Konya Mühendislik Bilimleri Dergisi, c. 9, s. 3, 713-734, 2021
Konya Journal of Engineering Sciences, v. 9, n. 3, 713-734, 2021
ISSN: 2667-8055 (Elektronik)
DOI: 10.36306/konjes.904335

HYBRID THE ARITHMETIC OPTIMIZATION ALGORITHM FOR CONSTRAINED OPTIMIZATION PROBLEMS

Emine BAŞ

Kulu Vocational School, Selçuk University, 42075, Konya, TURKEY
emineozcan@selcuk.edu.tr

(Geliş/Received: 27.03.2021; Kabul/Accepted in Revised Form: 29.07.2021)

ABSTRACT: Since many real-world problems can be designed as optimization problems, heuristic algorithms are increasingly preferred by researchers. The Arithmetic Optimization Algorithm (AOA) is a newly developed heuristic algorithm. It uses four arithmetic operations in its structure. The addition and subtraction operators enhanced the AOA's local search capability, while the multiplication and division operators enhanced the AOA's global search capability. It has been hybridized with the Tree Seed Algorithm (TSA) to increase the success of AOA. Thus, hybrid AOA-TSA (HAOA) has been proposed. The seed production mechanism of TSA is placed in the random walking stage of AOA. New candidate solutions (seeds) have been produced with the arithmetic operators involved in AOA and the candidate solutions have been compared with the existing solutions. Thus, the performance of AOA has increased. In this study, the success of AOA and HAOA was tested in thirteen constrained optimization problems. The success of AOA and HAOA has been tested for their performance in six different population sizes. The Wilcoxon Signed-Rank test was applied to the obtained results and its success has been proved statistically. The results proved the superiority of HAOA. HAOA has been compared with other heuristic methods in the literature and the success of HAOA has been shown. Additionally, AOA and HAOA have also been tested on three different engineering design problems. The results are discussed and evaluated.

Keywords: *Constrained optimization, Arithmetic optimization algorithm, Tree Seed Algorithm*

Kısıtlı Optimizasyon Problemleri İçin Hibrit Aritmetik Optimizasyon Algoritması

ÖZ: Pek çok gerçek dünya problemi optimizasyon problemleri olarak tasarlanabildiğinden, sezgisel algoritmalar araştırmacılar tarafından giderek daha fazla tercih edilmeye başlanmıştır. Aritmetik Optimizasyon Algoritması (AOA), yeni geliştirilmiş bir sezgisel algoritmadır. Yapısında dört aritmetik işlem kullanır. Toplama ve çıkarma operatörleri, AOA'nın yerel arama kabiliyetini geliştirirken, çarpma ve bölme operatörleri AOA'nın küresel arama kabiliyetini geliştirmiştir. AOA'nın başarısını artırmak için Ağaç Tohum Algoritması (TSA) ile hibritlenmiştir. Bu çalışmada, Hibrit AOA-TSA (HAOA) önerilmiştir. TSA'nın tohum üretim mekanizması, AOA'nın rastgele yürüme aşamasına yerleştirilmiştir. AOA'da yer alan aritmetik operatörler ile yeni aday çözümler (tohumlar) üretilmiş ve aday çözümler mevcut çözümlerle karşılaştırılmıştır. Böylece, AOA'nın performansı artmıştır. Bu çalışmada, AOA ve HAOA'nın başarısı on üç kısıtlı optimizasyon probleminde test edilmiştir. AOA ve HAOA'nın başarısı altı farklı popülasyon büyüklüğünde test edilmiştir. Elde edilen sonuçlara Wilcoxon Signed-Rank testi uygulanmış ve başarısı istatistiksel olarak kanıtlanmıştır. Sonuçlar HAOA'nın üstünlüğünü kanıtlamıştır. HAOA, literatürdeki diğer sezgisel yöntemlerle karşılaştırılmış ve HAOA'nın başarısı gösterilmiştir. Ek olarak, AOA ve HAOA, üç farklı mühendislik tasarım probleminde de test edilmiştir.

Anahtar Kelimeler: *Kısıtlı optimizasyon, Aritmetik optimizasyon algoritması, Ağaç tohum algoritması*

1. INTRODUCTION

The use of optimization in real-world problems is increasing in the literature. There are different types of optimization. For example, continuous and discrete optimization, constrained and unconstrained optimization, single and multi-objective optimization, etc (Kiran, 2015; Haklı, 2019). In recent years, many swarm intelligence-based algorithms have been improved to solve optimization problems in a short time. These algorithms can be applied directly to constrained optimization problems. While constrained optimization is being carried out, these algorithms need to check for violations. The various algorithms have been used to overcome constraints in the literature and a lot of algorithms based on swarm intelligence have been chosen to solve constrained optimization problems. Babalik et al. (2018) also solved the constrained optimization problem using the Tree Seed Algorithm (TSA) (Babalik et al., 2018). Aslan (2019) also proposed the new variant of the Elephant Herding Optimization (GL-EHO) algorithm and tested its performance on constrained optimization (Aslan, 2019). Xu et al. (2018) proposed a new approach by combining the Differential Evolution (DE) algorithm with an adaptive trial vector and solved constrained optimization problems (Xu et al., 2018). Lin (2013) hybridized the genetic algorithm with the rough set theory to evaluate constraints well and solved the constrained optimization problem (Lin, 2013). Apart from these, many swarm-based algorithms have been applied in the literature for the constrained optimization problem (Bansal et al, 2018; Garg, 2016; Runarsson and Yao, 2000; Kohli and Arora, 2017).

The Tree Seed Algorithm (TSA) is a population-based algorithm and was first proposed by Kiran (Kiran, 2015). There are two types (Tree and Seed) of solution sets in TSA. It was created by simulating the relationship between trees and their seeds. There are many studies on TSA in the literature. Babalik et al. (2018) tested the success of TSA on constrained optimization by using Deb's rules (Babalik et al., 2018). El-Fergany and Hasanien (2018) used TSA to solve the problem of the optimum power flow in large-scale power systems involving validations and comparisons (Fergany and Hasanien, 2018). Jiang et al. (2020) developed the TSA through a feedback mechanism to optimize continuous problems (Jiang et al., 2020). Beşkirli et al. (2019) examined a comparison of the modified tree seed algorithm for high-dimensional numerical functions (Beşkirli et al., 2019). Apart from these, there are many studies on TSA in the literature (Aslan et al., 2018; Cinar et al., 2020).

The Arithmetic Optimization Algorithm (AOA) was newly proposed by Abualigah et al. (Abualigah et al., 2021). Four arithmetic operations (Multiplication (M), Division (D), Subtraction (S), and Addition (A)) are used to improve the local and global search capability of the algorithm. The multiplication and division operations have been used to discover new points in the search space, and it has improved the exploration ability of the algorithm. Subtraction and Addition operations have been used to find local points in the search space, and it has improved the exploitation capability of the algorithm. Abualigah et al. tested the success of AOA on unimodal, multimodal, and hybrid composition benchmark functions.
Due to the new recommendation of AOA in the literature, AOA has not been studied yet. AOA's success on different problems has not yet been tested. In this study, its success was tested on constrained optimization problems. The success of AOA has been increased by hybridizing with TSA. Hybrid AOA (HAOA) has been developed by adapting the seed production mechanism implemented in TSA to AOA. The seed production mechanism of TSA is placed in the random walking stage of AOA. New candidate solutions (seeds) have been produced with the arithmetic operators involved in AOA and the candidate solutions have been compared with the existing solutions. Thus, the performance of AOA has increased. The success of the newly proposed HAOA on different problems has also been shown in the literature. HAOA has also been applied to various engineering design problems. There are various engineering design problems in the literature (Braik, 2021). The best known of these are the Welded Beam Design problem (WBD), the Pressure Vessel Design problem (PVD), and Compression Spring Design (CSD). AOA and HAOA have been tested in three different engineering design problems. The success of HAOA and AOA has been demonstrated.

The organization of the paper: the AOA, TSA, HAOA, and constrained optimization problems have been examined in Section 2. HAOA and AOA are tested on thirteen constrained benchmark problems for various population sizes in Section 3 and the results of HAOA were compared with AOA, ABC, DE, GA, PSO, EHO, EHO-NoB, HEHO, and GL-EHO on constrained optimization problems. In addition, AOA and HAOA have also been tested in three different engineering design problems. The results are discussed and evaluated.

2. MATERIAL AND METHOD

2.1. The Arithmetic Optimization Algorithm (AOA)

Abualigah et al. (2021) suggested a new meta-heuristic method called the Arithmetic Optimization Algorithm (AOA) for the first time (Abualigah et al., 2021). The basic structure of AOA consists of four main arithmetic operators used in mathematics (Multiplication (M), Division (D), Subtraction (S), and Addition (A)). These arithmetic operators formed the search mechanism of AOA in the search space. Addition and subtraction operators shaped the local search structure in AOA, while multiplication and division operators shaped the global search structure in AOA. Figure 1 shows a hierarchy of arithmetic operators.

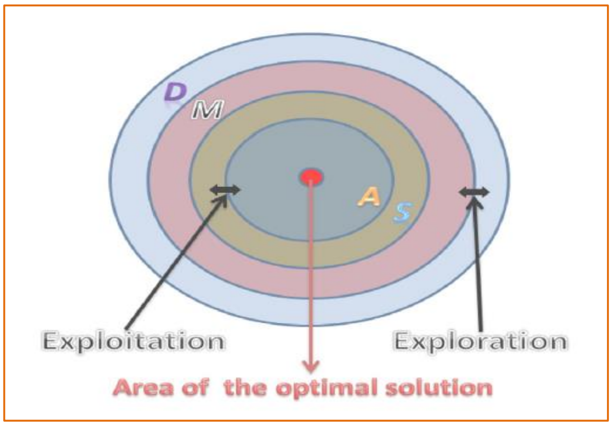

Figure 1. Hierarchy of arithmetic operators (Abualigah et al., 2021)

AOA chooses the exploration or exploitation phase at first. For this selection, the Math Optimizer Accelerated (MOA) function is calculated. Equation 1 shows the MOA function.

$$MOA(C\_Iter) = Min + C\_Iter \times \left(\frac{Max-Min}{Max\_Iter}\right) \tag{1}$$

where *MOA(C_Iter)* denotes the function value at the *t*th iteration, and *C _Iter* denotes the current iteration, and (*Max_ Iter*) denotes the maximum number of iterations. *Min* and *Max* denote the minimum and maximum values of the accelerated function, respectively.

2.1.1. Exploration phase

The exploration operators of AOA explore the search area randomly on several regions with Division (D) search strategy and Multiplication (M) search strategy and find a better solution. Equation 2 shows the exploration phase. This phase of searching (exploration search by executing D or M) is conditioned by the Math Optimizer accelerated (MOA) function for the condition of $r_1 > MOA$ (r_1 is a random number). Which of Division (D) search strategy or the Multiplication (M) search strategy to be used is determined by the value of r_2.

$$x_{i,j}(C\_Iter+1) = \begin{cases} best(x_j) \div (MOP+\epsilon) \times \left((UB_j - LB_j) \times \mu + LB_j\right), & r_2 < 0.5 \\ best(x_j) \times MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right), & otherwise \end{cases} \tag{2}$$

where r_2 is a random number. *best(x $_j$)* is the *j*th position in the best-obtained solution so far. ϵ is a small integer number, *UB$_j$* denotes the upper bound value of the *j*th position and *LB$_j$* denotes the lower bound value of the *j*th position μ is a control parameter to adjust the search process. Math Optimizer Probability (MOP) is shown by Equation 3.

$$MOP(C\_Iter) = 1 - \left(\frac{C\_Iter^{1/\alpha}}{Max\_Iter^{1/\alpha}}\right) \qquad (3)$$

where *MOP(C_Iter)* denotes the function value at the *t*th iteration, and *C _Iter* denotes the current iteration, and (*Max_ Iter*) denotes the maximum number of iterations. α is a sensitive parameter and defines the exploitation accuracy over the iterations (Abualigah et al., 2021).

2.1.2. Exploitation phase

The exploitation operators of AOA are carried out with the Addition (A) search strategy and Subtraction (S) search strategy. In AOA, AOA's exploitation operators search the search area in detail in several local regions. Equation 4 shows the exploitation phase. Which of the Subtraction (S) search strategy or the Addition (A) search strategy to be used is determined by the value of r3.

$$x_{i,j}(C\_Iter+1) = \begin{cases} best(x_j) - MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right), & r_3 < 0.5 \\ best(x_j) + MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right), & otherwise \end{cases} \qquad (4)$$

where r_3 is a random number. *best(x $_j$)* is the *j*th position in the best-obtained solution so far. ϵ is a small integer number, *UB$_j$* denotes the upper bound value of the *j*th position and *LB$_j$* denotes to lower bound value of the *j*th position μ is a control parameter to adjust the search process.

The Pseudo-code of the AOA has been explained in Algorithm 1. Figure 2 shows the flowchart of AOA.

Algorithm 1 Pseudo-code of the AOA

```
Assign parameter values (α, ??, ????) of AOA and initialize.
Initialize the solutions'positions (n) randomly. (Solutions: i=1, ..., N)
while (C_Iter < Max_Iter) do
  Calculate the objective function for the given solutions
  Find the best (best) solution
  Update the MOA and the MOP values with Eq. (1) and Eq. (3)
  for (i=1 to N) do
    for (j=1 to n) do
      Generate a random values between [0, 1] (r1, r2, and r3)
      if r1>MOA then
        if r2>0.5 then
          Update the ith solution positions using Eq. (2) ( Division (D) search strategy)
        else
          Update the ith solution positions using Eq. (2) (Multiplication (M) search strategy)
        end if
      else
        if r3>0.5 then
          Update the ith solution positions using Eq. (4) (Subtraction (S) search strategy)
        else
          Update the ith solution positions using Eq. (4) (Addition (A) search strategy)
        end if
      end if
    end for
  end for
  C_Iter=C_Iter+1
end while
The best solution
```

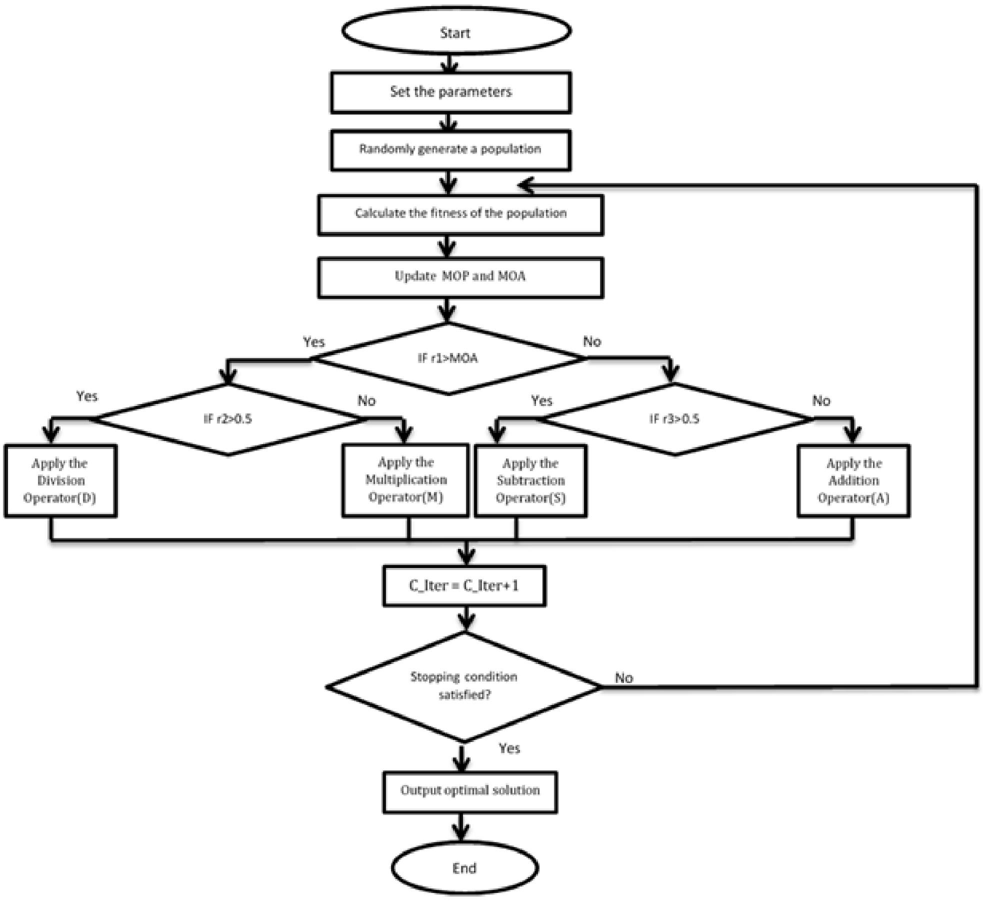
Start
Set the parameters
Randomly generate a population
Calculate the fitness of the population
Update MOP and MOA
Yes
IF r1>MOA
No
Yes
IF r2>0.5
No
Yes
IF r3>0.5
No
Apply the Division Operator(D)
Apply the Multiplication Operator(M)
Apply the Subtraction Operator(S)
Apply the Addition Operator(A)
C_Iter = C_Iter+1
Stopping condition satisfied?
No
Yes
Output optimal solution
End

Figure 2. The flowchart of AOA

2.2. The Tree Seed Algorithm (TSA)

The Tree Seed Algorithm (TSA) is a population-based algorithm and was first proposed by Kiran (Kiran, 2015). There are two types (Tree and Seed) of solution sets in TSA. It was created by simulating the relationship between trees and their seeds. In nature, trees spread their seeds and these seeds grow over time, eventually forming new trees. Considering the surface with these trees as the search space, the positions of the trees and seeds represent solutions for the optimization problem. Initial tree population formation in TSA is shown by Equation 5 (Babalık et al., 2018).

$$Tree_{i,j} = LB_j + r_{i,j}\left(UB_j - LB_j\right)\ \ i = 1, 2, \ldots, N\ and\ j = 1, 2, \ldots, n \tag{5}$$

where $Tree_{i,j}$ shows jth dimension of ith tree, $UB_j\ and\ LB_j$ are the upper and lower bound for the search space, N is the number of tree in the stand, n is the dimensionality of the optimization problem, and $r_{i,j}$ is a random number produced in a range of [0,1].

There are two different solution update equations in seed production. One of them uses the best seed location to improve the algorithm's exploitation capability, and the other uses a solution randomly selected from the population to explore different regions in the solution space. The equation to be used for seed production is decided by a control parameter called search tendency-ST (Babalık et al., 2018). Equation 6 shows the seed production mechanism.

$$Seed_{k,j} = \begin{cases} Tree_{i,j} + \alpha_{i,j} \times \left(best_j - Tree_{r,j}\right),\ \ r < ST \\ Tree_{i,j} + \alpha_{i,j} \times \left(Tree_{i,j} - Tree_{r,j}\right),\ otherwise \end{cases} \tag{6}$$

where ST is a pre-defined value in a range of [0,1], r is randomly produced in a range of [0,1], $\text{Tree}_{i,j}$ is a neighbor tree location randomly selected from the population. $\alpha_{i,j}$ is a scaling factor randomly produced in the range of [−1,1]. best_j is the jth position in the best-obtained solution so far. $\text{Seed}_{k,j}$ is the jth dimension of kth seed produced from ith tree.

The number of seeds for each tree is important for TSA's local exploration capability. Kiran (2015) analyzed the number of seeds for each tree in his study and suggested a number in a range of 10% and 25% of the number of trees in the stand.

The Pseudo-code of the TSA has been explained in Algorithm 2.

Algorithm 2 Pseudo-code of the TSA

```
Assign parameter values ($ST$, α, ?????) of TSA and initialize.
Initialize the trees'positions (n) randomly with Eq. (5). (Trees: i=1, ..., N)
Calculate the objective function for the given trees
Find the best (best) tree
while (C_Iter < Max_Iter) do
  for (i=1 to N) do
    Produce the seeds with Eq. (6).
    Calculate the objective function for the given seeds
    Find the best (best) seed
    Compare the best seed with the current tree
    If the best seed is better than the current tree, remove the current tree from the population and locate the best seed in the population.
  end for
  Find the best (best) tree
  C_Iter=C_Iter+1
end while
The best tree
```

2.3. Hybrid The Arithmetic Optimization Algorithm (HAOA)

Abualigah et al. (2021) suggested the Arithmetic Optimization Algorithm (AOA) for the first time (Abualigah et al., 2021). It performs local and global searches in the AOA search space by using four main arithmetic operators. These arithmetic operators are addition, subtraction, multiplication, and division. AOA prefers addition and subtraction strategies over exploring local points, whereas AOA prefers multiplication and division strategies over exploring global points. Kiran (2015) suggested the Tree Seed Algorithm (TSA) for the first time (Kiran, 2015). There are two types (Tree and Seed) of solution sets in TSA. It was created by simulating the relationship between trees and their seeds. In this study, TSA and AOA algorithms are hybridized. The seed production mechanism of TSA is placed in the random walking stage of AOA. New candidate solutions (seeds) have been produced with the arithmetic operators involved in AOA and the candidate solutions have been compared with the existing solutions. Thus, the performance of AOA has increased. Thus, the hybrid AOA-TSA (HAOA) has been proposed. In HAOA, the number of new candidate solutions (seeds) to be produced for each solution was found by Equations 7-9.

$$low = round(N * 0.1) \tag{7}$$

$$high = round(N * 0.25) \tag{8}$$

$$ns_i = round(low + (high - low) * r) + 1 \quad i = 1, 2, 3, \dots, N \tag{9}$$

where r is randomly produced in the range of [0,1], ns is the number of the candidate solutions (seeds) for each solution, and i is population size.

The Pseudo-code of the HAOA has been explained in Algorithm 3.

Algorithm 3 Pseudo-code of the HAOA

```
Assign parameter values ($\alpha$, ?, ????) of HAOA and initialize.
Initialize the solutions positions (n) randomly. (Solutions: i=1, ..., N)
while (C_Iter < Max_Iter) do
  Calculate the objective function for the given solutions
  Find the best (best) solution
  Update the MOA and the MOP values with Eq. (1) and Eq. (3)
  for (i=1 to N) do
    Calculate the number of candidate solutions (seeds) for each solution (ns) with Eqs.   (7-9)
    for (k=1 to ns) do
      for (j=1 to n) do
        Generate a random values between [0, 1] ($r_1$, $r_2$, and $r_3$)
        if $r_1$>MOA then
          if $r_2$>0.5 then
            Produce the candidate solutions (seeds) using Eq. (2) ( Division (D) search strategy)
          else
            Produce the candidate solutions (seeds) using Eq. (2) (Multiplication (M) search strategy)
          end if
        else
          if $r_3$>0.5 then
            Produce the candidate solutions (seeds) using Eq. (4) (Subtraction (S) search strategy)
          else
            Produce the candidate solutions (seeds) using Eq. (4) (Addition (A) search strategy)
          end if
        end if
      end for
    end for
    Compare the best candidate solution (seed) with the current solution
    If the best candidate solution (seed) is better than the current solution, remove the current solution from
    the population and locate the best candidate solution (seed) in the population.
  end for
  C_Iter=C_Iter+1
end while
The best solution
```

2.4. Constrained Optimization

Optimization problems search for the search space according to some constraints (Haklı, 2019). These constraints are shown in Equation 10.

$$\begin{aligned}&optimize\quad f(x)\\&subject\ to\\&g_i(x) \leq 0\ i = 1, 2, 3, \dots, q\\&h_i(x) = 0\ j = 1, 2, 3, \dots, p\end{aligned} \tag{10}$$

where f(x) is the fitness function of the problem, g(x) represents inequality constraint and h(x) is an equality constraint. q and p are respectively numbers of inequality constraints and equality constraints (Haklı, 2019). Equality constraints narrow the available search space. Therefore, it becomes difficult to find the most suitable solutions for optimization techniques (Haklı, 2019). In order to overcome this problem, equality constraints can be converted into inequality constraints (Strumberger et al., 2018):

$$\left|h_j(x)\right| \leq \varepsilon\ \ j = 1, 2, 3, \dots, p \tag{11}$$

where ε represents small violation tolerance. In constrained optimization, since the verification of the solution found depends on the violation of the constraints, the violation of the constraints is as important as the fitness value obtained from the objective function (Babalik et al., 2018).

3. EXPERIMENTAL ANALYSIS

The performances of AOA and Hybrid AOA (HAOA) have been tested on thirteen constrained problems (Haklı, 2019; Babalik et al., 2018; Runarsson and Yao, 2000). Descriptions of these problems are given in the appendix section of Babalik et al.'s study (Babalik et al., 2018). There are four maximization and nine minimization problems. G02, G03, G08, and G12 are maximization, and the G01, G04, G05, G06, G07, G09, G10, G11, and G13 are minimization problems. The dimension of each function is 13, 20, 10, 5, 4, 2, 10, 2, 7, 8, 2, 3, and 5, respectively. In this study, six different values were chosen for the size of the population {10, 20, 40, 60, 80, and 100} and the maximum number of function evaluations was determined as 2.4E+5 as in other studies (Babalik et al., 2018; Haklı, 2019; Ivana Strumberger et al., 2018). 30 independent runs were carried out for each function. The experiments are executed with a 2.3 GHz CPU and 4 GB RAM. The parameters setup of AOA and HAOA are shown in Table 1.

Table 1. Parameters setup for AOA and HAOA

Methods	Population size (N)	Maximum evaluation	μ	α	*Min*	*Max*	*r1, r2, and r3*
AOA	10, 20, 40, 60, 80, 100	2.4E+5	0.5	5	0.2	1	[0,1]
HAOA	10, 20, 40, 60, 80, 100	2.4E+5	0.5	5	0.2	1	[0,1]

3.1. Comparing AOA and HAOA algorithms for population size=10, 20, 40, 60, 80, and 100

In this subsection, the performance of AOA and HAOA in various population values has been evaluated for thirteen benchmark problems. The best, worst, mean, and standard deviation (SD) values were found for the obtained results. Table 2 shows AOA and HAOA comparison results for population size=10, 20, and 40. Table 3 shows AOA and HAOA comparison results for population size=60, 80, and 100. Table 4 shows HAOA comparison results for all the population sizes. Better results are marked in bold font. According to the results, the HAOA results exceeded the AOA results in all benchmark problems and achieved 100% success. According to Table 4, HAOA showed its best mean performance when the population size was selected as 60, 80, and 100 and HAOA showed its best SD performance when the population size was selected as 60 and 100.

A statistical test was performed on the results to show if there was a significant difference between AOA and HAOA results. Statistical test results are shown in Table 5. According to the results, if the h value is 1, there is a semantic difference in the results, and 0 indicates that there is no semantic difference in the results. The confidence interval of the Wilcoxon Signed-Rank test results of the AOA and HAOA algorithms is 0.05 in Table 5. The results showed that HAOA significantly improved the performance of AOA.

Figure 3 shows the convergence charts of AOA and HAOA for thirteen benchmark problems for population size=40. HAOA gave results that were more stable than AOA and did not show abrupt changes. HAOA converged to optimal results more quickly than AOA.

Table 2. Comparing AOA and HAOA algorithms for population size=10, 20, and 40

Problem	**Optimal**	**Min./Max.**		**AOA**			**HAOA**		
G1	**-15.000**	**Min.**		N=10	N=20	N=40	N=10	N=20	N=40
			Best	-8.9949	-10.2458	-11.7438	**-15.0000**	**-15.0000**	**-15.0000**
			Worst	-6.0000	-6.0000	-6.0000	-15.0000	-15.0000	-15.0000
			Mean	-7.5180	-8.2023807	-8.2022	**-15.0000**	**-15.0000**	**-15.0000**
			SD	1.1966	1.2291	1.4980	**4.25E-13**	1.04E-11	2.95E-11
G2	**0.803619**	**Max.**							
			Best	0.391954	0.409160	0.389537	**0.803777**	0.803774	0.803774
			Worst	0.238129	0.245952	0.255891	0.751332	0.786154	0.803765
			Mean	0.316291	0.314459	0.305839	0.787934	0.799721	**0.803771**
			SD	0.036056	0.041938	0.032353	0.015251	0.005497	**2.14E-06**
G3	**1.000**	**Max.**							
			Best	0.7487	0.5397	0.7807	**1.0000**	**1.0000**	**1.0000**
			Worst	0.2157	0.2821	0.3424	1.0000	1.0000	1.0000
			Mean	0.4277	0.4191	0.4464	**1.0000**	**1.0000**	**1.0000**
			SD	0.1107	0.0751	0.0920	**0.0000**	**0.0000**	**0.0000**
G4	**-30665.539**	**Min.**							
			Best	-30655.859	-30628.401	-30569.897	**-30665.539**	**-30665.539**	**-30665.539**
			Worst	-29353.432	-29413.147	-29440.351	-30665. 539	-30665.539	-30665.539
			Mean	-29702.704	-29864.681	-29769.803	**-30665.539**	**-30665.539**	**-30665.539**
			SD	295.935	376.664	304.024	3.50E-10	5.34E-11	**1.92E-11**
G5	**5126.498**	**Min.**							
			Best	5285.8941	5309.2067	5274.1580	5126.6811	**5126.6071**	5126.6281
			Worst	197072.6800	189380.3300	105649.540	5882.3934	5645.3743	5702.9671
			Mean	21161.6440	19757.8150	13678.9120	5230.8464	5210.8120	**5186.9092**
			SD	41158.5850	33847.7050	20748.0590	171.9845	126.0258	**114.3980**
G6	**-6961.814**	**Min.**							
			Best	-6811.4451	-6820.1903	-6717.4193	**-6961.8155**	**-6961.8155**	**-6961.8155**
			Worst	-5408.1825	-5371.1019	-4957.7713	-6961.8155	-6961.8155	-6961.8155
			Mean	-6034.3193	-6179.9272	-6153.6696	**-6961.8155**	**-6961.8155**	**-6961.8155**
			SD	382.1495	465.3159	448.1510	**0.0000**	**0.0000**	**0.0000**
G7	**24.306**	**Min.**							
			Best	100.7648	76.5262	233.0540	24.3470	**24.3346**	24.3392
			Worst	1969.0024	2455.7213	2898.0409	24.8349	24.5090	24.4809
			Mean	730.4350	837.0931	869.8616	24.5157	24.4131	**24.4034**
			SD	452.1383	585.5025	492.0641	0.1217	0.0497	**0.0346**
G8	**0.095825**	**Max.**							
			Best	0.095491	0.095685	0.095803	**0.095825**	**0.095825**	**0.095825**
			Worst	0.025357	0.025643	0.025797	0.029144	0.095825	0.095825
			Mean	0.085591	0.081791	0.085653	0.093602	**0.095825**	**0.095825**
			SD	0.019851	0.024683	0.020099	0.011970	**2.78E-17**	**2.78E-17**
G9	**680.63**	**Min.**							
			Best	722.7471	711.1926	692.4062	**680.6309**	680.6313	**680.6309**
			Worst	822.9434	823.2670	819.1971	680.6368	680.6348	680.6337
			Mean	783.3395	766.2433	742.4198	680.6334	680.6329	**680.6321**
			SD	29.70371	29.6735	32.2719	0.0016	0.0009	**0.0007**
G10	**7049.25**	**Min.**							
			Best	15121.503	15872.75	16315.177	7063.1916	7061.0968	**7059.6901**
			Worst	30000.342	30000.055	30000.181	7586.410	7367.1494	7192.8125
			Mean	28958.647	28878.606	28163.504	7208.5445	7129.5320	**7104.7907**
			SD	2991.3356	3303.9387	3769.9498	139.8415	62.2624	**36.7048**
G11	**0.75**	**Min**							
			Best	0.9997	0.9997	0.7627	0.7501	**0.7499**	**0.7499**
			Worst	0.9998	1.0002	1.0002	0.8950	0.9184	0.8778
			Mean	0.9997	0.9998	0.9842	0.7886	0.7777	**0.7727**
			SD	4.504E-05	9.604E-05	0.0505	0.0369	0.0389	**0.0320**
G12	**1.000**	**Max.**							
			Best	0.999998	0.999961	0.999992	**1.000**	**1.000**	**1.000**
			Worst	0.999735	0.999788	0.999713	1.000	1.000	1.000
			Mean	0.999916	0.999883	0.999907	**1.000**	**1.000**	**1.000**
			SD	6.505E-05	5.550E-05	6.796E-05	**0.000**	**0.000**	**0.000**
G13	**0.05395**	**Min.**							
			Best	0.65464	0.68307	**0.06604**	0.44009	0.46701	0.67033
			Worst	1.99980	7.12876	10.2887	3.48280	0.99999	1.15430
			Mean	1.22174	1.38483	1.34903	0.98898	**0.94725**	0.98781
			SD	0.40862	1.16457	1.77770	0.48450	0.12943	**0.06650**

Table 3. Comparing AOA and HAOA algorithms for population size=60, 80, and 100

Problem	**Optimal**	**Min./Max.**		**AOA**			**HAOA**		
				N=60	N=80	N=100	N=60	N=80	N=100
G1	**-15.000**	**Min.**							
			Best	-11.8171	-10.030321	-9.6961	**-15.0000**	**-15.0000**	**-15.0000**
			Worst	-6.0000	-6.0000	-6.0000	-15.0000	-15.0000	-15.0000
			Mean	-8.6623	-8.047555	-8.2317	**-15.0000**	**-15.0000**	**-15.0000**
			SD	1.1636	1.3695	1.2285	2.02E-11	7.55E-12	**8.28E-12**
G2	**0.803619**	**Max.**							
			Best	0.374180	0.460022	0.396971	0.803773	**0.803774**	**0.803774**
			Worst	0.246054	0.271129	0.263217	0.803768	0.803768	0.803768
			Mean	0.298437	0.323082	0.321634	0.803771	**0.803772**	0.803771
			SD	0.033894	0.048367	0.033699	**1.27E-06**	1.38E-06	1.48E-06
G3	**1.000**	**Max.**							
			Best	0.7746	0.6718	0.8850	**1.0000**	**1.0000**	**1.0000**
			Worst	0.2511	0.2083	0.2984	1.0000	1.0000	1.0000
			Mean	0.4373	0.4142	0.4310	**1.0000**	**1.0000**	**1.0000**
			SD	0.1134	0.0941	0.1125	**0.0000**	**0.0000**	**0.0000**
G4	**-30665.539**	**Min.**							
			Best	-30568.967	-30574.709	-30576.023	**-30665.540**	**-30665.540**	**-30665.540**
			Worst	-29415.253	-29466.224	-29440.550	-30665.540	-30665.540	-30665.540
			Mean	-29846.710	-29835.787	-29832.960	**-30665.540**	**-30665.540**	**-30665.540**
			SD	367.986	309.725	296.731	2.58E-11	1.92E-11	**7.28E-12**
G5	**5126.498**	**Min.**							
			Best	5271.0473	5376.0605	5300.0793	5126.5932	5126.5955	**5126.5322**
			Worst	53831.806	41811.412	131241.64	5415.7013	5549.4217	6050.0505
			Mean	10208.656	8932.1443	10948.742	**5166.3762**	5185.0307	5245.1527
			SD	10055.296	8325.3616	22446.066	**57.6578**	92.9843	185.3663
G6	**-6961.814**	**Min.**							
			Best	-6784.7484	-6933.5702	-6115.7646	**-6961.8155**	**-6961.8155**	**-6961.8155**
			Worst	-5113.3448	-4821.1231	-2353.9016	-6961.8155	-6961.8155	-6961.8155
			Mean	-6200.1311	-6143.5361	-5711.0374	**-6961.8155**	**-6961.8155**	**-6961.8155**
			SD	344.3004	440.9267	628.6112	**0.0000**	**0.0000**	**0.0000**
G7	**24.306**	**Min.**							
			Best	425.6965	436.5212	483.2933	24.3426	**24.3395**	24.3401
			Worst	1323.9415	2478.5446	2300.7003	24.4497	24.4470	24.4460
			Mean	904.5381	1003.9447	1015.6819	24.3963	**24.3868**	24.3953
			SD	286.0819	393.7003	408.9039	**0.02607**	0.02818	0.02716
G8	**0.095825**	**Max.**							
			Best	0.095701	0.095315	0.095768	**0.095825**	**0.095825**	**0.095825**
			Worst	0.025969	0.025748	0.025662	0.095825	0.095825	0.095825
			Mean	0.090408	0.085514	0.087666	**0.095825**	**0.095825**	**0.095825**
			SD	0.012172	0.019443	0.016457	**2.78E-17**	**2.78E-17**	**2.78E-17**
G9	**680.63**	**Min.**							
			Best	699.8415	686.9464	689.1064	**680.6306**	**680.6306**	680.6309
			Worst	813.3713	793.0415	832.6263	680.6330	680.6329	680.6329
			Mean	732.7981	730.0166	738.3557	680.6318	680.6319	**680.6317**
			SD	26.7901	24.1272	33.7390	0.0006	**0.0005**	**0.0005**
G10	**7049.25**	**Min.**							
			Best	11602.2480	17915.4610	23531.7840	7060.0143	7053.7270	**7049.8481**
			Worst	30000.0770	30000.0580	30000.1500	7187.5255	7156.2179	7179.2562
			Mean	28149.1710	28759.9030	29232.6030	7099.2976	7100.7723	**7093.6483**
			SD	4484.2020	3021.6864	1637.2815	**27.3408**	28.0190	27.5670
G11	**0.75**	**Min**							
			Best	0.9997	0.7553	0.8603	**0.7499**	**0.7499**	**0.7499**
			Worst	1.0002	1.0002	1.0002	0.7922	0.7938	0.7755
			Mean	0.9998	0.9891	0.9826	0.7554	0.7552	**0.7528**
			SD	**0.0002**	0.0457	0.0442	0.0084	0.0095	0.0054
G12	**1.000**	**Max.**							
			Best	0.999992	0.999995	0.999996	**1.000**	**1.000**	**1.000**
			Worst	0.999785	0.99971	0.999817	1.000	1.000	1.000
			Mean	0.999899	0.99990	0.999916	**1.000**	**1.000**	**1.000**
			SD	5.48E-05	7.67E-05	4.98E-05	**0.000**	**0.000**	**0.000**
G13	**0.05395**	**Min.**							
			Best	0.478609	**0.146282**	0.169063	0.865197	0.481268	0.935540
			Worst	7.884131	7.214411	9.650241	1.391072	1.495773	0.999997
			Mean	1.365598	1.463003	1.471600	1.003210	**0.982606**	0.995394
			SD	1.288158	1.500493	1.642668	0.076274	0.150232	**0.012438**

Table 4. Comparing the variations of HAOA algorithm for population size=10, 20, 40, 60, 80, and 100

ID	N=10		N=20		N=40		N=60		N=80		N=100	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD	Mean	SD
G1	**-15.0000**	**4.25E-13**	**-15.0000**	1.04E-11	**-15.0000**	2.95E-11	**-15.0000**	2.02E-11	**-15.0000**	7.55E-12	**-15.0000**	8.28E-12
G2	0.787934	0.015251	0.799721	0.005497	0.803771	2.14E-06	0.803771	**1.27E-06**	**0.803772**	1.38E-06	0.803771	1.48E-06
G3	**1.0000**	**0.0000**	**1.0000**	**0.0000**	**1.0000**	**0.0000**	**1.0000**	**0.0000**	**1.0000**	**0.0000**	**1.0000**	**0.0000**
G4	**-30665.539**	3.50E-10	**-30665.539**	5.34E-11	**-30665.539**	1.92E-11	**-30665.539**	2.58E-11	**-30665.539**	1.92E-11	**-30665.539**	**7.28E-12**
G5	5230.8464	171.9845	5210.8120	126.0258	5186.9092	114.3980	**5166.3762**	**57.6578**	5185.0307	92.9843	5245.1527	185.3663
G6	**-6961.8155**	**0.0000**	**-6961.8155**	**0.0000**	**-6961.8155**	**0.0000**	**-6961.8155**	**0.0000**	**-6961.8155**	**0.0000**	**-6961.8155**	**0.0000**
G7	24.5157	0.1217	24.4131	0.0497	24.4034	0.0346	24.3963	**0.02607**	**24.3868**	0.02818	24.3953	0.02716
G8	0.093602	0.011970	**0.095825**	**2.78E-17**	**0.095825**	**2.78E-17**	**0.095825**	**2.78E-17**	**0.095825**	**2.78E-17**	**0.095825**	**2.78E-17**
G9	680.6334	0.0016	680.6329	0.0009	680.6321	0.0007	680.6318	0.0006	680.6319	**0.0005**	**680.6317**	**0.0005**
G10	7208.5445	139.8415	7129.5320	62.2624	7104.7907	36.7048	7099.2976	**27.3408**	7100.7723	28.0190	**7093.6483**	27.5670
G11	0.7886	0.0369	0.7777	0.0389	0.7727	0.0320	0.7554	0.0084	0.7552	0.0095	**0.7528**	**0.0054**
G12	**1.000**	**0.0000**	**1.000**	**0.0000**	**1.000**	**0.0000**	**1.000**	**0.0000**	**1.000**	**0.0000**	**1.000**	**0.0000**
G13	0.98898	0.48450	**0.94725**	0.12943	0.98781	0.06650	1.00321	0.07627	0.98261	0.15023	0.99539	**0.01244**

Table 5. The results of Wilcoxon Signed-Rank Test on the results of HAOA and AOA algorithms for population size=10, 20, 40, 60, 80, and 100

ID	HAOA-AOA											
	N=10		N=20		N=40		N=60		N=80		N=100	
	P-Value	h	P-Value	h	P-Value	h	P-Value	h	P-Value	h	P-Value	h
G1	1.61e-06	1	1.73E-06	1	1.67E-06	1	1.73E-06	1	1.64E-06	1	1.71E-06	1
G2	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G3	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G4	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G5	2.88E-06	1	1.92E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G6	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G7	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G8	2.37E-05	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G9	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G10	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G11	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.92E-06	1	1.73E-06	1
G12	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G13	0.0003589	1	9.71E-05	1	0.3285711	0	0.0027653	1	0.0544625	0	0.0024147	1

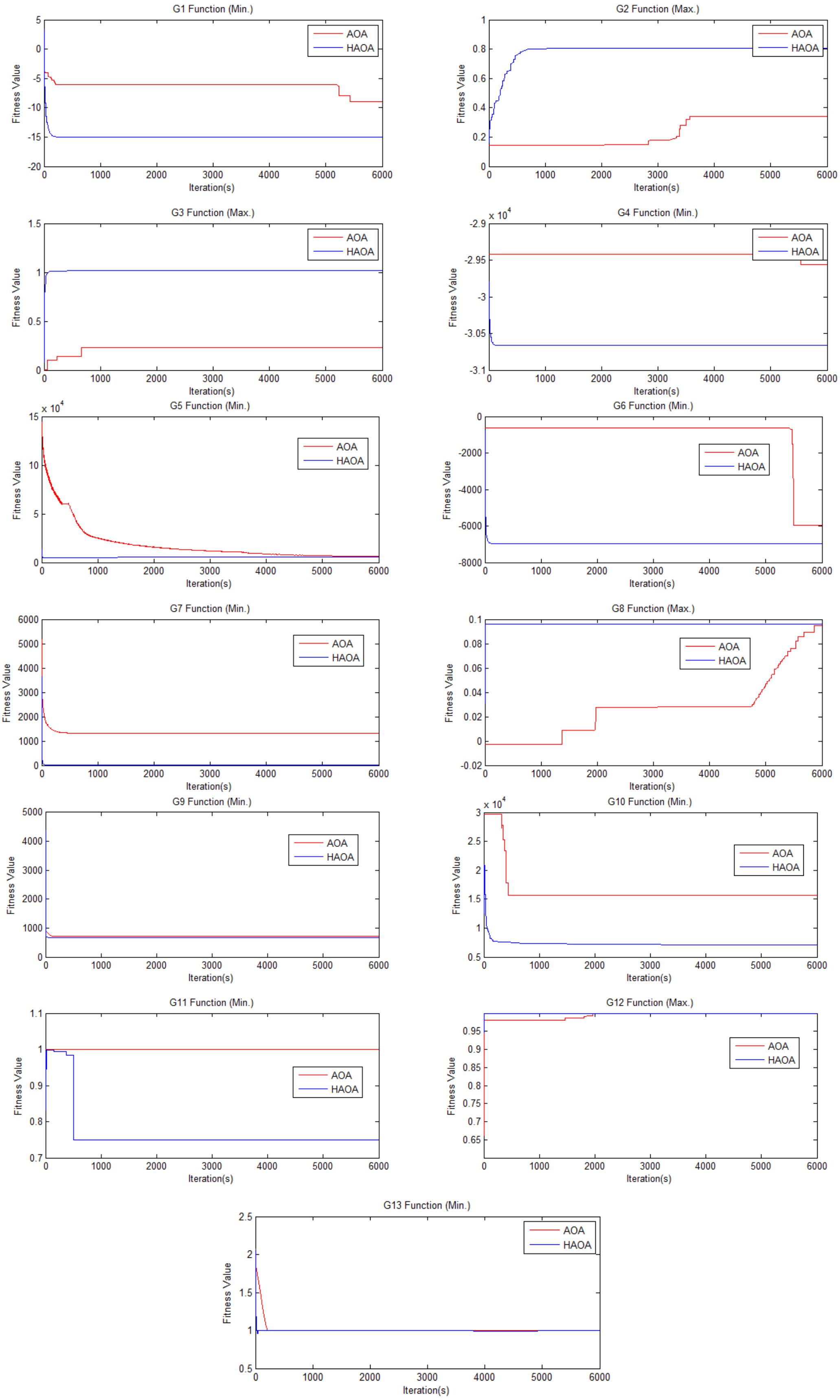

Figure 3. The convergence charts of AOA and HAOA for thirteen benchmark problems

3.2. A comparison of HAOA with other algorithms

In the experiments, it has been proven that the success of HAOA exceeds the success of AOA. In this subsection, the success of HAOA has been compared with different methods in the literature. The HAOA is compared with Artificial Bee Colony (ABC), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Differential Evolution (DE), Elephant Herding Optimization (EHO), Elephant Herding Optimization with Global and Local search (GL-EHO), Hybridized EHO (HEHO), EHO-NoB, and Tree Seed Algorithm (CTSA). The experimental results are shown in Tables 7 - 9 (Babalik et al., 2018; Haklı, 2019). 30 independent runs were carried out for each function for all the algorithms except GA. The selected parameters for the comparison algorithms are shown in Table 6.

Table 6. Parameters setup for HAOA and other algorithms

Methods	Population size (N)	Maximum evaluation	Other parameters
HAOA	40	2.4E+5	*μ =0.5; α=5; Min=0.2; Max=1; r1, r2, and r3=[0,1]*
EHO	50	2.4E+5	*the number of elephant in each clan is set 10; α = 0.5; β= 0.1; c1 and c2 = 1.5*
GL-EHO	50	2.4E+5	*the number of elephant in each clan is set 10; α = 0.5; β= 0.1; c1 and c2 = 1.5*
HEHO	50	2.4E+5	*the number of elephant in each clan is set 10; α = 0.5; β= 0.1; c1 and c2 = 1.5*
EHO-NoB	50	2.4E+5	*the number of elephant in each clan is set 10; α = 0.5; β= 0.1; c1 and c2 = 1.5*
ABC	40	2.4E+5	-
PSO	40	2.4E+5	-
GA	40	2.4E+5	-
DE	40	2.4E+5	-
CTSA	40	2.4E+5	*ST=0.1*

According to Table 7, HAOA performed better than other algorithms (ABC, DE, GA, and PSO) in 8 out of 13 benchmark problems (G1, G2, G3, G4, G6, G8, G10, and G12). According to Table 8, HAOA performed better than other algorithms (GL-EHO, EHO, HEHO, and EHO-NoB) in 8 out of 13 benchmark problems (G1, G2, G3, G4, G8, G9, G10, and G12). After HAOA, HEHO has been the most successful algorithm (in 7 out of 13 benchmark problems). According to Table 9, HAOA performed better than other algorithms (AOA and CTSA) in 12 out of 13 benchmark problems (G1, G2, G3, G4, G5, G6, G7, G8, G9, G10, G11, and G12). CTSA performed well in 7 out of 13 benchmark problems (G1, G3, G4, G6, G8, G12, and G13).

A statistical test was performed on the results to show if there was a significant difference between HAOA, ABC, PSO, GA, DE, EHO, GL-EHO, HEHO, EHO-NoB, and CTSA results. Statistical test results are shown in Table 10 and Table 11. According to the results, if the h value is 1, there is a semantic difference in the results, and 0 indicates that there is no semantic difference in the results. The confidence interval of the Wilcoxon Signed-Rank test results of the HAOA and other algorithms is 0.05 in Table 10 and Table 11. The results showed that there is a significant difference between HAOA and the results of other comparison algorithms.

Figure 4 shows the convergence charts of AOA, HAOA, EHO, GL-EHO, HEHO, and EHO-NoB for five benchmark problems. Figure 5 shows the convergence charts of AOA, HAOA, ABC, PSO, DE, and CTSA for five benchmark problems. The convergence results have proven the success of HAOA.

Table 7. Comparing HAOA and other algorithms (ABC, PSO, GA, and DE)

Problem	**Optimal**		**ABC**	**PSO**	**GA**	**DE**	**HAOA**
G1	**-15.000**	**Mean**	-15.000	-10.5551	-14.236	-14.2406	**-15.0000**
		Difference	0.0205	4.4449	0.7640	0.7594	**0.0000**
G2	**0.803619**	**Mean**	0.4795	0.4043	0.7886	0.6660	**0.803771**
		Difference	0.3241	0.3993	0.0150	0.1376	**0.00015**
G3	**1.000**	**Mean**	3.0191	1.1675	0.9760	1.1694	**1.0000**
		Difference	2.0191	0.1675	-0.0240	0.1694	**0.0000**
G4	**-30665.539**	**Mean**	-30610.974	-30661.740	-30590.455	-30665.540	**-30665.539**
		Difference	54.565	3.799	75.084	0.001	**0.0000**
G5	**5126.498**	**Mean**	**5115.056**	5298.284	N/A	5329.197	5186.9092
		Difference	**11.441**	171.787	N/A	202.700	60.4112
G6	**-6961.814**	**Mean**	-7579.630	-6961.819	-6872.204	-6765.482	**-6961.8155**
		Difference	617.816	0.005	89.610	196.332	**0.0015**
G7	**24.306**	**Mean**	29.0956	28.7418	34.9800	**24.3160**	24.4034
		Difference	4.7896	4.4358	10.6740	**0.0100**	0.0974
G8	**0.095825**	**Mean**	6.5347	0.0847	**0.0958**	**0.0958**	**0.095825**
		Difference	6.4389	0.0111	**0.0000**	**0.0000**	**0.0000**
G9	**680.63**	**Mean**	683.8941	680.7815	692.0640	**680.6308**	680.6321
		Difference	3.2641	0.1515	11.4340	**0.0008**	0.0021
G10	**7049.25**	**Mean**	7259.028	8128.793	10003.225	7162.592	**7104.7907**
		Difference	209.778	1079.543	2953.975	113.342	**55.5407**
G11	**0.75**	**Mean**	0.7171	0.7626	**0.7500**	0.9545	0.7727
		Difference	0.0329	0.0126	**0.0000**	0.2045	0.0227
G12	**1.000**	**Mean**	1.0001	**1.0000**	**1.0000**	**1.0000**	**1.0000**
		Difference	0.0001	**0.0000**	**0.0000**	**0.0000**	**0.0000**
G13	**0.05395**	**Mean**	**0.0955**	1.4228	N/A	0.9492	0.98781
		Difference	**0.04155**	1.3273	N/A	0.8537	0.93386

Table 8. Comparing HAOA and other algorithms (EHO, EHO-NoB, HEHO, and GL-EHO)

Problem	Optimal		EHO	EHO-NoB	HEHO	GL-EHO	HAOA
G1	**-15.000**	**Mean**	-1.088	-14.500	-14.958	**-15.0000**	**-15.0000**
		Difference	-	-	-14.958	**0.0000**	**0.0000**
G2	**0.803619**	**Mean**	0.2522	0.4490	0.799125	0.6405	**0.803771**
		Difference	-	-	0.026	0.1631	**0.00015**
G3	**1.000**	**Mean**	0.6560	0.4864	**1.000**	0.9026	**1.0000**
		Difference	-	-	**0.000**	-0.0974	**0.0000**
G4	**-30665.539**	**Mean**	30333.809	-30304.074	-30499.033	-30665.540	**-30665.539**
		Difference	-	-	16.302	0.001	**0.0000**
G5	**5126.498**	**Mean**	5373.189	5182.527	**5126.505**	5502.522	5186.9092
		Difference	-	-	**0.041**	376.025	60.4112
G6	**-6961.814**	**Mean**	-6943.713	-6227.937	-6957.361	**-6961.814**	-6961.8155
		Difference	-	-	1.005	0.002	**0.0015**
G7	**24.306**	**Mean**	446.6258	83.0228	**24.309**	36.9279	24.4034
		Difference	-	-	**0.003**	12.6219	0.0974
G8	**0.095825**	**Mean**	0.095376	**0.095825**	**0.095825**	**0.0958**	**0.095825**
		Difference	-	-	**0.000**	**0.0000**	**0.0000**
G9	**680.63**	**Mean**	927.874	709.6519	680.653	681.7680	**680.6321**
		Difference	-	-	0.011	1.1380	**0.0021**
G10	**7049.25**	**Mean**	10236.025	8162.372	7152.895	8374.642	**7104.7907**
		Difference	-	-	95.239	1325.392	**55.5407**
G11	**0.75**	**Mean**	**0.7400**	0.7399	**0.751**	0.7399	0.7727
		Difference	-	-	**0.001**	0.0101	0.0227
G12	**1.000**	**Mean**	**1.000**	**1.000**	**1.000**	**1.0000**	**1.0000**
		Difference	-	-	**0.000**	**0.0000**	**0.0000**
G13	**0.05395**	**Mean**	1.3335	1.0946	**0.246**	0.4043	0.98781
		Difference	-	-	**0.106**	0.3088	0.93386

Table 9. Comparing HAOA and other algorithms (AOA and CTSA)

Problem	Optimal		CTSA	AOA	HAOA
G1	**-15.000**	**Mean**	**-15.0000**	-8.2022	**-15.0000**
		SD	**0.0000**	1.4980	2.95E-11
G2	**0.803619**	**Mean**	0.801098	0.305839	**0.803771**
		SD	0.003500	0.032353	**2.14E-06**
G3	**1.000**	**Mean**	**1.0000**	0.4464	**1.0000**
		SD	**0.0000**	0.0920	**0.0000**
G4	**-30665.539**	**Mean**	**-30665.539**	-29769.803	**-30665.539**
		SD	**0.0000**	304.024	1.92E-11
G5	**5126.498**	**Mean**	5195.827	13678.9120	**5186.9092**
		SD	121.868	20748.0590	**114.3980**
G6	**-6961.814**	**Mean**	**−6961.816**	-6153.6696	**-6961.816**
		SD	**0.0000**	448.1510	**0.0000**
G7	**24.306**	**Mean**	24.488	869.8616	**24.4034**
		SD	0.083	492.0641	**0.0346**
G8	**0.095825**	**Mean**	**0.095825**	0.085653	**0.095825**
		SD	**0.0000**	0.020099	2.78E-17
G9	**680.63**	**Mean**	680.642	742.4198	**680.6321**
		SD	0.004	32.2719	**0.0007**
G10	**7049.25**	**Mean**	7131.794	28163.504	**7104.7907**
		SD	71.905	3769.9498	**36.7048**
G11	**0.75**	**Mean**	0.806	0.9842	**0.7727**
		SD	0.055	0.0505	**0.0320**
G12	**1.000**	**Mean**	**1.0000**	0.999907	**1.0000**
		SD	**0.0000**	6.796E-05	**0.0000**
G13	**0.05395**	**Mean**	**0.9876**	1.34903	0.98781
		SD	0.2816	1.77770	**0.06650**

Table 10. The results of the Wilcoxon Signed-Rank Test on the results of HAOA, ABC, PSO, GA, DE, and CTSA algorithms

ID	HAOA									
	ABC		PSO		GA		DE		CTSA	
	P-Value	h	P-Value	h	P-Value	h	P-Value	h	P-Value	h
G1	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G2	1.73E-06	1	1.73E-06	1	1.92E-06	1	1.73E-06	1	3.82E-01	0
G3	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G4	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G5	1.73E-06	1	1.73E-06	1	N/A	-	1.73E-06	1	1.73E-06	1
G6	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G7	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G8	6.80E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G9	1.72E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G10	1.72E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G11	1.73E-06	1	0.24519	0	1.92E-06	1	1.73E-06	1	1.20E-03	1
G12	4.32E-08	1	1	0	1	0	1	0	1	0
G13	1.73E-06	1	1.73E-06	1	N/A	-	1.57E-02	1	3.39E-01	0

Table 11. The results of the Wilcoxon Signed-Rank Test on the results of HAOA, EHO, GL-EHO, HEHO, and EHO-NoB algorithms

ID	HAOA							
	EHO		GL-EHO		HEHO		EHO-NoB	
	P-Value	h	P-Value	h	P-Value	h	P-Value	h
G1	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G2	1.73E-06	1	1.73E-06	1	4.66E-01	0	1.73E-06	1
G3	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G4	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G5	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G6	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G7	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G8	4.32E-08	1	4.32E-08	1	4.32E-08	1	4.32E-08	1
G9	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G10	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
G11	1.73E-06	1	1.73E-06	1	9.32E-06	1	1.73E-06	1
G12	1	0	1	0	1	0	1	0
G13	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1

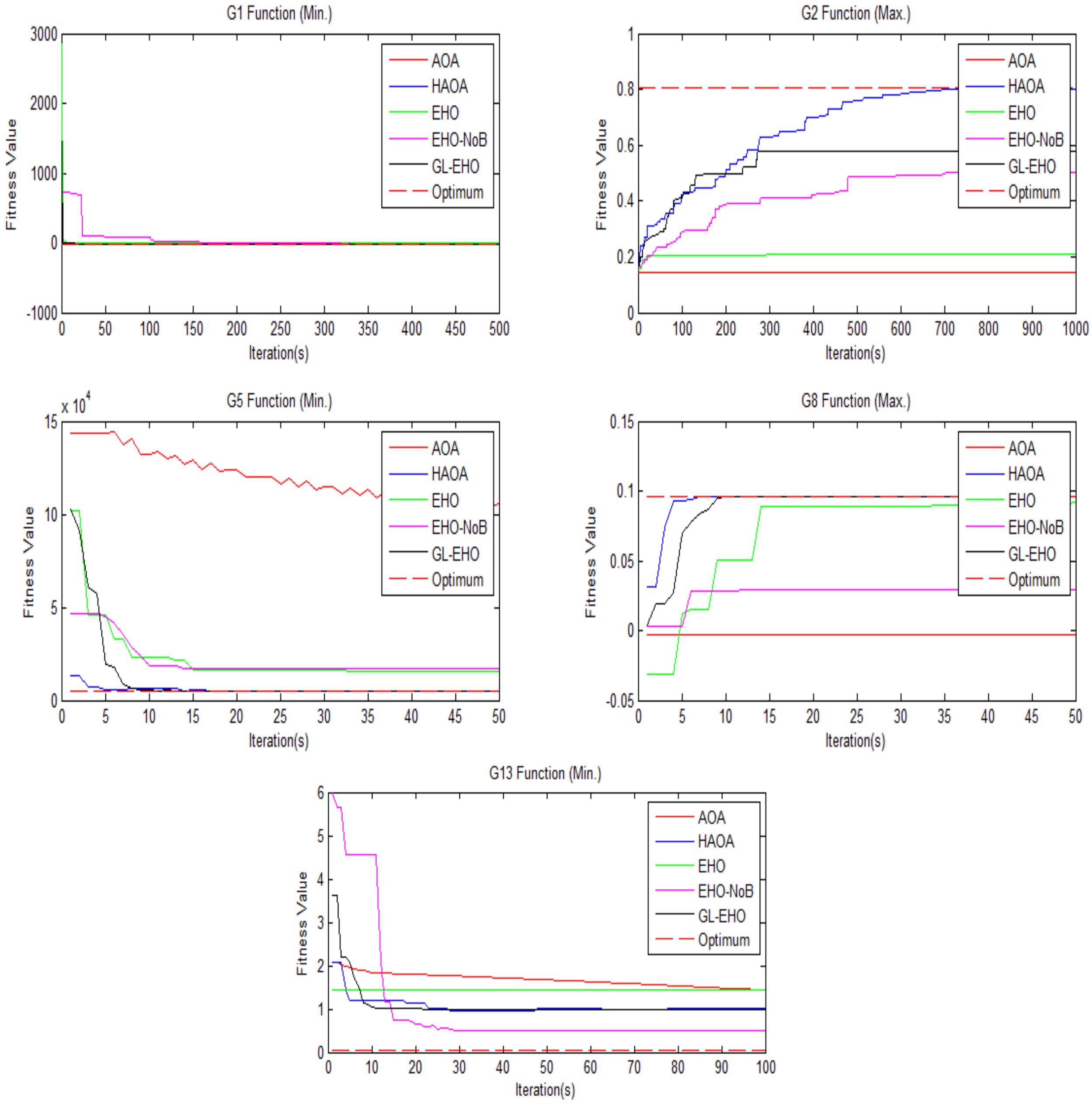

Figure 4. The convergence charts of HAOA, AOA, EHO, EHO-NoB, and GL-EHO for five benchmark problems

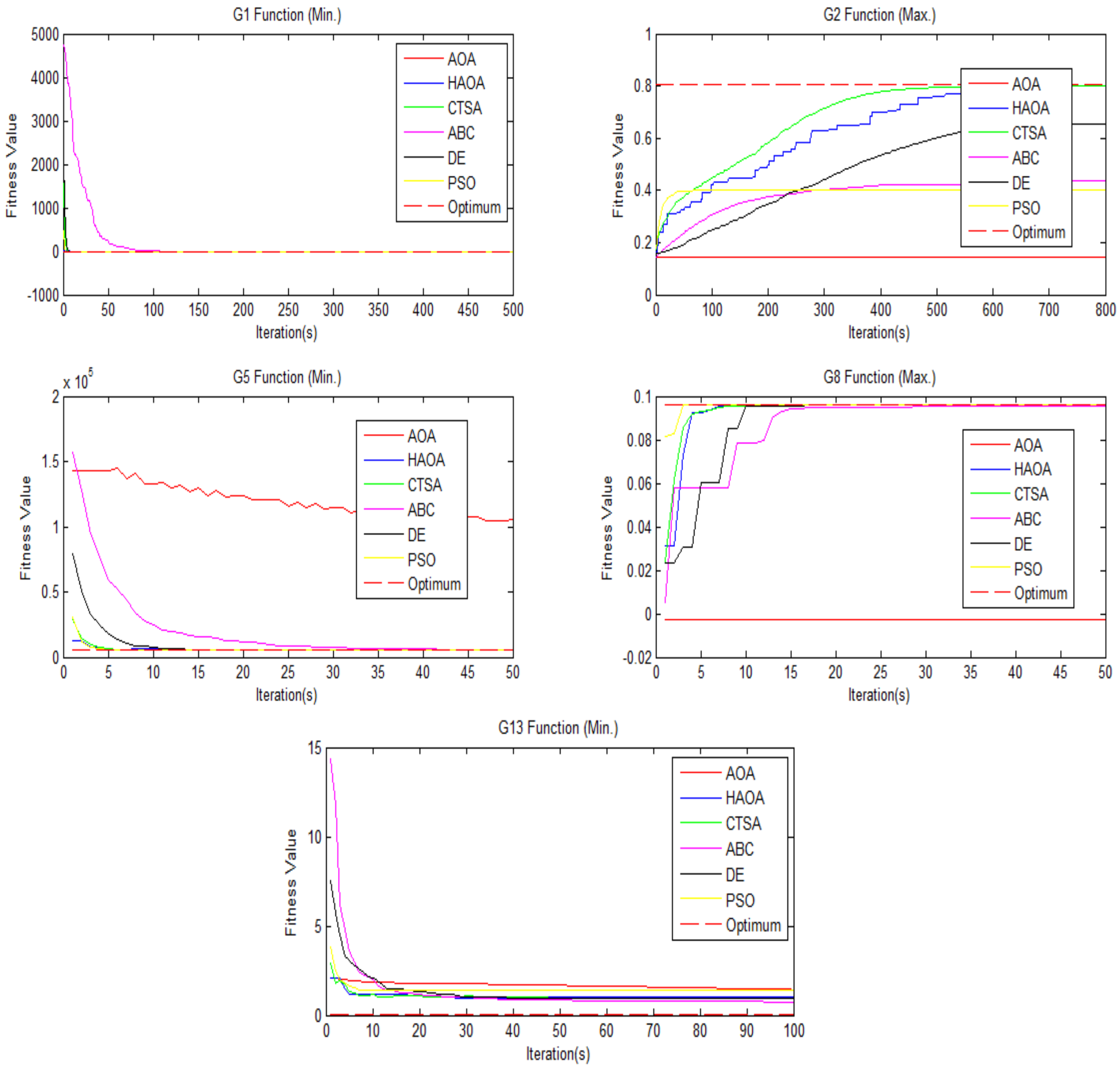

Figure 5. The convergence charts of HAOA, AOA, CTSA, ABC, DE, and PSO for five benchmark problems

3.3. A comparison of HAOA on the engineering design problems

In this subsection, HAOA has been applied to various engineering design problems. There are various engineering design problems in the literature (Braik, 2021). The best known of these are the Welded Beam Design problem (WBD), the Pressure Vessel Design problem (PVD), and Compression Spring Design (CSD). The mathematical models of these problems are taking directly from Babalik et al., (2018). Descriptions of these problems are given in the appendix section of Babalik et al.'s study (Babalik et al., 2018). In this study, 3E + 04 as the maximum evaluation and 20 and 40 as the population sizes were selected, and we tested the success of AOA and HAOA in three different engineering design problems. 30 independent runs were carried out for each function for all the algorithms. The results of CTSA on engineering design problems have been obtained from Babalik et al. (2018). Population size and maximum evaluation parameters were chosen equally to make a fair comparison. In CTSA, the ST value was chosen as 0.2 and 0.5 in the population size of 20 and 40, respectively (Babalik et al., 2018).

Table 12 shows a comparing AOA, HAOA, and CTSA algorithms for population size=20 and 40 on various engineering design problems. Table 13 shows the results of the Wilcoxon Signed-Rank Test on the results of HAOA, AOA, and CTSA algorithms on various engineering design problems. Table 14 shows a comparing HAOA and other algorithms on various engineering design problems. Successful results are marked with bold font in Table 12 and Table 14. According to Table 12, HAOA is more successful than AOA and CTSA.

Table 12. Comparing AOA, HAOA, and CTSA algorithms for population size=20 and 40 on various engineering design problems

Problem		**AOA**		**HAOA**		**CTSA**	
PVD		**N=20**	**N=40**	**N=20**	**N=40**	**N=20**	**N=40**
	Best	6719.607	6963.081	**5901.344**	5907.115	5930.4781	6053.1729
	Worst	22315.94	15660.85	6077.501	5995.747	6459.0682	6398.9258
	Mean	10593.84	9742.615	5977.512	**5944.55**	6104.2594	6194.5821
	SD	3189.202	2200.163	48.08271	**23.04729**	-	-
WBD							
	Best	2.630093	2.61196	2.380959	**2.380958**	2.382411	2.3901825
	Worst	4.542383	3.806217	2.381671	2.381091	2.4807722	2.5505803
	Mean	3.166143	3.158344	2.381056	**2.380987**	2.3891574	2.4290572
	SD	0.403487	0.259825	0.000154	**3.11E-05**	-	-
CSD							
	Best	0.012998	0.013167	0.012674	**0.012671**	-	-
	Worst	0.032921	0.022827	0.012997	0.012763	-	-
	Mean	0.014821	0.013552	0.012753	**0.012704**	-	-
	SD	0.004934	0.001723	6.75E-05	**2.82E-05**	-	-

Table 13. The results of the Wilcoxon Signed-Rank Test on the results of HAOA, AOA, and CTSA algorithms on various engineering design problems

HAOA	**AOA**				**CTSA**			
	N=20		**N=40**		**N=20**		**N=40**	
	P-Value	*h*	*P-Value*	*h*	*P-Value*	*h*	*P-Value*	*h*
PVD	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
WBD	1.73E-06	1	1.73E-06	1	1.73E-06	1	1.73E-06	1
CSD	1.73E-06	1	1.73E-06	1	-	-	-	-

Table 14. Comparing HAOA and other algorithms on various engineering design problems

PVD		f(x)	Fes	Reference
	HAOA	5907.115	30000	-
	AOA	6963.081	30000	-
	CTSA	6053.1729	30000	Babalik et al., 2018
	GWO	**6051.56390000**	N/A	Mirjalili et al., 2014
	MFO	6059.71430000	N/A	Mirjalili, 2015
	WOA	6059.74100000	6300	Mirjalili and Lewis, 2016
WBD				
	HAOA	2.380958	30000	-
	AOA	2.61196	30000	-
	CTSA	2.3901825	30000	Babalik et al., 2018
	HS	**2.38070000**	110000	Lee and Geem, 2005
	Deb	2.43311600	5000	Deb, 1991

4. CONCLUSION

The use of heuristic algorithms in solving real-world problems has increased in recent years. This is due to their success in problem-solving. AOA is a newly developed heuristic algorithm. It uses four arithmetic operations in its structure. The addition and subtraction operators enhanced the AOA's local search capability, while the multiplication and division operators enhanced the AOA's global search capability. It has been hybridized with the TSA algorithm to increase the success of AOA. Thus, hybrid AOA (HAOA) has been proposed. The seed production mechanism of TSA is placed in the random walking stage of AOA. New candidate solutions (seeds) have been produced with the arithmetic operators involved in AOA and the candidate solutions have been compared with the existing solutions. Thus, the performance of AOA has increased. In this study, the success of AOA and HAOA was tested in thirteen constrained optimization problems. There are four maximization and nine minimization problems. The success of AOA and HAOA has been tested for their performance in six different population sizes. Statistical test was applied to the obtained results. The results proved the superiority of HAOA. HAOA has been compared with other heuristic methods (ABC, DE, GA, PSO, EHO, EHO-NoB, HEHO, and GL-EHO) in the literature and the success of HAOA has been shown. In addition, the success of AOA and HAOA has also been tested in three different engineering design problems. The results obtained were evaluated and compared. HAOA's success has been proven.

Since the AOA algorithm is a new algorithm, its performance has not been tested on different problems. In future studies, the performance of AOA on discrete and binary optimization problems is considered.

REFERENCES

Abualigah, L., Diabat, A., Mirjalili, S., Elaziz, MA., Gandomi, A.H., (2021), The Arithmetic Optimization Algorithm, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113609.

Aslan, M., Beskirli, M., Kodaz, H., Kiran, M.S., (2018), An Improved Tree Seed Algorithm for Optimization Problems, International Journal of Machine Learning and Computing, Vol. 8, No. 1.

Babalik, A., Cinar, A.C., Kiran, M.S., (2018), A modification of tree-seed algorithm using Deb's rules for constrained optimization, Applied Soft Computing 63, 289–305.

Bansal, J.C., Joshi, S.K., Sharma, H., (2018), Modified global best artificial bee colony for constrained optimization problems, Computers and Electrical Engineering 67, 365–382.

Beşkirli, A., Özdemir, D., Temurtaş, H., (2020), A comparison of modified tree–seed algorithm for high-dimensional numerical functions, Neural Computing, and Applications, 32:6877–6911.

Braik, M.S., (2021), Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems, Expert Systems With Applications 174, 114685.

Cinar, A.C., Korkmaz, S., Kiran, M.S., (2020), A discrete tree-seed algorithm for solving symmetric traveling salesman problem, Volume 23, Issue 4, Pages 879-890.

Deb, K., (1991), Optimal design of a welded beam via genetic algorithms, AIAA J. 29(11) (1991) 2013–2015.

El-Fergany, A., Hasanien, H.M., (2018), Tree-seed algorithm for solving optimal power flow problem in large-scale power systems incorporating validations and comparisons, Applied Soft Computing, Volume 64, Pages 307-316.

Garg, H., (2016), "A hybrid PSO-GA algorithm for constrained optimization problems", Applied Mathematics and Computation, 274, 292-305. doi:10.1016/j.amc.2015.11.001.

Haklı, H., (2019), A Novel Approach Based On Elephant Herding Optimization For Constrained Optimization Problems, Selcuk Univ. J. Eng. Sci. Tech., v.7, n.2, pp. 405-419.

Jiang, J., Meng, X., Chen, Y., Qiu, C., Liu, Y., Li, K., (2020), Enhancing tree-seed algorithm via feed-back mechanism for optimizing continuous problems, Applied Soft Computing, Volume 92, 106314.

Kiran, M.S., (2015), TSA: Tree-seed algorithm for continuous optimization, Volume 42, Issue 19, Pages 6686-6698.

Kohli, M., Arora, S., (2017), "Chaotic grey wolf optimization algorithm for constrained optimization problems", Journal of Computational Design and Engineering, In Press. Doi:10.1016/j.jcde.2017.02.005.

Lin, C. H., (2013), "A rough penalty genetic algorithm for constrained optimization", Information Sciences, 241, 119-137. Doi:10.1016/j.ins.2013.04.001.

Lee, K.S., Geem, Z.W., (2005), A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput.Methods Appl. Mech. Eng. 194 (36) (2005) 3902–3933.

Mirjalili, S., Mirjalili, S.M., Lewis, A., (2014), Grey wolf optimizer, Adv. Eng. Software 69(2014) 46–61.

Mirjalili, S., (2015), Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm, Knowledge-Based Systems 89 (2015) 228–249.

Mirjalili, S., Lewis, A., (2016), The whale optimization algorithm, Adv. Eng. Software 95(2016) 51–67.

Runarsson, T.P., Yao, X., (2000), "Stochastic ranking for constrained evolutionary optimization", Ieee Transactions on Evolutionary Computation, 4(3), 284-294. Doi: 10.1109/4235.873238.

Strumberger, I., Bacanin, N., Tuba, M., (2018), "Hybridized Elephant Herding Optimization Algorithm for Constrained Optimization", Cham. 158-166.

Xu, B., Chen, X., Tao, L. L., (2018), "Differential evolution with adaptive trial vector generation strategy and cluster-replacement-based feasibility rule for constrained optimization", Information Sciences, 435, 240-262. Doi:10.1016/j.ins.2018.01.014.