Düzce Üniversitesi
Journal of Science & Technology

Research Article

A Comprehensive Comparison of Binary Archimedes Optimization Algorithms on Uncapacitated Facility Location Problems

Ahmet Cevahir CINAR [a,*]

[a] *Department of Computer Engineering, Faculty of Technology, Selçuk University, Konya, TÜRKİYE*
** Corresponding author's e-mail address: accinar@selcuk.edu.tr*
DOI: 10.29130/dubited.876284

ABSTRACT

Metaheuristic optimization algorithms are widely used in solving NP-hard continuous optimization problems. Whereas, in the real world, many optimization problems are discrete. The uncapacitated facility location problem (UFLP) is a pure discrete binary optimization problem. Archimedes optimization algorithm (AOA) is a recently develop metaheuristic optimization algorithm and there is no binary variant of AOA. In this work, 17 transfer functions (TF1-TF17) are used for mapping continuous values to binary values. 17 binary variants of AOA (BAOA1- BAOA17) are proposed for solving UFLPs. 16 to 100-dimensional UFLPs were solved with binary variants of AOA. Stationary and non-stationary transfer functions were compared in terms of solution quality. The non-stationary transfer functions were produced better solutions than stationary transfer functions. Peculiar parameter analyzes for binary optimization problems were performed in the best variant (BAOA9) produced with TF9 transfer function.

Keywords: *Binary optimization, Uncapacitated facility location problem, Archimedes optimization algorithm*

Kapasitesiz Tesis Yerleşim Problemleri Üzerinde İkili Arşimet Optimizasyon Algoritmalarının Kapsamlı Bir Karşılaştırması

ÖZ

Meta-sezgisel optimizasyon algoritmaları, NP-zor sürekli optimizasyon problemlerinin çözümünde yaygın olarak kullanılmaktadır. Oysa gerçek dünyada pek çok optimizasyon problemi ayrıktır. Kapasitesiz tesis yerleşimi problemi, saf bir ayrık ikili optimizasyon problemidir. Arşimet optimizasyon algoritması (AOA), yakın zamanda geliştirilmiş bir meta-sezgisel optimizasyon algoritmasıdır ve AOA'nın ikili bir varyantı yoktur. Bu çalışmada, sürekli değerleri ikili değerlere eşlemek için 17 transfer fonksiyonu (TF1-TF17) kullanılmıştır. UFLP'leri çözmek için AOA'nın (BAOA1-BAOA17) 17 ikili varyantı önerilmiştir. 16 ila 100 boyutlu UFLP'ler, AOA'nın ikili varyantları ile çözülmüştür. Durağan ve durağan olmayan transfer fonksiyonları çözüm kalitesi açısından karşılaştırılmıştır. Durağan olmayan transfer fonksiyonları, sabit transfer fonksiyonlarından daha iyi çözümler üretmiştir. İkili optimizasyon problemleri için özel parametre analizleri, TF9 transfer fonksiyonu ile üretilmiş olan en iyi varyantta (BAOA9) gerçekleştirilmiştir.

Anahtar Kelimeler: *İkili optimizasyon, Kapasitesiz tesis yerleşimi problemi, Arşimet optimizasyon algoritması*

Received: 07/02/2021, Revised: 01/07/2021, Accepted: 05/07/2021

I. INTRODUCTION

Metaheuristic optimization algorithms are widely used in solving NP-hard problems. In literature, there are many metaheuristic optimization algorithms are existed, such as, Multi-Centered Vortex Search Algorithm [1], Tree-Seed Algorithm [2, 3], Artificial Algae Algorithm [4], Bull Optimization Algorithm [5], Bat Algorithm [6], Roulette Electromagnetic Field Optimization Algorithm [7], Boosting Galactic Swarm Optimization [8]. Metaheuristics are used for solving the various problem, such as, increase the driving safety for autonomous vehicles [9], training the artificial neural network [10], spectrum handoff [11].

Archimedes optimization algorithm (AOA) is a metaheuristic optimization algorithm proposed by Hashim et al. [12] in 2020. The main inspiration source of AOA is Archimedes' Principle. Archimedes' Principle is a law of physics that related to the immersion of objects in the fluid. The buoyant force, the immersion of the object in the fluid, the weight of the displaced fluid are mathematically modeled for solving optimization problems.

The aforementioned algorithms are only solving continuous optimization problems. Whereas, in the real world, many optimization problems are discrete. Traveling salesman problem [13, 14], uncapacitated facility location problem [15] are some of the well-known discrete optimization problems. The uncapacitated facility location problem is a pure binary optimization problem [16]. In literature, there are many binary metaheuristics are existed for solving UFLPs.

Kiran and Gündüz [16] used XOR-based artificial bee colony algorithm for solving UFLPs. A clustering based Genetic Algorithm is used for solving UFLPs by Çelikbilek [17]. Sahman et al. [18] used binary differential search algorithm for solving UFLPs. 8 different binary TSAs with S-shaped and V-shaped transfer functions are proposed for solving UFLPs by Sahman and Cinar [19]. Two binary variants of AAA [20, 21] are proposed for solving UFLPs. Aslan et al. [22] proposed a XOR-based Jaya algorithm for solving UFLPs. 3 binary variants of TSA [23] are proposed for solving UFLPs. Logic gates (LogicTSA), similarity measurement techniques (SimTSA) and a hybrid variant (SimLogicTSA) produced competitive solutions for UFLP.

According to the literature review, there is no binary variant of AOA. In this work, 17 transfer functions (TF1-TF17) are used for mapping continuous values to binary values. 17 binary variants of AOA (BAOA1- BAOA17) are proposed for solving UFLPs. 16 to 100-dimensional UFLPs were solved with binary variants of AOA. Stationary and non-stationary transfer functions were compared in terms of solution quality. The non-stationary transfer functions were produced better solutions than stationary transfer functions. Peculiar parameter analyzes for binary optimization problems were performed in the best variant (BAOA9) produced with TF9 transfer function.

The main contributions of the study are,
(i) the first binarization for AOA was conducted with transfer functions,
(ii) 17 new binary variant of AOA were proposed,
(iii) 15 well-known UFLPs were solved with these binary AOA variants,
(iv) the peculiar parameter analysis were conducted firstly for binary optimization problems,
(v) the non-stationary transfer functions were produced better solutions than stationary transfer functions.

The remainder of the paper is organized as follows. Archimedes optimization algorithm and binary variants of Archimedes optimization algorithm are explained in Section 2 and Section 3, respectively. In Section 4, the experimental setup is presented and in Section 5, the experimental results and discussions are reported. Finally, the study is concluded in Section 6.

II. ARCHIMEDES OPTIMIZATION ALGORITHM

Archimedes optimization algorithm (AOA) is a metaheuristic optimization algorithm proposed by Hashim et al. [12] in 2020. The main inspiration source of AOA is Archimedes' Principle. Archimedes' Principle is a law of physics that related to the immersion of objects in the fluid. The buoyant force, the immersion of the object in the fluid, the weight of the displaced fluid are mathematically modeled for solving optimization problems. The buoyant force is equal to weight of the displaced fluid by the object [24].

AOA is a population-based continuous optimization algorithm. In AOA, objects immersed in the fluid are modeled as individuals in the population. Every individual has volume, density, and acceleration properties. At the initialization phase, individuals are created randomly in a predetermined search space. At every iteration, volume and density values are changed in accordance with the fitness function value. The new position of an object is determined by the updating density, volume, and acceleration values.

The detailed pseudocode with formulas of AOA is given in Figure 1. In Figure 1, N means population size, t_{max} means the maximum iteration number, C1, C2, C3, and C4 are predetermined peculiar parameters of AOA, lb is the lower bound of search space, ub is the upper bound of search space, den_i is the density value of the ith individual, x_i is the position of the ith individual, vol_i is the volume value of the ith individual, $rand$ is a uniform random between 0 and 1, x_{best} is the best individual in the population, den_{best} is the density value of the best individual, vol_{best} is the volume value of the best individual, acc_{best} is the acceleration value of the best individual, acc_i is the acceleration value of the ith individual, t is the iteration number, den_{ri} is the density value of the random individual, vol_{ri} is the volume value of the random individual, acc_{ri} is the acceleration value of the random individual, TF is the transfer operator, d is the density factor, u and l are the range of normalization values and set as 0.9 and 0.1, respectively, acc_{i-norm}^{t+1} is the normalized acceleration value of the ith individual, min (acc) is the minimum value of the acceleration value, $max(acc)$ is the maximum value of the acceleration value, x_{ri} is the random individual, F is the flag for the direction of motion.

Determine the population size (N),
Determine the maximum iteration number (t_{max}),
Determine the peculiar parameters (C1, C2, C3, C4)
 FOR i=1 to N
 $x_i = lb + rand(0,1) \times (ub - lb)$
 $den_i = rand(0,1)$
 $vol_i = rand(0,1)$
 $acc_i = lb + rand(0,1) \times (ub - lb)$
 END
Evaluate the population
Determine the best individual ($x_{best}, den_{best}, vol_{best}, acc_{best}$)
t=1 // Set iteration number
 WHILE t ≤ t_{max}
 FOR i=1 to N
 $den_i^{t+1} = den_i^t + rand(0,1) \times (den_{best} - den_i^t)$
 $vol_i^{t+1} = vol_i^t + rand(0,1) \times (vol_{best} - vol_i^t)$
 $TF = exp\left(\frac{t - t_{max}}{t_{max}}\right)$
 $d^{t+1} = exp\left(\frac{t_{max} - t}{t_{max}}\right) - \left(\frac{t}{t_{max}}\right)$
 IF TF <0.5

$$acc_i^{t+1} = \frac{den_{ri} + vol_{ri} \times acc_{ri}}{den_i^{t+1} \times vol_i^{t+1}}$$
$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} - \min(acc)}{max(acc) - \min(acc)} + l$$
$$x_i^{t+1} = x_i^t + C1 \times rand(0,1) \times acc_{i-norm}^{t+1} \times d \times (x_{ri} - x_i^t)$$

ELSE

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}}$$
$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} - \min(acc)}{max(acc) - \min(acc)} + l$$
$$x_i^{t+1} = x_{best}^t + C2 \times rand(0,1) \times acc_{i-norm}^{t+1} \times d \times (x_{best} - x_i^t)$$
$$P = 2 \times rand(0,1) - C4$$
$$F = \begin{cases} +1, & P \le 0.5 \\ -1, & P > 0.5 \end{cases}$$

END IF

END FOR

Evaluate the population

Determine the best individual ($x_{best}, den_{best}, vol_{best}, acc_{best}$)

t=t+1 // Increase the iteration counter

END WHILE

Print the best fitness value

Figure 1. *The pseudocode of AOA.*

The computational complexity of the AOA is related to the population size (N) and the maximum iteration number (t_{max}). The Big-0 notation of the computational complexity of the AOA is O(AOA)=O(t_{max}×N). If we suppose that t_{max} and N are equals the computational complexity is O(AOA)= O(N^2). Generally, t_{max} is bigger than N. Thus, for calculation the in worst-case the most important part is t_{max}.

III. BINARY ARCHIMEDES OPTIMIZATION ALGORITHMS

A. TRANSFER FUNCTIONS

The mathematical formulas of transfer functions are given in Table 1. TF1, TF2, TF3, TF4, TF5, TF6, TF8, TF13, TF14, TF15, TF16 and TF17 are stationary transfer functions. TF7, TF9, TF10, TF11 and TF12 are non-stationary transfer functions. In Table 1, c means the continuous value, tv means transferred value, e means exponential function, tanh is hyperbolic tangent function, atan is inverse tangent in radians, erf is error function, $failure$ is the number of failures, $Qmax$ and $Qmin$ are predetermined values for non-stationary transfer functions, Max_iter is the maximum number of iteration, $iter$ is the current iteration number, sin is the sinus function, cos is the cosinus function, mod is the modulo function, $round$ is the round function, abs is the absolute value function, fix is the round towards zero function.

__Table 1.__ The mathematical formulas of transfer functions.

Name	Formula
TF1	$tv = \frac{1}{(1+e^{-c})}$
TF2	$tv = \frac{1}{(1+e^{-2c})}$
TF3	$tv = \frac{1}{\left(1+e^{-\frac{c}{2}}\right)}$
TF4	$tv = abs(\tanh(2c))$
TF5	$tv = abs\left(\frac{2}{\pi} \times \mathrm{atan}\left(\frac{\pi}{2} \times c\right)\right)$
TF6	$tv = \left(\mathrm{erf}\left(\frac{\sqrt{\pi}}{2} \times c\right)\right)$
TF7	$tv = \mathrm{erf}\left(\frac{failure}{Max\_iter}\right) + 1 - \mathrm{erf}\left(\frac{failure}{Max\_iter}\right) \times abs(\tanh(c))$
TF8	$tv = abs\left(\frac{c}{\sqrt{1+c^2}}\right)$
TF9	$tv = \frac{1}{\left(1+e^{\frac{-2c}{Qmax-iter\times((Qmax-Qmin)/Max\_iter)}}\right)}$
TF10	$tv = \frac{1}{\left(1+e^{\frac{-c}{Qmax-iter\times((Qmax-Qmin)/Max\_iter)}}\right)}$
TF11	$tv = \begin{cases} 1 - \frac{2}{1+e^{\frac{-2c}{Qmax-iter\times((Qmax-Qmin)/Max\_iter)}}}, & x \leq 0 \\ \frac{2}{1+e^{\frac{-2c}{Qmax-iter\times((Qmax-Qmin)/Max\_iter)}}} - 1, & x > 0 \end{cases}$
TF12	$tv = \begin{cases} 1 - \frac{2}{1+e^{\frac{-c}{2\times(Qmax-iter\times((Qmax-Qmin)/Max\_iter))}}}, & x \leq 0 \\ \frac{2}{1+e^{\frac{-c}{2\times(Qmax-iter\times((Qmax-Qmin)/Max\_iter))}}} - 1, & x > 0 \end{cases}$
TF13	$tv = (c-(-10))/(20)$
TF14	$tv = abs(\tanh(C))$
TF15	$tv = abs(sin(2\times\pi\times c\ \times cos(2\times\pi\times c\ )))$
TF16	$tv = mod(round(abs(mod(c,2))),2)$
TF17	$tv = fix(abs(mod(x,2)))$

B. BINARY VARIANTS OF ARCHIMEDES OPTIMIZATION ALGORITHM

The binary variants of the Archimedes optimization algorithm use the transfer functions located in Table 1. The relationship is depicted in Figure 2.

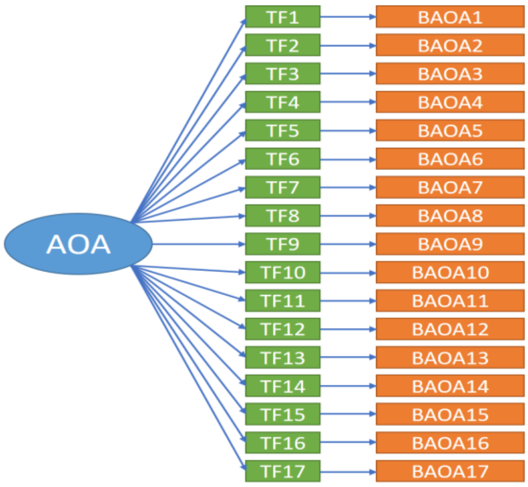

Figure 2. *The proposed binary variants of AOA.*

The transferred value is controlled at every iteration and if this value is smaller (or equal) than 0.5 the binary value is set as 0, and vice versa, if the transferred value is bigger than 0.5, the binary value is set as 1.

IV. EXPERIMENTAL SETUP

The mathematical model of UFLP is directly taken from [23]. The experiments were conducted with MATLAB. The range of normalization values (u and l) are set as 0.9 and 0.1, respectively. The peculiar parameters of AOA is directly taken from [12] as C1 = 2, C2 = 6, C3 = 2, and C4 = 0.5. The properties of UFL problems are given in Table 2. These problems are directly taken from OR-Library [25], the detailed information about problems and how optimum values reached can be found in [25]. The maximum iteration number is set as 2000. The population number is set as 40. 80000 function evaluation number is used in experiments. 30 different runs are conducted for comparisons.

Table 2. *The properties of UFL problems.*

Problem Name	Number of Facility	Number of Customer	Optimum Value
Cap71	16	50	932,651.80
Cap72	16	50	977,799.40
Cap73	16	50	1,010,641.00
Cap74	16	50	1,034,977.00
Cap101	25	50	796,648.40
Cap102	25	50	854,704.20
Cap103	25	50	893,782.10
Cap104	25	50	928,941.80
Cap131	50	50	793,439.60
Cap132	50	50	851,495.30
Cap133	50	50	893,076.70
Cap134	50	50	928,941.80
CapA	100	1000	17,156,454.48
CapB	100	1000	12,979,071.58
CapC	100	1000	11,505,594.33

Cap71, Cap72, Cap73 and Cap74 problems are small-sized (16 dimensions), Cap101, Cap102, Cap103 and Cap104 problems are medium-sized (25 dimensions), Cap131, Cap132, Cap133 and Cap134 problems are large-sized (50 dimensions), and CapA, CapB and CapC problems are huge-sized (100 dimensions). The values in the tables are GAP values and the values in the convergence figures are fitness function values. The GAP value is calculated as in Equation 1. GAP is not an abbreviation and it is a word with meanings of distance or space.

$$GAP(\%) = \frac{Obtained\ Mean\ Result\ in\ 30\ different\ runs - Optimum\ Value}{Optimum\ Value} \tag{1}$$

The best results are highlighted with **bold** font.

V. RESULTS AND DISCUSSION

The first experiment is related to determining the best transfer functions in terms of solution quality. The mean GAP results of 17 variants of BAOAs on UFL problems are located in Table 3. According to Table 3, BAOA9 has a minimum GAP value. Cap73, Cap74, Cap104 were solved optimally in 30 different runs by BAOA9. The second approach is BAOA10 in terms of GAP value. The third one is BAOA12. BAOA9 uses TF9, BAOA10 uses TF10 and BAOA12 uses TF12, these are all non-stationary transfer functions. According to total GAP values of problems, the easiest problem is Cap72 with a 0.41 GAP value and the hardest problem is CapA with 1509.22 GAP value. CapA is a 100-dimensional problem and BAOA12 had produced the best results in terms of GAP value, but BAOA9 produced better results for CapB and CapC. As a result, BAOA9 is the best binary variant of AOA for solving UFLPs.

Table 3. *The mean GAP results of 17 variants of BAOAs on UFL problems.*

	Cap71	Cap72	Cap73	Cap74	Cap101	Cap102	Cap103	Cap104	Cap131	Cap132	Cap133	Cap134	CapA	CapB	CapC	Total
BAOA1	0.25	**0.00**	**0.00**	**0.00**	1.08	0.02	**0.01**	**0.00**	0.83	0.62	0.32	0.34	50.51	21.32	17.53	92.84
BAOA2	0.33	**0.00**	0.01	**0.00**	1.15	**0.01**	0.02	0.02	0.26	0.21	**0.13**	**0.06**	15.60	8.40	8.20	34.39
BAOA3	0.05	**0.00**	**0.00**	**0.00**	0.88	0.09	0.06	0.02	1.93	2.40	2.55	3.99	125.23	56.81	39.77	233.78
BAOA4	**0.00**	0.01	0.01	0.13	0.12	0.39	0.69	1.77	4.57	5.91	5.67	13.45	59.82	41.64	34.53	168.70
BAOA5	0.05	0.15	0.10	0.34	0.91	0.91	1.09	2.23	5.64	6.91	9.44	13.78	154.23	71.17	47.43	314.40
BAOA6	**0.00**	**0.00**	**0.00**	0.01	**0.04**	0.10	0.16	0.40	3.27	4.39	6.24	11.45	75.14	42.50	30.67	174.39
BAOA7	**0.00**	0.01	0.03	0.20	0.18	0.57	0.77	1.87	5.38	8.24	12.19	20.08	196.07	99.95	75.52	421.06
BAOA8	0.00	**0.00**	0.02	0.09	0.17	0.37	0.56	0.95	4.01	6.37	8.54	12.52	109.18	55.56	32.75	231.11
BAOA9	0.44	0.03	**0.00**	**0.00**	1.06	0.03	0.02	**0.00**	**0.19**	**0.11**	0.14	0.18	6.47	**4.20**	**3.86**	**16.73**
BAOA10	0.53	**0.00**	0.01	**0.00**	0.86	**0.01**	0.02	**0.00**	0.28	0.12	0.14	0.14	8.93	6.25	5.38	22.67
BAOA11	**0.00**	0.03	0.21	0.97	0.36	1.38	2.97	5.13	6.28	9.53	9.20	22.18	96.46	72.46	52.94	280.11
BAOA12	0.03	0.08	0.01	0.01	0.71	0.95	0.46	0.05	3.69	2.55	1.31	0.74	**6.29**	7.00	7.45	31.34
BAOA13	0.14	**0.00**	**0.00**	0.02	0.90	0.57	0.34	0.56	4.15	4.97	5.93	10.15	203.23	88.40	64.61	383.97
BAOA14	**0.00**	**0.00**	**0.00**	0.01	0.10	0.14	0.32	0.70	3.39	5.21	7.81	11.46	102.71	48.57	43.40	223.82
BAOA15	0.46	0.05	0.05	0.21	0.45	0.70	0.98	2.14	4.59	6.76	9.86	15.87	247.69	112.92	83.30	486.03
BAOA16	0.90	0.03	0.01	0.02	0.88	0.20	0.11	0.18	1.00	0.94	0.89	0.77	26.25	13.01	12.17	57.36
BAOA17	0.69	0.01	0.02	**0.00**	0.56	0.25	0.12	0.11	0.65	0.66	0.57	0.69	25.40	10.66	10.88	51.28
Total	3.88	0.41	0.49	1.99	10.41	6.71	8.70	16.14	50.11	65.89	80.93	137.87	1509.22	760.83	570.39	

The convergence figures of BAOA variants in the CapA problem are located in Figure 3. BAOA9 (blue bold curve) has fast convergence and produces near-optimal results after 200 iterations. BAOA12 (light green tiny curve) is better than BAOA9 and BAOA10 (light orange tiny curve) is third in convergence analysis.

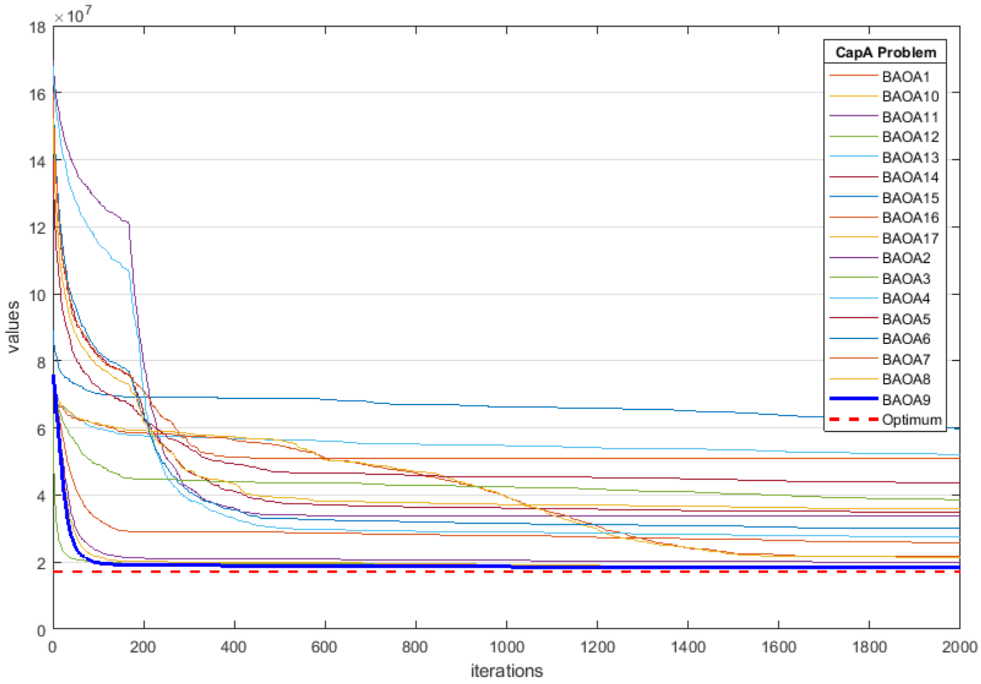

Figure 3. *The convergence figures of BAOA variants in the CapA problem.*

The convergence figures of BAOA variants in the CapB problem are located in Figure 4. BAOA9 (blue bold curve) has fast convergence and produces near-optimal results after 200 iterations. The convergence figures of BAOA variants in the CapC problem are located in Figure 5. BAOA9 (blue bold curve) has fast convergence and produces near-optimal results after 200 iterations

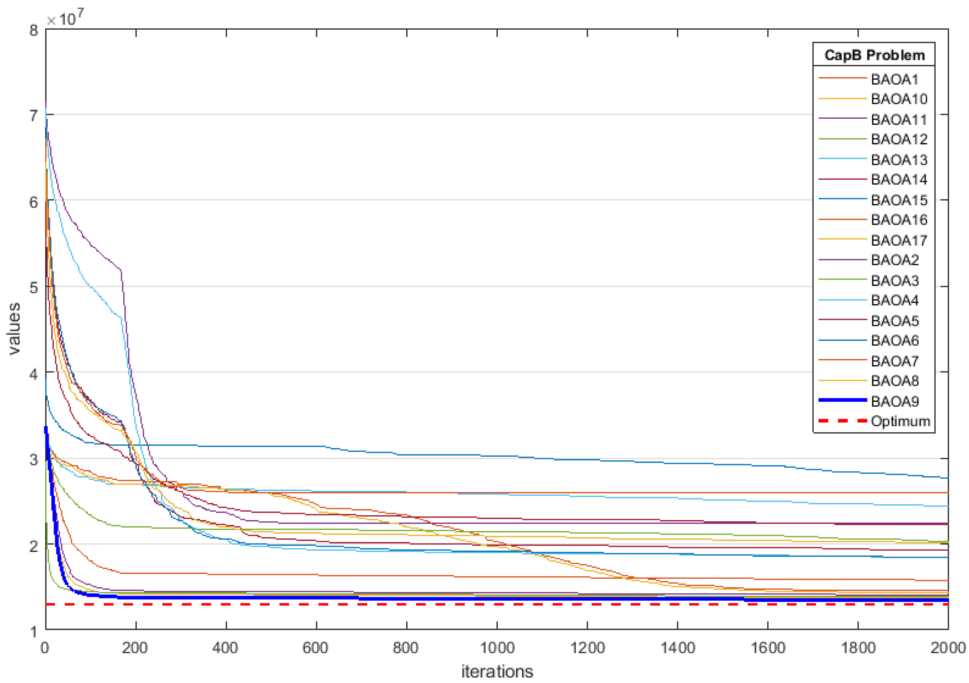

Figure 4. *The convergence figures of BAOA variants in the CapB problem.*

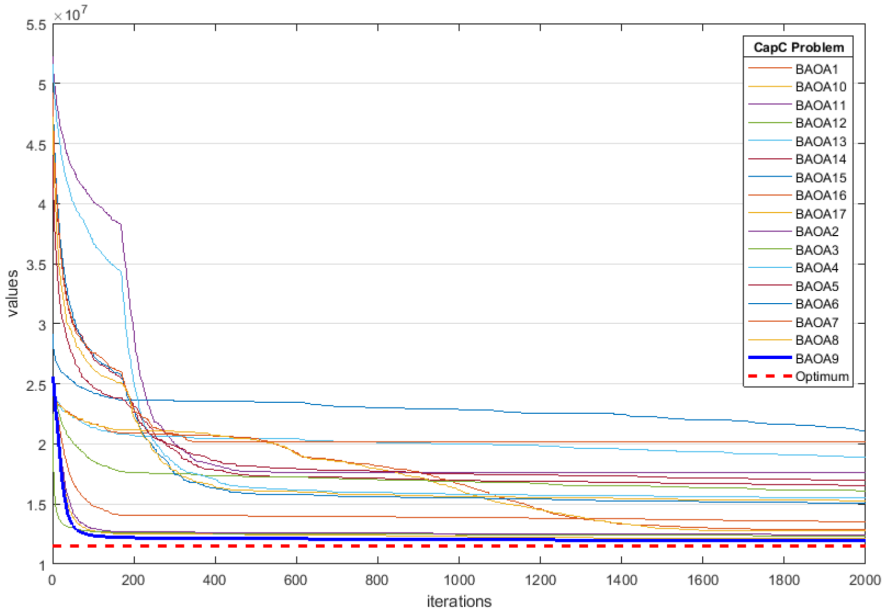

Figure 5. *The convergence figures of BAOA variants in the CapC problem.*

The second experiment is investigating the effects of peculiar parameters of AOA. For experiments, 24 different C1, C2, C3, and C4 combinations are used for comparisons. The mean GAP results for peculiar parameters of BAOA9 on CapA, CapB and CapC problems are given in Table 4. When C1=2, C2=6, C3=2, C4=0.5 are used the best results were produced. The best GAP results for peculiar parameters of BAOA9 on CapA, CapB and CapC problems are given in Table 5.

Table 4. *The mean GAP results for peculiar parameters of BAOA9 on CapA, CapB and CapC problems.*

No	C1	C2	C3	C4	CapA	CapB	CapC	Total	Rank
1	1	2	1	0.5	31.41	12.93	10.63	54.98	21
2	1	2	1	1	36.02	15.38	11.90	63.30	22
3	1	2	2	0.5	25.51	11.82	10.15	47.48	16
4	1	2	2	1	29.55	13.09	9.63	52.27	20
5	1	4	1	0.5	26.21	12.20	9.73	48.14	18
6	1	4	1	1	37.18	15.12	12.29	64.59	23
7	1	4	2	0.5	16.26	9.42	7.68	33.36	14
8	1	4	2	1	26.63	12.67	9.34	48.64	19
9	1	6	1	0.5	24.70	12.15	10.79	47.64	17
10	1	6	1	1	39.07	14.76	11.19	65.01	24
11	1	6	2	0.5	13.31	8.67	7.44	29.43	13
12	1	6	2	1	25.54	11.83	9.86	47.23	15
13	2	2	1	0.5	6.94	4.38	4.62	15.94	3
14	2	2	1	1	7.20	5.37	5.23	17.80	7
15	2	2	2	0.5	7.83	5.18	4.81	17.82	8
16	2	2	2	1	8.09	5.24	5.08	18.41	10
17	2	4	1	0.5	6.99	4.99	4.87	16.85	5
18	2	4	1	1	7.92	5.76	5.38	19.06	11
19	2	4	2	0.5	**6.53**	4.52	4.37	15.43	2
20	2	4	2	1	8.02	5.55	4.16	17.73	6
21	2	6	1	0.5	6.98	4.83	4.69	16.50	4
22	2	6	1	1	8.97	5.52	5.35	19.84	12
23	2	6	2	0.5	6.91	**4.27**	**4.03**	**15.21**	**1**
24	2	6	2	1	8.39	4.91	4.76	18.06	9

Table 5. *The best GAP results for peculiar parameters of BAOA9 on CapA, CapB and CapC problems.*

No	C1	C2	C3	C4	CapA	CapB	CapC	Total	Rank
1	1	2	1	0.5	9.87	6.14	3.46	19.48	22
2	1	2	1	1	6.57	5.78	3.57	15.92	18
3	1	2	2	0.5	6.00	5.34	5.19	16.53	20
4	1	2	2	1	9.81	6.51	4.63	20.94	23
5	1	4	1	0.5	6.95	4.87	3.91	15.73	17
6	1	4	1	1	5.13	5.27	6.13	16.53	19
7	1	4	2	0.5	4.96	2.92	3.41	11.29	15
8	1	4	2	1	8.57	4.30	4.53	17.40	21
9	1	6	1	0.5	8.33	3.58	2.90	14.80	16
10	1	6	1	1	15.89	8.12	4.83	28.84	24
11	1	6	2	0.5	5.48	2.55	1.37	9.41	13
12	1	6	2	1	2.97	3.45	4.84	11.25	14
13	2	2	1	0.5	0.14	1.67	2.90	4.71	7
14	2	2	1	1	2.52	2.43	2.19	7.14	12
15	2	2	2	0.5	0.14	0.71	1.56	**2.40**	1
16	2	2	2	1	1.58	**0.63**	1.98	4.18	6
17	2	4	1	0.5	1.58	2.46	1.74	5.77	9
18	2	4	1	1	1.64	2.12	2.54	6.30	10
19	2	4	2	0.5	0.68	1.74	1.14	3.56	4
20	2	4	2	1	**0.00**	1.54	1.76	3.31	3
21	2	6	1	0.5	0.14	2.01	**0.72**	2.87	2
22	2	6	1	1	2.47	2.54	1.65	6.66	11
23	2	6	2	0.5	1.54	1.56	0.89	3.99	5
24	2	6	2	1	2.21	1.42	1.93	5.57	8

When C1=2, C2=2, C3=2, C4=0.5 are used the best results were produced. Additionally, if C1=2, C2=4, C3=2, C4=1 are used CapA problem solved optimally. To obtain a general performance in binary optimization problems, the researchers can be used this parameter vector [C1=2, C2=2, C3=2, C4=0.5].

VI. CONCLUSION

Metaheuristic optimization algorithms are widely used for solving NP-Hard problems. Generally, metaheuristic algorithms are proposed for solving continuous optimization problems. However, in the real-world many optimization problems are discrete. Archimedes optimization algorithm (AOA) is a recently develop metaheuristic optimization algorithm and there is no binary variant of AOA. The uncapacitated facility location problem (UFLP) is a pure discrete binary optimization problem.

In this work, the first binarization process for AOA was conducted with 17 transfer functions. 17 new binary variants of AOA were proposed for solving 15 well-known UFLPs. The peculiar parameter analysis was conducted on 100-dimensional UFLPs. Stationary and non-stationary transfer functions were compared in terms of solution quality. The non-stationary transfer functions were produced better solutions than stationary transfer functions. Peculiar parameter analyzes for binary optimization problems were performed in the best variant (BAOA9) produced with the TF9 transfer function. The obtained results are supported by convergence figures.

In the future, new transfer functions for mapping the continuous search space values to binary search space values can be proposed and different binary optimization problems can be solved with binary variants of AOA.

ACKNOWLEDGEMENT: The authors wish to thank Scientific Research Projects Coordinatorship at Selcuk University and The Scientific and Technological Research Council of Turkey for their institutional supports.

V. REFERENCES

[1] T. Sağ, "Çok Merkezli Girdap Arama Algoritması," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi,* vol. 8, no. 2, pp. 1279-1294, 2020.

[2] M. S. Kiran, "TSA: Tree-seed algorithm for continuous optimization," *Expert Systems with Applications,* vol. 42, no. 19, pp. 6686-6698, 2015.

[3] I. Gungor, B. G. Emiroglu, A. C. Cinar and M. S. Kiran, "Integration search strategies in tree seed algorithm for high dimensional function optimization," *International Journal of Machine Learning and Cybernetics,* vol. 11, no. 2, pp. 249-267, 2020.

[4] S. A. Uymaz, G. Tezel and E. Yel, "Artificial algae algorithm (AAA) for nonlinear global optimization," *Applied Soft Computing,* vol. 31, pp. 153-171, 2015.

[5] O. FINDIK, "Bull optimization algorithm based on genetic operators for continuous optimization problems," *Turkish Journal of Electrical Engineering & Computer Sciences,* vol. 23, 2015.

[6] G. Yildizdan and Ö. K. Baykan, "A novel modified bat algorithm hybridizing by differential evolution algorithm," *Expert Systems with Applications,* vol. 141, p. 112949, 2020.

[7] H. T. KAHRAMAN, "Rulet Elektromanyetik Alan Optimizasyon (R-EFO) Algoritması," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi,* vol. 8, no. 1, pp. 69-80, 2020.

[8] E. Kaya, S. A. Uymaz and B. Kocer, "Boosting galactic swarm optimization with ABC," *International Journal of Machine Learning and Cybernetics,* vol. 10, no. 9, pp. 2401-2419, 2019.

[9] M. N. Demir and Y. Altun, "Otonom Araçla Genetik Algoritma Kullanılarak Haritalama ve Lokasyon," *Düzce Üniversitesi Bilim ve Teknoloji Dergisi,* vol. 8, no. 1, pp. 654-666, 2020.

[10] A. C. Cinar, "Training Feed-Forward Multi-Layer Perceptron Artificial Neural Networks with a Tree-Seed Algorithm," *Arabian Journal for Science and Engineering,* vol. 45, no. 12, pp. 10915-10938, 2020.

[11] M. E. Bayrakdar and A. Çalhan, "Artificial bee colony–based spectrum handoff algorithm in wireless cognitive radio networks," *International Journal of Communication Systems,* vol. 31, no. 5, p. e3495, 2018.

[12] F. A. Hashim, K. Hussain, E. H. Houssein, M. S. Mabrouk and W. Al-Atabany, "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence,* pp. 1-21, 2020.

[13] A. O. Dundar, M. A. Şahman, M. Tekin and M. S. Kıran, "A comparative application regarding the effects of traveling salesman problem on logistics costs," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 7, no. 4, pp. 207-2015, 2019.

[14] A. C. Cinar, S. Korkmaz and M. S. Kiran, "A discrete tree-seed algorithm for solving symmetric traveling salesman problem," *Engineering Science and Technology, an International Journal,* vol. 23, no. 4, pp. 879-890, 2020.

[15] H. Hakli and Z. Ortacay, "An improved scatter search algorithm for the uncapacitated facility location problem," *Computers & Industrial Engineering,* vol. 135, pp. 855-867, 2019.

[16] M. S. Kiran and M. Gündüz, "XOR-based artificial bee colony algorithm for binary optimization," *Turkish Journal of Electrical Engineering & Computer Sciences,* vol. 21, no. Sup. 2, pp. 2307-2328, 2013.

[17] Y. Çelikbilek, "Facility Location Selection Using Clustering Based Genetic Algorithm," *The Journal of International Scientific Researches,* vol. 5, no. 2, pp. 90-98, 2020.

[18] M. A. Sahman, A. A. Altun and A. O. Dündar, "The binary differential search algorithm approach for solving uncapacitated facility location problems," *Journal of Computational and Theoretical Nanoscience,* vol. 14, no. 1, pp. 670-684, 2017.

[19] M. A. Sahman and A. C. Cinar, "Binary tree-seed algorithms with S-shaped and V-shaped transfer functions," *International Journal of Intelligent Systems and Applications in Engineering,* vol. 7, no. 2, pp. 111-117, 2019.

[20] S. Korkmaz and M. S. Kiran, "An artificial algae algorithm with stigmergic behavior for binary optimization," *Applied Soft Computing,* vol. 64, pp. 627-640, 2018.

[21] S. Korkmaz, A. Babalik and M. S. Kiran, "An artificial algae algorithm for solving binary optimization problems," *International Journal of Machine Learning and Cybernetics,* vol. 9, no. 7, pp. 1233-1247, 2018.

[22] M. Aslan, M. Gunduz and M. S. Kiran, "JayaX: Jaya algorithm with xor operator for binary optimization," *Applied Soft Computing,* vol. 82, p. 105576, 2019.

[23] A. C. Cinar and M. S. Kiran, "Similarity and logic gate-based tree-seed algorithms for binary optimization," *Computers & Industrial Engineering,* vol. 115, pp. 631-646, 2018.

[24] C. Rorres, "Completing book II of Archimedes's on floating bodies," *The mathematical intelligencer,* vol. 26, no. 3, pp. 32-42, 2004.

[25] J. E. Beasley, "OR-Library: distributing test problems by electronic mail," *Journal of the operational research society,* vol. 41, no. 11, pp. 1069-1072, 1990.