EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
Vol. 16, No. 3, 2023, 1608-1623
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Using the modified artificial bee colony algorithm to find the non-Archimedean epsilon for evaluating the efficiency in DEA

Sara Zeidani[1], Babak Asady[2], Mohsen Rostamy- Malkhalifeh[3,*], Taher Lotfi[1]

[1] *Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran*
[2] *Department of Mathematics, Arak Branch, Islamic Azad University, Arak, Iran*
[3] *Department of Mathematics, Science and Research Branch, Islamic Azad University, Tehran, Iran*

Abstract. The artificial bee colony algorithm is one of the population-based optimization methods inspired by the evolutionary principles of the social behavior of bees. On the other hand, one of the sub-fields of operations research science is data envelopment analysis. There are some difficulties in DEA models for selecting the appropriate numerical value for an infinitesimal non-Archimedean epsilon. So far, various methods have been proposed to solve this problem and choose the suitable non-Archimedean epsilon. In order to solve the problem, the artificial bee colony algorithm (ABC), and modification of the original ABC algorithm (MABC) are adopted and proposed in this paper. The impacts of our proposed algorithms on the suitable non-Archimedean epsilon by solving only one linear programming (LP), instead of n LP are investigated. Finally, the performance of the proposed algorithms is evaluated by comparing the solutions obtained from GAMS software based on the presented examples.

2020 Mathematics Subject Classifications: 90C08, 68T20, 68W20

Key Words and Phrases: Artificial bee colony algorithm, Data envelopment analysis, Linear programming, Non-Archimedean infinitesimal

1. Introduction

Well-known population-based meta-heuristic algorithms include evolutionary computation [23], particle swarm optimization method [18], genetic algorithm [13], ant colony optimization[10], artificial bee colony algorithm [15], and so on. The ABC algorithm has been used in many topics such as the numerical function optimization [16], neural networks [17], real parameter optimization or fuzzy polynomial interpolation ([1], [6], and [19]). It should be noted that this algorithm was introduced by [15]. Also, the different

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v16i3.4780

Email addresses: sara.zeidani@gmail.com (S. Zeidani), basadi@iau-arak.ac.ir (B. Asady), mohsen_ rostamy@yahoo.com (M. Rostamy-Malkhalifeh), lotfi@iauh.ac.ir (T. Lotfi)

ratios of the onlooker and employee bees for the first modifications of the artificial bee colony algorithm suggested by [4]. They have proposed these ratios {1:1, 1:2, 1:3, 1:4, 2:1, 3:1, 4:1, 2:2, 2:3, 3:2} and also have considered the execution of the main loop of the algorithm to be constant for different ratios. [4] showed that in the modified ABC algorithm, more onlooker bees had a better effect on the results. On the other hand, one of the sub-fields of operations research science is data envelopment analysis. Data envelopment analysis is an effective way to evaluate and analyze the efficiency of systems with multiple inputs and outputs. One of the practical topics in data envelopment analysis is finding the efficiency of decision-making units based on the indicators considered in the problem. The first DEA model introduced by Professor Cooper, Charnes and Rhodes in 1978 was the CCR model [7]. This model became the basis of many studies in the DEA. One of the problems in solving this model is that it declares inefficient units as efficient, which happens due to finding zero weights. The first version of the DEA model was updated in 1979 with the addition of a non-Archimedean ε as the lower limit for inputs and outputs weights of DMU under evaluation [8]. Various methods have been suggested to calculate the appropriate value for ε. [2] proposed a method for finding a suitable value for ε. [20] modified this technique and presented an LP to select a suitable value for ε. [21] showed that there exists threshold value and if the epsilon is smaller than this value, the solution to the single stage program is exact. So far, some researchers have prepared methods and studies about the non- Archimedean ε such as [5], [3], [11], [24], [25], and [27] but some proposed techniques may not lead to the correct recognition of efficient and inefficient DMUs. [9], as the first and most significant alternative technique to epsilon based DEA solutions proposed the two-phase method for evaluating efficiency in DEA without helping of ε. In this proposed method, the efficiency of each DMU obtains by solving two LPs. Besides, DEA requires huge computer resources in terms of memory and CPU time for a large data set with many input/output variables and/or DMUs and takes a long time even with a very fast computer [12]. In addition, for each DMU unit, a mathematical programming problem must be solved separately. Therefore, to solve this problem, some methods have been proposed and used in related works. For example, to estimate efficiency frontiers, neural networks (NN) can be considered as a possible alternative to replace or complement the DEA technique [26]. One of the studies that explicitly considered the big data set is the work of [12]. They considered five large random data sets of DMUs to measure efficiency and by analyzing the results obtained by conventional DEA with a back-propagation neural network; they showed that the estimation error decreases for larger data sets. As determining an appropriate value for epsilon is a challenging issue in the literature of DEA, as well as in recent years, the population-based algorithms have been used a lot to solve various problems in data envelopment analysis (see [14] and [28]). In this work, in order to surmount the mentioned drawbacks, by finding strictly positive weights and estimating the efficiency of DMUs, we developed an efficient metaheuristic algorithm: the MABC algorithm. As explained earlier, many methods to find a suitable value for the non-Archimedean epsilon have been used in previous works. Here, our novelty approach is to use MABC for the first time in this area. Then the performance of our proposed algorithm is evaluated based on the provided examples. As a result, we can evaluate technical efficiency of DMUs based on the MABC algorithm. Finally, the MABC algorithm can help the decision maker to correctly identify the efficient unit without the need to solve many linear programming models and regardless of the size of the problem. The paper is organized as follows. Section 2 reviews the Preliminaries of DEA and this section also contains the procedure of ABC and MABC algorithms. The analyses and results from ABC and modified ABC algorithms and Gams software for finding the non Archimedean epsilon are detailed in Sections 3 and 4. Finally, section 5 concludes this work.

2. Preliminaries

2.1. Data Envelopment Analysis

The first model of data envelopment analysis is called CCR [7]. In this model, in order to include the inputs and outputs of other decision-making units to determine the optimal weight of the unit under study and also to determine the highest efficiency ratio. Assume that we have n DMUs, DMU_j,$j \in \{1, 2, ..., n\}$ to be evaluated, each DMU using m inputs to produce s outputs. Xj=$\{x_{1j}, ..., x_{mj}\}$ and Yj=$\{y_{1j}, ..., y_{sj}\}$ are the input and output vectors of DMU_j respectively, in which $X_jY_j \geq 0$ and $X_j, Y_j \neq 0$. The virtual input and output are formed with weights v_i and u_r (yet unknown).
virtual input $= v_1x_{1o} + ... + v_mx_{mo}$.
virtual output $= u_1y_{1o} + ... + u_sy_{so}$.
Then, using the basic model introduced below, the weight is determined to reach the maximum ratio.

$$\begin{aligned} & \max \frac{\sum_{r=1}^{s} u_r y_{ro}}{\sum_{i=1}^{m} v_i x_{io}} \\ & \frac{\sum_{r=1}^{s} u_r y_{rj}}{\sum_{i=1}^{m} v_i x_{ij}} \leq 1 \qquad j = 1, 2, ..., n \\ & v_i \geq 0 \qquad i = 1, 2, ..., m \\ & u_r \geq 0 \qquad r = 1, 2, ..., s \end{aligned} \tag{1}$$

This problem is denominated the CCR fractional model. At the suggestion of Charnes and Cooper, by imposing constraints on the CCR fractional programming model, this model became the following linear programming model, in which DMU_o,$o \in \{1, 2, ..., n\}$ is the unit under evaluation.

$$\begin{aligned} \max Z_o = & \sum_{r=1}^{s} u_r y_{ro} \\ & \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 \qquad j = 1, 2, ..., n \\ & \sum_{i=1}^{m} v_i x_{io} = 1 \\ & v_i \geq 0 \qquad i = 1, 2, ..., m \\ & u_r \geq 0 \qquad r = 1, 2, ..., s \end{aligned} \tag{2}$$

Model (2) states that if the optimal value of the objective function is equal to 1 and there is at least one optimal solution with all i and r, $v_i > 0$,$u_r > 0$ it will be strong efficient. In order to find the strong efficiency units, both models 3 and 4 are solved simultaneously.

$$\begin{array}{ll} \max \varepsilon & \\ \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 & j = 1, 2, ..., n \\ \sum_{i=1}^{m} v_i x_{ij} \leq 1 & j = 1, 2, ..., n \\ v_i - \varepsilon \geq 0 & i = 1, 2, ..., m \\ u_r - \varepsilon \geq 0 & r = 1, 2, ..., s \end{array} \tag{3}$$

Optimal solution of this model (ε^*) can be considered as a lower bound for the variables v_i and u_r in the multiplier CCR model.

$$\begin{array}{ll} \max Z_o = \sum_{r=1}^{s} u_r y_{ro} & \\ \sum_{r=1}^{s} u_r y_{rj} - \sum_{i=1}^{m} v_i x_{ij} \leq 0 & j = 1, 2, ..., n \\ \sum_{i=1}^{m} v_i x_{io} = 1 & \\ v_i \geq \varepsilon^*, & i = 1, 2, ..., m \\ u_r \geq \varepsilon^*, & r = 1, 2, ..., s \end{array} \tag{4}$$

Epsilon is used to prevent the removal of inputs and outputs due to weights being zero. Also, epsilon is called a Non-Archimedean infinitesimal.

2.2. Artificial bee colony algorithm

The ABC algorithm consists of two search methods: local search and global search, employed and onlooker bees performed the local search, and onlookers and scouts managed the global search [15]. In the search space of the ABC algorithm, employed bees are sent to food sources to measure their nectar. After sharing information with the employed bees, the amount of nectar in the food source is determined and the onlooker bees select the food sources. Then the scout bees are sent to food sources after identification.
In summary, The ABC algorithm actually employs different selection processes:

01. The overall selection process described by artificial onlooker bees to explore the promising areas described. The onlooker bee selects each food source based on the probability value associated with it, and this selection is calculated using the following formula:

$$fitness_i = \begin{cases} \frac{1}{(1+f_i)} & if \quad f_i \geq 0 \\ 1 + abs(f_i) & if \quad f_i < 0 \end{cases} \tag{5}$$

and

$$P_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{6}$$

Also, f_i is the suitability of solution i, which is obtained by evaluating the amount of nectar in position i.

02. A local selection process in the area is performed by artificial bees and the onlookers are given the local information to determine a food source neighbor around a previously defined source of memory; this greedy selection is computed by:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), k \in \{1, 2, ..., SN\}, j = \{1, 2, ..., D\}, k \neq i, \phi_{ij} \in [-1, 1] \quad (7)$$

From the above explanation, it is clear that three control parameters of the main artificial bee colony algorithm are used:

01. The number of food sources (SN)
02. Limit value
03. Maximum number of cycles

2.3. Modified ABC algorithm

The different ratios of the onlooker and employee bees for the first modifications of the artificial bee colony algorithm suggested by [4]. Their proposed algorithms have been tested for optimizing some well- known numerical functions. The details of examples are shown in [4]. They have proposed these ratios {1:1, 1:2, 1:3, 1:4, 2:1, 3:1, 4:1, 2:2, 2:3, 3:2} and also have considered the execution of the main loop of the algorithm to be constant for different ratios. [4] showed that in the modified ABC algorithm, more onlooker bees had a better effect on the results.

3. Proposed modified artificial bee colony algorithm for finding the non-Archimedean epsilon

In this paper, we have used these ratios {1:2, 1:3, 1:4, 2:1, 3:1, 4:1} taken from the modified ABC algorithm presented by [4]. Therefore, for these ratios, the population size will be {66, 50, 40, 66, 50, 40}, respectively. Also, the maximum number of cycle iterations is 3000 for two examples. Furthermore, we have determined ε^* from the multiplier CCR model without needing to solve two-stage approach. So, ε^* may be non-unique but positive. It has been shown that using the modified ABC algorithm determines the magnitude of epsilon referring to two illustrative examples, however, it has not been mentioned how one can determine an assurance value for the non-Archimedean ε^*. The pseudo-code of our proposed modified ABC algorithm for solving the multiplier CCR model is given below:

01. Initialize population solutions v_i and u_r
02. Set maximum cycle number (MCN=3000),
03. Evaluate the fitness function value (the CCR model (2))
04. For each decision maker unit
05. Assessment of the fitness value for each solution,

06. Repeat, Cycle=1, for the different ratios of employed and onlooker bees {1:2, 1:3, 1:4, 2:1, 3:1, 4:1}, also for the ratios mentioned above, the population size will be {66, 50, 40, 66, 50, 40}},
07. Produce the new solution v_{ij} for the employed bees by (7),
08. Apply the greedy selection process,
09. Calculate the probability values by using (6),
10. Apply the greedy selection process between the old solution and new solution,
11. Determine the abandoned solution and replace it with a new randomly produced solution,
12. Memorize the optimized solution achieved so far,
13. Cycle= cycle+1, until cycle= MCN,
14. End.

4. Experiments

Most classical mathematical methods consider the local optimal point as a global optimal. In each of these techniques, the number of calculations increases exponentially and is used for difficult and special problems. One of these problems is determining ε^* in the DEA model which has been a controversial subject for a long time, so we used ABC and modified ABC method for solving the above CCR model (2) optimization problem and we find the best values of weight coefficients and ε^*. The accuracy of ABC and MABC algorithms is compared with the GAMS software. All numerical examples and results will be explained in 4.1 and 4.2, respectively.

4.1. Example 1:

Table 1 exhibits a numerical example that was previously used by [27]. In this example, there are five DMUs, each using two inputs to generate a single unitized output. We obtain the weight coefficients v_1,v_2 and u_1 by solving Gams software and the original ABC and MABC algorithms.

Table 1: The data set of first numerical example.

DMU	I_1	I_2	O_1
1	1	2	1
2	2	1	1
3	4	1	1
4	1	4	1
5	1	8	1

Table 2: The results of a multiplier CCR model (2) by using Gams software.

DMU	Efficiency	v_1	v_2	u_1
1	1.00000	1.00000	0.00000	1.00000
2	1.00000	0.00000	1.00000	1.00000
3	1.00000	0.00000	1.00000	1.00000
4	1.00000	1.00000	0.00000	1.00000
5	1.00000	1.00000	0.00000	1.00000

Table 3: The results of the CCR models 3 and 4 by using Gams software($\varepsilon^* = 0.11111$).

DMU	z	v_1	v_2	u_1
1	1.00000	0.77778	0.11111	1.00000
2	1.00000	0.11111	0.77778	1.00000
3	0.77778	0.11111	0.55556	0.77778
4	0.77778	0.55556	0.11111	0.77778
5	0.33333	0.11111	0.11111	0.33333

Table 4: The results of the basic ABC algorithm [SN=100 and ratio 1:1] on CCR model (2).

DMU	z^*	v_1	v_2	u_1	$\lvert z - z^* \rvert$
1	1.00133	0.77618	0.11291	1.00133	0.00133
2	1.00131	0.11292	0.77616	1.00131	0.00131
3	0.77904	0.11407	0.54572	0.77904	0.00126
4	0.77901	0.54564	0.11409	0.77901	0.00123
5	0.33460	0.11114	0.11123	0.33460	0.00127

Table 5: The results of MABC algorithm on CCR model (2) for ratio 4:1 (employed/onlooker)).

DMU	z^*	v_1	v_2	u_1	$\lvert z - z^* \rvert$
1	1.00104	0.77644	0.11218	1.00104	0.00104
2	1.00105	0.11219	0.77642	1.00105	0.00105
3	0.77883	0.11328	0.54778	0.77883	0.00105
4	0.77882	0.54766	0.11331	0.77882	0.00104
5	0.33436	0.11112	0.11121	0.33436	0.00103

Table 6: The results MABC algorithm on CCR model (2) for ratio 3:1.

DMU	z^*	v_1	v_2	u_1	$\|z - z^*\|$
1	1.00078	0.77658	0.11196	1.00078	0.00078
2	1.00076	0.11197	0.77656	1.00076	0.00076
3	0.77852	0.11251	0.55046	0.77852	0.00074
4	0.77853	0.55050	0.11250	0.77853	0.00075
5	0.33408	0.11111	0.11120	0.33408	0.00075

Table 7: The results MABC algorithm on CCR model (2) for ratio 2:1.

DMU	z^*	v_1	v_2	u_1	$\|z - z^*\|$
1	1.00042	0.77724	0.11148	1.00042	0.00042
2	1.00044	0.11147	0.77726	1.00044	0.00044
3	0.77817	0.11213	0.55168	0.77817	0.00039
4	0.77815	0.55170	0.11215	0.77815	0.00037
5	0.33374	0.11109	0.11118	0.33374	0.00041

Table 8: The results MABC algorithm on CCR model (2) for ratio 1:2.

DMU	z^*	v_1	v_2	u_1	$\|z - z^*\|$
1	1.00025	0.77757	0.11125	1.00025	0.00025
2	1.00023	0.11125	0.77758	1.00023	0.00023
3	0.77803	0.11157	0.55380	0.77803	0.00025
4	0.77802	0.55381	0.11157	0.77802	0.00024
5	0.33356	0.11106	0.11117	0.33356	0.00023

Table 9: The results MABC algorithm on CCR model (2) for ratio 1:3.

DMU	z^*	v_1	v_2	u_1	$\|z - z^*\|$
1	1.00008	0.77870	0.11067	1.00008	0.00008
2	1.00007	0.11066	0.77872	1.00007	0.00007
3	0.77788	0.11106	0.55580	0.77788	0.00010
4	0.77787	0.55580	0.11111	0.77787	0.00009
5	0.33341	0.11104	0.11113	0.33341	0.00008

Table 10: The results MABC algorithm on CCR model (2) for ratio 1:4.

DMU	z^*	v_1	v_2	u_1	$\|z - z^*\|$
1	1.00002	0.77889	0.11056	1.00002	0.00002
2	1.00002	0.11056	0.77890	1.00002	0.00002
3	0.77779	0.11094	0.55626	0.77779	0.00001
4	0.77777	0.55654	0.11087	0.77777	0.00001
5	0.33335	0.11101	0.11112	0.33335	0.00002

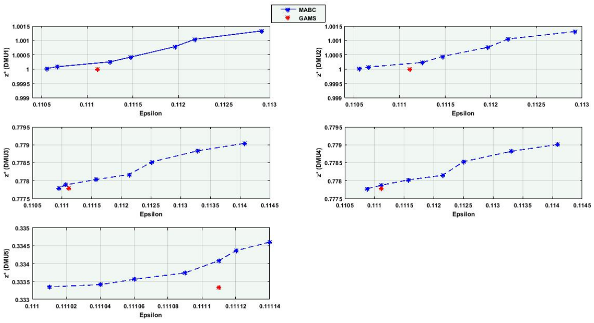

Figure 1: Comparison of GAMS, ABC and MABC in terms of ε^*.

Using GAMS software, the basic CCR model (2) and both models 3 and 4 are solved and the results are presented in Tables 2 and 3. There are two CCR-efficient DMUs from both models 3 and 4 simultaneously. It should be noticed that model (2) is unable to accurately determine the efficient DMU. In order to handle this inconsistency, we propose the following modified ABC algorithm to determine an appropriate epsilon for the model (2). The results of the above example, it can be seen that the group of onlooker bees with a higher ratio has a better effect on the obtained results. In addition, by using Gams software the optimal epsilon value of models (3) and (4) for this data set is equal to $\varepsilon^* = 0.11111$. Obviously, in this situation, the decision maker to select efficient units must solve two models to achieve a more accurate solution. So, according to the obtained results, DMU_3 and DMU_4 and DMU_5 drop to inefficient for the Gams software and the original ABC and modified ABC algorithms (See Figure 1). Also, $|z - z^*|$ is the difference between z from the multiplier CCR models 3 and 4 (red star in Figure 1), and z^* from the Basic ABC and MABC algorithms. As Tables 4-10 indicate, ε^* is a parameter ranging the interval [0, 1], and ε^* had the most decrease in Table 10.

4.2. Example 2:

This example has been sketched for nine units with two inputs and two outputs. Whose data has been listed in Table 11 [22]. We obtained the weight coefficients v_1,v_2 and u_1,u_2 by using Gams software and the original ABC and MABC algorithms. Comparison of GAMS, ABC, and MABC in terms of ε^* shows in Figure 2.

Table 11: The data set of second numerical example.

DMU	I_1	I_2	O_1	O_2
1	3	4	2	3
2	2	1	3	5
3	1	1	4	6
4	5	1	7	6
5	10	1	1	8
6	2	3	3	4
7	3	4	8	5
8	2	6	2	4
9	0.5	0.5	1	1

Table 12: The results of a multiplier CCR model (2) by using Gams software.

DMU	Efficiency	v_1	v_2	u_1	u_2
1	0.16667	0.33333	0.00000	0.00000	0.05556
2	0.80515	0.02206	0.95588	0.02941	0.14338
3	1.00000	1.00000	0.00000	0.00000	0.16667
4	1.00000	0.10714	0.46429	0.14286	0.00000
5	1.00000	0.00000	1.00000	0.04000	0.12000
6	0.37500	0.50000	0.00000	0.12500	0.00000
7	0.66667	0.33333	0.00000	0.08333	0.00000
8	0.33333	0.50000	0.00000	0.00000	0.08333
9	0.50000	2.00000	0.00000	0.50000	0.00000

Table 13: The results of the CCR models 3 and 4 by using Gams software ($\varepsilon^* = 0.02241$).

DMU	Efficiency	v_1	v_2	u_1	u_2
1	0.16293	0.30345	0.02241	0.02241	0.03937
2	0.80511	0.02241	0.95517	0.02865	0.14383
3	1.00000	0.97759	0.02241	0.02241	0.15172
4	1.00000	0.02241	0.88793	0.12365	0.02241
5	0.96724	0.02241	0.77586	0.02241	0.11810
6	0.35539	0.46638	0.02241	0.08858	0.02241
7	0.49483	0.30345	0.02241	0.04784	0.02241
8	0.28851	0.43276	0.02241	0.02241	0.06092
9	0.48879	1.97759	0.02241	0.46638	0.02241

Table 14: The results of the basic ABC algorithm on CCR model (2).

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16718	0.30255	0.02409	0.02408	0.03967	0.00425
2	0.80912	0.02473	0.95314	0.03081	0.14334	0.00401
3	1.00466	0.97999	0.02401	0.02405	0.15141	0.00466
4	1.00285	0.02355	0.88510	0.12309	0.02354	0.00285
5	0.97375	0.02473	0.75770	0.02472	0.11860	0.00651
6	0.35900	0.46565	0.02390	0.08784	0.02387	0.00361
7	0.49838	0.30232	0.02401	0.04732	0.02396	0.00355
8	0.29195	0.43102	0.02351	0.02348	0.06125	0.00344
9	0.49280	1.98319	0.02441	0.46840	0.02440	0.00401

Table 15: The results of MABC algorithm on CCR model (2) for ratio 4:1 (employed/onlooker)).

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16696	0.30277	0.02380	0.02382	0.03977	0.00403
2	0.80886	0.02433	0.95344	0.03057	0.14343	0.00375
3	1.00446	0.97970	0.02380	0.02381	0.15154	0.00446
4	1.00261	0.02324	0.88630	0.12333	0.02322	0.00261
5	0.97336	0.02422	0.76240	0.02422	0.11864	0.00612
6	0.35851	0.46583	0.02361	0.08802	0.02361	0.00312
7	0.49790	0.30241	0.02382	0.04736	0.02380	0.00307
8	0.29172	0.43188	0.02314	0.02309	0.06138	0.00321
9	0.49265	1.98331	0.02409	0.46858	0.02407	0.00386

Table 16: The results of MABC algorithm on CCR model (2) for ratio 3:1.

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16669	0.30280	0.02365	0.02365	0.03980	0.00376
2	0.80841	0.02390	0.95350	0.03033	0.14348	0.00330
3	1.00415	0.97955	0.02355	0.02355	0.15166	0.00415
4	1.00242	0.02304	0.88680	0.12348	0.02301	0.00242
5	0.97303	0.02397	0.76440	0.02396	0.11867	0.00579
6	0.35835	0.46585	0.02343	0.08820	0.02344	0.00296
7	0.49758	0.30253	0.02363	0.04743	0.02362	0.00275
8	0.29154	0.43215	0.02295	0.02291	0.06143	0.00303
9	0.49249	1.98333	0.02387	0.46863	0.02386	0.00370

Table 17: The results of MABC algorithm on CCR model (2) for ratio 2:1.

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16648	0.30293	0.02343	0.02341	0.03989	0.00355
2	0.80808	0.02356	0.95358	0.03002	0.14360	0.00297
3	1.00391	0.97934	0.02326	0.02328	0.15180	0.00391
4	1.00215	0.02278	0.88772	0.12363	0.02279	0.00215
5	0.97277	0.02371	0.76590	0.02373	0.11868	0.00553
6	0.35810	0.46610	0.02313	0.08854	0.02312	0.00271
7	0.49739	0.30268	0.02344	0.04752	0.02344	0.00256
8	0.29130	0.43256	0.02273	0.02274	0.06146	0.00279
9	0.49232	1.98335	0.02365	0.46867	0.02365	0.00353

Table 18: The results of MABC algorithm on CCR model (2) for ratio 1:2.

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16622	0.30311	0.02317	0.02316	0.03997	0.00329
2	0.80782	0.02320	0.95390	0.02972	0.14373	0.00271
3	1.00375	0.97910	0.02310	0.02309	0.15190	0.00375
4	1.00184	0.02253	0.88856	0.12381	0.02253	0.00184
5	0.97235	0.02345	0.76791	0.02344	0.11868	0.00511
6	0.35784	0.46627	0.02285	0.08881	0.02285	0.00245
7	0.49697	0.30299	0.02313	0.04768	0.02310	0.00214
8	0.29105	0.43276	0.02258	0.02256	0.06148	0.00254
9	0.49220	1.98357	0.02343	0.46880	0.02340	0.00341

Table 19: The results of MABC algorithm on CCR model (2) for ratio 1:3.

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16594	0.30331	0.02292	0.02290	0.04005	0.00301
2	0.80764	0.02295	0.95430	0.02926	0.14397	0.00253
3	1.00354	0.97901	0.02279	0.02278	0.15207	0.00354
4	1.00153	0.02230	0.88931	0.12396	0.02230	0.00153
5	0.97209	0.02315	0.77050	0.02315	0.11870	0.00485
6	0.35752	0.46648	0.02258	0.08909	0.02256	0.00213
7	0.49676	0.30320	0.02285	0.04781	0.02285	0.00193
8	0.29091	0.43295	0.02245	0.02245	0.06151	0.00240
9	0.49205	1.98372	0.02328	0.46882	0.02323	0.00326

Table 20: The results of MABC algorithm on CCR model (2) for ratio 1:4.

DMU	z^*	v_1	v_2	u_1	u_2	$\|z - z^*\|$
1	0.16573	0.30338	0.02274	0.02274	0.04008	0.00280
2	0.80749	0.02274	0.95453	0.02898	0.14411	0.00238
3	1.00337	0.97893	0.02257	0.02257	0.15218	0.00337
4	1.00131	0.02220	0.88955	0.12402	0.02219	0.00131
5	0.97133	0.02290	0.77280	0.02289	0.11874	0.00444
6	0.35720	0.46656	0.02239	0.08923	0.02238	0.00181
7	0.49663	0.30324	0.02272	0.04788	0.02271	0.00180
8	0.29069	0.43317	0.02229	0.02230	0.06153	0.00218
9	0.49179	1.98379	0.02301	0.46885	0.02294	0.00300

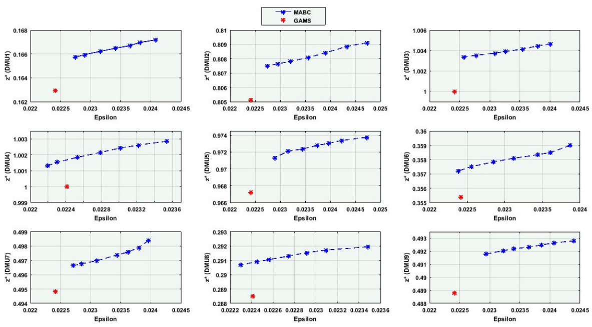

Figure 2: Comparison of GAMS, ABC and MABC in terms of ε^*.

We have three efficient DMUs from Table 12, but their efficiency scores have decreased in Table 13. Now we apply the modified ABC algorithm for Table 11 (model 2) which gives us $\varepsilon^* = 0.02219$ for DMU_4. As we expected, this value is less than $\varepsilon^* = 0.02241$. As we pointed out, better results are obtained when the ratio of onlooker bees is higher. It can be easily seen that in model (2), alternative optimal solutions exist and consequently the best DMU cannot be determined, correctly. On the other hand, as shown in Table 12, for DMU_5, the optimal weight of the first input is equal to zero, hence DMU5 is efficient. Indeed, in this stage, this model fails to find efficient DMUs. In conclusion, the efficiency of each DMU obtains by solving two LPs. Similar to the previous example, $|z - z^*|$ value shows the convergence rate of algorithms and it's a variable that is a good criterion for comparing the exact results from Gams and the goodness of various algorithms. It is noteworthy that the ABC and MABC are simulated by randomly generating a situation.

5. Conclusion

In this work, we used ABC and MABC methods for solving the CCR model (multiplier form) of the optimization problems. Using the proposed method, the decision maker is able to detect the strong efficiency units by solving only one LP, instead of n LP, therefore, can achieve faster results. In ABC and MABC, we attempt to find an optimal solution to the CCR model which satisfies all the constraints. Solving models by using the MABC algorithm for computing the suitable non-Archimedean epsilon without needing to solve the two-stage approach and the ability to find efficient DMUs are the advantages of the new method. Theoretically, we will emphasize that this one-stage MABC algorithm is solvable, although we have already done this practically. Unfortunately, there is no single solution for dealing with the general DEA models, because of the unstable behavior of the meta-heuristic algorithms. It is worth noting that the model used in this paper is input-oriented, but can also be extended to output-oriented. Although in this paper, we have dealt with the case of constant returns to scale, we can apply this model to other cases of returns to scale by imposing restrictions on the initial intensity vector. Also, despite this modification that is mentioned, there can be some new modifications on other aspects of ABC algorithm that are remained for future works.

References

[1] Bahriye Akay and Dervis Karaboga. A modified artificial bee colony algorithm for real-parameter optimization. *Information Sciences*, 192:120–142, 2012.

[2] Agha Iqbal Ali and Lawrence M. Seiford. Computational accuracy and infinitesimals in data envelopment analysis. *INFOR*, 31(4):290–297, 1993.

[3] Mohammadreza Alirezaee and Masoud Khalili. Recognizing the efficiency, weak efficiency and inefficiency of dmus with an epsilon independent linear program. *Applied Mathematics and Computation*, 183:1323–1327, 2006.

[4] Amir Alizadegan, Babak Asady, and Mohammad Ahmadpour. Two modified versions of artificial bee colony algorithm. *Applied Mathematics and Computation*, 225:601–609, 2013.

[5] Gholam.R. Amin and Mehdi Toloo. A polynomial-time algorithm for finding epsilon in dea models. *Computers and Operations Research*, 31(5):803–805, 2004.

[6] Babak Asady, Parvaneh Mansouri, and Neelima Gupta. The modify version of artificial bee colony algorithm to solve real optimization problems. *International Journal of Electrical and Computer Engineering*, 2:473–480, 2012.

[7] Abraham Charnes, William W. Cooper, and Edwardo Rhodes. Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6):429–444, 1978.

[8] Abraham Charnes, William W. Cooper, and Edwardo Rhodes. Short communication: Measuring the efficiency of decision making units. *European Journal of Operational Research*, 3(4):338–339, 1979.

[9] William W. Cooper, Lawrence M. Seiford, and Kaoru Tone. Data envelopment analysis: A comprehensive text with models, applications, references and dea-solver software. *Kluwer Academic Publishers*, 1999.

[10] Marco. Dorigo, Vittorio Maniezzo, and Alberto Colorni. The ant system: optimization by a colony of cooperating agents. *IEEE Transactions On Systems Man and cybernetics Part B*, 26:1–13, 1996.

[11] Bohlool Ebrahimi and Ehsan Hajizadeh. A novel dea model for solving performance measurement problems with flexible measures: An application to tehran stock exchange. *Measurement*, 179:109444, 2021.

[12] Ali Emrouznejad and Estelle Shale. A combined neural network and dea for measuring efficiency of large scale datasets. *Computers and Industrial Engineering*, 56(1):249–254, 2009.

[13] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley Professional, New York, 1989.

[14] Martin Gonzalez, Jose J. Lopez-Espin, Juan Aparicio, Domingo Gimenez, and Jesus T. Pastor. Using genetic algorithms for maximizing technical efficiency in data envelopment analysis. *Procedia Computer Science*, 51:374–383, 2015.

[15] Dervis Karaboga. An Idea Based On Honey Bee Swarm for Numerical Optimization. In *Technical Report-TR06.*, Erciyes University, 2005. Engineering Faculty.

[16] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Procedia Computer Science*, 39(3):459–471, 2007.

[17] Dervis Karaboga, Bahriye Basturk, and Celal Ozturk. Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks. *LNCS: Modeling 173 Decisions for Artificial Intelligence*, 4617(Springer-Verlag):318–329, 2007.

[18] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks.*, volume 4, pages 1942–1948 vol.4, 1995.

[19] Parvaneh Mansouri, Babak Asady, and Neelima Gupta. An approximation algorithm for fuzzy polynomial interpolation with artificial bee colony algorithm. *Applied Soft Computing*, 13:1997–2002, 2013.

[20] Saeid Mehrabian, Gholam R. Jahanshahloo, Mohammadreza Alirezaee, and Gholam R. Amin. An assurance interval for the non-archimedean epsilon in dea models. *Operations Research*, 48(2):3444–3447, 2000.

[21] Victor V. Podinovski and Tatiana Bouzdine-Chameeva. Solving dea models in a single optimization stage: Can the non-archimedean infinitesimal be replaced by a small finite epsilon? *European Journal of Operational Research*, 257(2):412–419, 2017.

[22] Majid Soleimani-damaneh and Amin Mostafaee. Stability of the classification of returns to scale in fdh models. *European Journal of Operational Research*, 196:1223–1228, 2009.

[23] Rainer M. Storn and Kenneth Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous. *Journal of Global Optimization*, 11:341–359, 1997.

[24] Mehdi Toloo. The role of non-archimedean epsilon in finding the most efficient unit: With an application of professional tennis players. *Applied Mathematical Modelling*, 38:5334–5346, 2014.

[25] Mehdi Toloo, Bohlool Ebrahimi, and Gholam R. Amin. New data envelopment analysis models for classifying flexible measures: The role of non-archimedean epsilon. *European Journal of Operational Research*, 292(3):1037–1050, 2021.

[26] Mehdi Toloo, Ameneh Zandi, and Ali Emrouznejad. Evaluation efficiency of large-scale data set with negative data: An artificial neural network approach. *The Journal of Supercomputing*, 71(7):2397–2411, 2015.

[27] Kaoru Tone and Miki Tsutsui. An epsilon-based measure of efficiency in dea revisited-a third pole of technical efficiency. *European Journal of Operational Research*, 207(3):1554–1563, 2010.

[28] Yeboon Yun, Hirotaka Nakayama, and Min Yoon. Generation of pareto optimal solutions using generalized dea and pso. *Journal of Global Optimization*, 64(1):49–61, 2016.