	100%	0%	100%	100%
12	Sumpow (10D)	100%	100%	100%	100%
13	Bukin6 (2D)	6%	4%	0%	0%
14	Crossit (2D)	100%	100%	100%	100%
15	Drop (2D)	74%	100%	100%	100%
17	Beale (2D)	76%	98%	100%	100%
18	McCorm (2D)	100%	100%	100%	100%
19	Camel6 (2D)	100%	100%	100%	100%
20	Bohal (2D)	100%	100%	100%	100%
21	Colville (2D)	12%	100%	100%	100%
22	Powersum (2D)	18%	100%	100%	100%
23	Solomon (4D)	48%	96%	100%	100%
24	Alpine	100%	100%	100%	100%
SR Average:		65%	77%	69%	92%

Table 10: NFE figures for BASR, BBA, qABC and SPSO2011 algorithms

No.	Functions	Result of 50 Runs		Result of 50 Runs		Result of 50 Runs	
		BASR	qABC	BASR	SPSO2011	BASR	BBA
1	Sphere (10 D)	50	0	50	0	50	0
2	Rosenbrock (10D)	50	0	44	6	50	0
3	Quartic (30D)	40	10	48	2	50	0
4	Ackley (10D)	50	0	49	0	50	0
5	Schaffer (2D)	50	0	50	0	50	0
6	Easom (2D)	19	31	0	50	44	6
7	Rastrigin (10D)	50	0	50	0	50	0
8	Shekel (4D)	7	43	0	50	25	25
9	Langerman (10D)	9	41	0	50	8	42
10	Griewank (10D)	50	0	50	0	50	0
11	Branin (2D)	30	20	50	0	49	1
12	Sumpow (10D)	49	1	50	0	50	0
13	Bukin6 (2D)	26	24	0	50	3	46
14	Crossit (2D)	39	11	2	48	37	4
15	Drop (2D)	50	0	0	50	50	0
16	Shubert (2D)	8	42	0	50	50	0
17	Beale (2D)	47	3	44	6	12	33
18	McCorm (2D)	46	4	0	50	35	12
19	Camel6 (2D)	17	33	0	50	23	19
20	Boha1 (2D)	50	0	50	0	50	0
21	Colville (2D)	46	4	6	44	50	0
22	Powersum (2D)	50	0	38	12	45	5
23	Solomon (4D)	50	0	50	0	50	0
24	Alpine	50	0	50	0	50	0
Total:		BASR:19		BASR:14		BASR:20	
		qABC:5		SPSO2011:10		BBA:3 None:1	

Table 11: Mean and Std.Dev. figures for BASR, BBA, qABC and SPSO2011

No.	Functions	BBA		qABC		SPSO2011		BASR	
		Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std.Dev.	Mean	Std
1	Sphere (10D)	9180.34	4550.16	33446.14	9643.9	5884	437.89	102	0
2	Rosenbrock (10D)	500042.2	223747	500048.9	25.9	209694	190711.6	55705.44	2036.26
3	Quartic (30)	500068	225933.7	377869.9	180710.9	500000	0	160903.4	151059.8
4	Ackley (10D)	500039	235690.6	468309.6	70444.77	139692	213573	102	0
5	Schaffer (2D)	5066.22	2820.48	233117.2	235650	2122	544.53	102	0
6	Easom (2D)	4060.24	1890.02	2650.48	4249.69	100	0	1972.66	814.79
7	Rastrigin (10D)	500068.9	235704.7	180555.7	92942.62	500000	0	102	0
8	Shekel (4D)	62742.04	33644.05	43292.62	96619.86	100	0	51442.88	11641.98
9	Langerman (10D)	385919.2	237557	491350.5	60942.64	100	0	431486.8	150993.1
11	Griewank (10D)	500045.5	235691.8	500049.6	27.24	500000	0	110	0
10	Branin (2D)	4115.46	2048.44	1052.04	885.35	500000	0	857.66	388.14
12	Sumpow (10D)	951.2	438.41	150	0	1502	327.1	75.68	47.38
13	Bukin6 (2D)	500012.5	235700.7	490635.5	57616.16	490634	65562	500031.9	14
14	Crossit (2D)	400.84	207.09	274	115.86	100	0	164.24	69.92
15	Drop (2D)	27392.88	21770.94	236593.4	196388.3	100	0	102	0
16	Shubert (2D)	460845.8	223514.6	5662.18	9603.01	100	0	10468.8	3658.4
17	Beale (2D)	654.84	468.63	180170.2	213103.4	11656	69765.16	954.96	349.25
18	McCorm (2D)	1319.68	690.69	13572.06	17897.62	100	0	598.82	332.08
19	Camel6 (2D)	1025.76	540.44	790	369.86	120	44.72	874.4	300.75
20	Bohal (2D)	93133.46	69487.08	1872.26	1145.41	4992	742.12	102	0
21	Colville (2D)	90739.72	43316.04	453006.8	141528.2	14160	4506	31410.62	17993.65
22	Powersum (2D)	74731.52	50147.5	463048.3	94270.83	23094	23524.03	11361.32	8803.96
23	Solomon (2D)	38934.28	29375.34	273926.7	238028.5	26946	96573.06	102	0
24	Alpine (2D)	3546.6	1893.11	2344	8793.92	2912	1601.83	102	0

The p-value statistical figures in Table 12 for BASR against BBA and qABC confirming that BASR improved significantly in terms of NFE. BASR was better than BBA in 19 test functions and better than qABC in 17 functions. This means that BASR performed faster than these two algorithms. With regards to SPSO2011, the p-value figures were lower suggesting less significant performance by BASR; however, as discussed earlier, this cannot be considered an indication of good performance by SPSO2011. It is an aspect of the premature convergence phenomenon where algorithms abruptly stop processing without finding any good solution.

Table 12: P-values using Mann-Whitney test ($\alpha = 0.05$) for NFE acquired by BASR over qABC, BBA, and SPSO2011

No.	Functions	BASR -qABC		BASR -SPSO2011		BASR -BBA	
		p value	Significant	p value	Significant	p value	Significant
1	Sphere (10D)	3.31E-20	Yes	1.72E-20	Yes	3.28E-20	Yes
2	Rosenbrock (10D)	6.84E-18	Yes	3.05E-11	Yes	6.69E-18	Yes
3	Quartic (30 D)	4.58E-08	Yes	1.73E-17	Yes	6.21E-18	Yes
4	Ackley (10D)	3.30E-20	Yes	1.59E-20	Yes	3.11E-20	Yes
5	Schaffer (2D)	3.30E-20	Yes	1.72E-20	Yes	3.30E-20	Yes
6	Easom (2D)	1.21E-02	qABC	2.50E-20	PSO	2.11E-13	Yes
7	Rastrigin (10D)	3.31E-20	Yes	1.59E-23	Yes	3.16E-20	Yes
8	Shekel (4D)	2.10E-10	qABC	2.55E-20	PSO	0.297714	No
9	Langerman (10D)	2.65E-08	qABC	2.46E-20	PSO	1.46E-09	BBA
11	Griewank (10D)	3.30E-20	Yes	1.59E-23	Yes	3.04E-20	Yes
10	Branin (2D)	9.96E-02	No	2.44E-20	Yes	3.41E-16	Yes
12	Sumpow (10D)	5.98E-21	Yes	1.24E-18	Yes	1.69E-18	Yes
13	Bukin6 (2D)	2.17E-01	No	3.63E-20	PSO	3.88E-13	BBA
14	Crossit (2D)	1.42E-07	Yes	8.39E-18	PSO	5.54E-09	Yes
15	Drop (2D)	3.31E-20	Yes	1.59E-23	PSO	3.31E-20	Yes
16	Shubert (2D)	7.67E-10	qABC	2.38E-20	PSO	6.34E-18	Yes
17	Beale (2D)	1.36E-14	Yes	3.55E-12	Yes	3.23E-05	BBA
18	McCorm (2D)	1.14E-12	Yes	2.32E-20	PSO	4.92E-08	Yes
19	Camel6 (2D)	2.02E-02	qABC	2.86E-19	PSO	0.1711787	No
20	Boha1 (2D)	3.19E-20	Yes	1.74E-20	Yes	3.31E-20	Yes
21	Colville (2D)	7.63E-14	Yes	2.96E-12	PSO	3.77E-15	Yes
22	Powersum (2D)	7.06E-18	Yes	1.07E-03	Yes	8.59E-12	Yes
23	Solomon (2D)	3.30E-20	Yes	1.78E-20	Yes	3.31E-20	Yes
24	Alpine (2D)	3.11E-20	Yes	1.76E-20	Yes	3.28E-20	Yes
Total:		BASR= 17 qABC= 5		BASR= 14 SPSO2011=10		BASR= 19 BBA= 2 None= 3	

5 Testing with Engineering Problems

To further evaluate the performance of the proposed algorithm BASR, tow engineering problems where selected to apply the proposed algorithm. The two engineering problems are gear train problem, a none-constraint problem and tension/compression spring, a constrained problem.

5.0.1 Gear Train Problem

The gear train design problem is an engineering problem aiming to minimize the gear ratio to be particularly close to $1/6.931$ ([17]). The design variables here represent the number of teeth for every gear, restricted to values between 12 and 60.

Figure 2: gear train design scheme

To reduce the error value, the gear ratio should be as close as possible to $1/6.931$ ([12]). To calculate the error value:

$$Error\ value(\%) = \frac{Gear\ Ratio - (\frac{1}{6.931})}{(\frac{1}{6.931})} \times 100\% \quad (5)$$

$$Gear\ Ratio = \frac{x_3 x_2}{x_1 x_4} \quad (6)$$

The objective function formulated as:

$$\min_X f(X) = \left(\frac{1}{6.931} - \frac{x_3 x_2}{x_1 x_4} \right)^2 \quad (7)$$

The result of the experiment was taken out of 100 runs. The selection of parameter values for the proposed algorithms, was partially decided from the literature review of other algorithms based on BA and partially through trial and error. To evaluate the performance of the proposed algorithm, their performances were first compared with the same algorithms used to assess the performance on benchmark functions namely: BBA, qABC and SPSO2011 with the same parameter settings. The figures for BBA, SPSO2011 and qABC were obtained experimentally. The algorithm was forced to conduct a minimum of 100 iterations to acquire a good result. The stopping criteria used are the same criteria used to test the algorithm with benchmark functions, or 500000 NFE. Furthermore, the results obtained by the test were compared with some of the well performing algorithms on this problem obtained from the literature such as: ABC ([1]), CS ([6]) and PSO-GA ([7]).

Table 13: BASR Parameter Setting for Gear Train Problem

Parameter	Parameter Value
Number of scout bees, n	20
Number of selected bees, m	4
Number of elite bees, e	2
Number of forager bees for elite bee, nep	30
Number of foragers for selected bee, nsp	10
Neighborhood size, ngh	0.1

Table 14 Performance figures for Gear train problem for BASR compared with qABC, SPSO2011 and BBA

Item	qABC	SPSO2011	BBA	BASR
Mean	0.0001642317	0.0001754215	1.580352 E-05	6.12116 E-05
Std. Dev	0.0002518493	0.000262571	1.874908 E-05	7.858313 E-05
Best Solution	2.357641 E-09	1.117291 E-08	9.745653 E-10	2.700857 E-12
X1	48	51	47	49
X2	12	15	24	19
X3	30	27	13	16
X4	52	55	46	43
Gear Ratio	0.1442308	0.144385	0.1443108	0.144281
Error %	0.004%	0.01%	0.003%	0.001%
Evaluations	150	100	14526	14555

According to Table 14, the figures show that the proposed algorithm performed better than all the other algorithms in terms of best value achieved. Additionally, the statistical indicators suggest that BASR and BBA behaved in a more stable and

Table 15 Performance figures for Gear train problem of BASR compared with ABC, PSO-GA and CS

Item	ABC	PSO-GA	CS	BASR
Mean	3.641339 E-10	1.285602 E-10	1.9841E-9	6.12116 E-05
Std. Dev	5.525811 E-10	8.7787 E-10	3.5546 E-9	7.858313 E-05
Best Solution	2.700857 E-12	2.70085 E-12	2.7009 E-12	2.700857 E-12
X1	49	43	43	49
X2	16	16	16	19
X3	19	19	19	16
X4	43	49	49	43
Gear Ratio				
Error	0.144281	0.14428096	0.144281	0.144281
Evaluations	0.001%	0.001%	0.001%	0.001%
	60	NA	500	1455

consistent manner than the other two. However, the comparison with the second group of algorithms: PSO-GA, BASR, ABC, and CS in 15, indicate that the proposed algorithm performed competitively with the other algorithms in terms of best value; obtaining a gear ratio of 0.14428 and an error value of 0.001%. However, the mean and standard deviation suggested that BASR was not as stable and consistent as other algorithms. With regards to the speed,, ABC was the fastest as it needed only 60 NFE to achieve its best result, which is the lowest among all. BASR was the slowest among this group with 14555 NFE while CS comes in the middle with 5000 NFE.

5.0.2 The Tension/Compression Spring

The aim of tension/compression spring problem is to reduce the spring steel wire volume, which means decreasing the weight of the spring ([8]).

Figure 3: Tension/Compression Spring

The objective function of tension/compression problem is:

$$\min_X f(X) = (x_3 + 2)x_2x_1^2$$

parameter values restricted to:

$$0.05 \leq x_1 \leq 2.0$$

$$0.25 \leq x_2 \leq 1.3$$

$$2.0 \leq x_3 \leq 15.0$$

Subject to the following constraints:

$$g_1(x) = 1 - \frac{(x_2^3 x_3)}{(71785 x_1^4)} \leq 0$$

$$g_2(x) = \frac{(4x_2^2 - x_1 x_2)}{12566 x_2 x_1^3 - x_1^4} + \frac{1}{(5108 x_1^2)} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{(140.45 x_1)}{(x_2^2 x_3)} \leq 0$$

$$g_4(x) = (x_2 + x_1)/(1.5) - 1 \leq 0$$

Engineering design problems are predominantly complex and nonlinear with many different variables and constraints. These variables and constraints are required to be satisfied in order to achieve an optimized solution ([6]). To ensure the variables restricted to the domain of the values, a method implemented to check that the variables values are within the limit of the domain of value and to regenerated them if they were located outside. With regards to the problem

constraints, only the feasible solutions that satisfy all the constraints will be considered. 100 runs were performed, and every run was forced to achieve at least 100 iterations. To evaluate the performance of the three proposed algorithms, their performances were first compared with the same algorithms used to assess the performance on benchmark functions: BBA, qABC and SPSO2011 using the same parameter settings. The figures for BBA and qABC were obtained experimentally; while SPSO2011 figures were obtained from the literature of applying SPSO2011 on this problem ([23]) due to inability to modify the algorithm code to test the problem constraints. Additionally, to further assess the performance, a collection of algorithms which were applied successfully as indicated by the literature used to compare performance. They are the following: evolutionary strategies ([15]), PSO ([9]), ABC ([1]), society and civilization algorithm (SCA) ([22]) and unified particle swarm Optimisation (UPSOM) ([17]). Table 16 shows the parameters used to apply the proposed algorithms on the tension/compression spring.

Table 16 BASR parameter values for tension/compression spring

Parameter	Parameter Value
Number of scout bees, n	100
Number of selected bees, m	4
Number of elite bees, e	2
Number of forager bees for elite bee, nep	30
Number of foragers for selected bee, nsp	10
Neighborhood size, ngh	0.0001

The inspection of the figures for Tables 17 and 18, indicates that BBA achieved the best value among all; next comes the proposed algorithm BASR. This also true in terms of NFE or speed, with BBA and BASR achieving 8099,17889 respectively. However, in terms of mean and standard deviation, both BBA and BASR were in the middle, suggesting that their performance was moderately stable and consistent. Although BASR could not improve over BBA in this engineering problem, but the result indicates that the method used has the potential to achieve good performance.

Table 17: Comparison of the figures of the tension/compression spring problems for qABC, SPSO2011, BBA, and BASR algorithms involved in the testing

Tension/Compression Spring	qABC	SPSO2011	BBA	BASR
Best	0.01290396	0.01266856	0.0061625	0.008838
Mean	0.067087	0.01274005	0.01378989	0.012702
Std. Dev.	0.044828	0.000210	0.01302519	0.012017
Evaluations	23661	NA	8099	17889

Table 18: Comparison of the figures of the tension/compression spring problems with ABC, SCA, $(\mu + \lambda)$ -ES, UPSOm, PSO, and BASR algorithms involved in the testing

Tension/Compression Spring	ABC	SCA	$(\mu + \lambda)$ -ES	UPSOm	PSO	BASR
Best	0.012665	0.012669	0.012689	0.01312	0.012665	0.008838
Mean	0.012709	0.012923	0.013165	0.0229478	0.012702	0.035154
Std. Dev.	0.0128	0.00059	0.00039	0.0072	4.1E-05	0.012017
Evaluations	30,000	25167	30000	100000	15000	17889

Table 19 List of variable and constraints values achieved by BASR

Variables and Constraints Values	BASR
x1	0.0685
x2	0.5667
x3	12.204
g1	-0.4
g2	-0.341
g3	-1.456
g4	-0.576

6 Conclusion Future Work

This research presented a new method BASR to improve the performance of Bees Algorithm. It used an adaptation of two numerical method namely: Bracketing-exhaustive search and region elimination methods to fit in metaheuristic approach. It was tested on twenty-four benchmark functions and two engineering problems. Its performance on benchmarks was apparently accurate and fast with an acceptable level of consistency. However, it performed moderately well on engineering problem with moderate level of consistency. The analysis of the obtained result indicated that the search space reduction has the potential to improve the BA performance on optimization of problems with continuous search space due to its ability to achieve focused search. However, this suggests the need for further improvement of the articulated method to improve on engineering problems.

References

- [1] B. Akay and D. Karaboga. “Artificial bee colony algorithm for large-scale problems and engineering design optimization”. In: *Journal of intelligent manufacturing* 23.4 (2012), pp.1001–1014.
- [2] A. Alfi and A. Khosravi. “Constrained Nonlinear Optimal Control via a Hybrid BA-SD”. In: *International Journal of Engineering-Transactions C: Aspects* 25.3 (2012), pp.197–204.
- [3] Z. Beheshti and S. M. H. Shamsuddin. “A review of population-based meta-heuristic algorithms”. In: *Int. J. Adv. Soft Comput. Appl* 5.1 (2013), pp.1–35.
- [4] R. K. Bhattacharjya. “Region Elimination Method”. In: *CE 602: Optimization Method [Lecture notes]*. Guwahati: Indian Institute of Technology, 2009. URL: <http://www.iitg.ac.in/rkbc/CE602/ce602.htm>.
- [5] K. Deb. *Optimization for engineering design: Algorithms and examples*. New Delhi, India: PHI Learning Pvt. Ltd, 2012.
- [6] A. H. Gandomi, X. S. Yang, and A. H. Alavi. “Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems”. In: *Engineering with computers* 29.1 (2013), pp.17–35.
- [7] H. Garg. “A hybrid PSO-GA algorithm for constrained optimization problems”. In: *Applied Mathematics and Computation* 274 (2016), pp.292–305.
- [8] C. X. Guo, J. S. Hu, B. Ye, and Y. J. Cao. “Swarm intelligence for mixed-variable design optimization”. In: *Journal of Zhejiang University-SCIENCE A* 5.7 (2004), pp.851–860.
- [9] S. He, E. Prempan, and Q. H. Wu. “An improved particle swarm optimizer for mechanical design optimization problems”. In: *Engineering optimization* 36.5 (2004), pp.585–605.
- [10] M. Jamil and X. Yang. “A Literature Survey of Benchmark Functions for Global Optimisation Problems”. In: *Journal of Numerical Optimization* 4.2 (2013), pp.150–194.
- [11] S. H. Kamsani. “Improvements on the bees algorithm for continuous optimisation problems (Doctoral dissertation)”. PhD thesis. University of Birmingham, 2016.
- [12] B. K. Kannan and S. N. Kramer. “Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and Its Applications to Mechanical Design”. In: *Journal of Mechanical Design, Transactions of the ASME* 116.2 (1994), pp.405–411.

- [13] D. Karaboga and B. Gorkemli. “A quick artificial bee colony-qABC-algorithm for optimization problems.” In: *2012 International symposium on innovations in intelligent systems and applications*. Trabzon, Turkey, 2012, pp.1–5.
- [14] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar. “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions”. In: *IEEE transactions on evolutionary computation* 10.3 (2006), pp.281–295.
- [15] E. Mezura-Montes and C. A. Coello. “Useful infeasible solutions in engineering optimization with evolutionary algorithms.” In: *MICAI 2005: Advances in Artificial Intelligence*. Berlin, Germany: Springer, 2005, pp.652–662.
- [16] J. Nievergelt, R. Gasser, F. Mäser, and C. Wirth. “All the needles in a haystack: Can exhaustive search overcome combinatorial chaos?” In: *In Computer Science Today 1000* (1995), PP.254–274.
- [17] K. E. Parsopoulos and M. N. Vrahatis. “Unified particle swarm optimization for solving constrained engineering optimization problems”. In: *First International conference on natural computation, ICNC 2005, Changsha, China*. Berlin, Germany: Springer, 2005, pp.582–591.
- [18] D. T. Pham, M. Castellani, and A. Fahmy. “Learning the inverse kinematics of a robot manipulator using the bees algorithm”. In: *2008 6th IEEE International Conference on Industrial Informatics*. IEEE, 2008, pp.493–498.
- [19] D. T. Pham and H. A. Darwish. “Using the bees algorithm with Kalman filtering to train an artificial neural network for pattern classification”. In: *Proceedings of the Institution of Mechanical Engineers. Journal of Systems and Control Engineering*, 2010, pp.885–892.
- [20] D. T. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi. *The bees algorithm Technical Note*. Tech. rep. Manufacturing Engineering Centre, Cardiff University, UK, 2005.
- [21] D. T. Pham, Q. T. Pham, A. Ghanbarzadeh, and M. Castellani. “Dynamic optimisation of chemical engineering processes using the bees algorithm”. In: *IFAC Proceedings* 41.2 (2008), pp.6100–6105.
- [22] T. Ray and K. M. Liew. “Society and civilization: An optimization algorithm based on the simulation of social behavior”. In: *IEEE Transactions on Evolutionary Computation* 7.4 (2003), pp.386–396.

- [23] P. Sun, Y. Zhang, J. Liu, and J. Bi. “An Improved Atom Search Optimization With Cellular Automata, a Lévy Flight and an Adaptive Weight Strategy”. In: *IEEE* 8 (2020), pp.49137–49159.
- [24] B. Yuce, D. T. Pham, M. S. Packianather, and E. Mastrocinque. “An enhancement to the Bees Algorithm with slope angle computation and Hill Climbing Algorithm and its applications on scheduling and continuous-type optimisation problem”. In: *Production & Manufacturing Research* 3.1 (2015), pp.3–19.